Contributions to probabilistic non-negative matrix factorization - Maximum marginal likelihood estimation and Markovian temporal models

Louis Filstroff

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Signal, Image, Acoustique et Optimisation

Présentée et soutenue par :

M. LOUIS FILSTROFF

le mercredi 13 novembre 2019

Titre :

Contributions to probabilistic non-negative matrix factorization - Maximum marginal likelihood estimation and Markovian temporal models

---

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (IRIT)

Directeur(s) de Thèse :

M. CEDRIC FEVOTTE

Rapporteurs :

M. ERIC GAUSSIER, UNIVERSITE GRENOBLE ALPES
M. PIERRE ALQUIER, ECOLE NATIONALE DE STATISTIQUE

Membre(s) du jury :

Mme ELISABETH GASSIAT, UNIVERSITE PARIS-SUD, Président
M. CEDRIC FEVOTTE, CNRS TOULOUSE, Membre
M. JOSEPH SALMON, UNIVERSITE DE MONTPELLIER, Membre

Ah ça y est, je viens de comprendre à quoi ça sert la canne. [...] En fait ça sert à rien. [...] Du coup, ça nous renvoie à notre propre utilité : l'Homme, face à l'absurde !

— *Kaamelott*, Livre IV, Épisode 95 : *L'Inspiration*

Remerciements

Cette thèse n'est certainement pas un exploit individuel ; j'espère que ces quelques mots sauront faire honneur à toutes les personnes qui ont contribué à la réussite de de cette aventure.

Tout d'abord, merci à Cédric de m'avoir fait confiance pour travailler sur cet intrigant sujet de la NMF semi-bayésienne un jour du printemps 2016. Merci de m'avoir encadré et de m'avoir toujours soutenu au cours de ces trois années, en dépit des fausses pistes et frustrations inhérentes au travail de recherche. Travailler avec toi m'a appris de nombreuses choses.

Je souhaiterais ensuite remercier les membres du jury pour leur investissement. Un grand merci à Pierre Alquier et à Éric Gaussier d'avoir accepté de rapporté cette thèse. Je remercie également Élisabeth Gassiat et Joseph Salmon d'avoir examiné mon travail. J'ai été très honoré de votre présence le jour de la soutenance.

Le contenu scientifique de cette thèse doit aussi beaucoup aux personnes suivantes.

Merci beaucoup à Alberto d'avoir été mon plus proche collaborateur scientifique pendant la première moitié de ma thèse. Merci de m'avoir accompagné dans les méandres du Gamma-Poisson et d'avoir encaissé avec moi les (trop nombreux) *rejects* de nos papiers !

Merci à Olivier Cappé de m'avoir accueilli trois jours à l'ENS au mois de Décembre 2018. Les discussions que nous avons eues à ce moment-là ont grandement aidé à former ce qui est aujourd'hui le quatrième chapitre de cette thèse.

Merci à Olivier (Gouvert !) de m'avoir supporté pendant trois ans (en réalité, depuis plus longtemps que ça), pour les innombrables relectures et suggestions, les réunions de crise, et les modèles génératifs de l'impossible. Merci de t'être lancé avec moi dans les modèles temporels.

Cette thèse aurait été bien plus triste sans l'incroyable ambiance régnant au sein de l'équipe SC. Merci à Marie, Nicolas, Thomas, Emmanuel. Vous avez toujours été de précieux conseils. Un immense merci aux doctorant·e·s, passés et présents : Pierre-Antoine, Yanna, Vinicius, Jessica, Dylan, Serdar, Adrien, Étienne, Baha, Mouna, Maxime, Claire, Camille, Asma, Pierre-Hugo. Ce fut un plaisir d'avoir partagé ces trois années avec vous. Merci de m'avoir souvent écouté râler, pour les expertises scientifiques, techniques et logistiques, ainsi que pour les nombreuses grilles de mots fléchés.

Je souhaiterais également remercier tou·te·s les gestionnaires des diverses structures dans lesquelles j'ai pu évoluer. Un merci particulier à Annabelle du côté de l'ENSEEIHT, et à

Clémentine et Chloé du côté de l'UPS.

Avant de conclure, j'aimerais avoir un mot pour les personnes qui m'ont guidé vers la voie de la recherche, bien avant cette thèse. Merci à Pierre Chainais. Merci à Mylène Maïda et David Coupier qui ont encadré ma première expérience de recherche. Merci à Patrick Bas et à tous mes anciens collègues de l'équipe SigMA.

Merci à ma famille d'avoir traversé la France et d'avoir soldé leurs RTT pour venir me voir présenter.

Merci à Rita d'être à mes côtés. En avant pour la prochaine aventure.

Résumé

La factorisation en matrices non-négatives (NMF, de l'anglais *non-negative matrix factorization*) est aujourd'hui l'une des techniques de réduction de la dimensionnalité les plus répandues, dont les domaines d'application recouvrent le traitement du signal audio, l'imagerie hyperspectrale, ou encore les systèmes de recommandation. Sous sa forme la plus simple, la NMF a pour but de trouver une approximation d'une matrice des données non-négative (c'est-à-dire à coefficients positifs ou nuls) par le produit de deux matrices non-négatives, appelées les facteurs. L'une de ces matrices peut être interprétée comme un dictionnaire de motifs caractéristiques des données, et l'autre comme les coefficients d'activation de ces motifs. La recherche de cette approximation de rang faible s'effectue généralement en optimisant une mesure de similarité entre la matrice des données et son approximation. Il s'avère que pour de nombreux choix de mesures de similarité, ce problème est équivalent à l'estimation jointe des facteurs au sens du maximum de vraisemblance sous un certain modèle probabiliste décrivant les données. Cela nous amène à considérer un paradigme alternatif pour la NMF, dans lequel les taches d'apprentissage se portent sur des modèles probabilistes dont la densité d'observation est paramétrisée par le produit des facteurs non-négatifs. Ce cadre général, que nous appelons NMF probabiliste, inclut de nombreux modèles à variables latentes bien connus de la littérature, tels que certains modèles pour des données de comptage.

Dans cette thèse, nous nous intéressons à des modèles de NMF probabilistes particuliers pour lesquels on suppose une distribution a priori pour les coefficients d'activation, mais pas pour le dictionnaire, qui reste un paramètre déterministe. L'objectif est alors de maximiser la vraisemblance marginale de ces modèles semi-bayésiens, c'est-à-dire la vraisemblance jointe intégrée par rapport aux coefficients d'activation. Cela revient à n'apprendre que le dictionnaire, les coefficients d'activation pouvant être inférés dans un second temps si nécessaire. Nous entreprenons d'approfondir l'étude de ce processus d'estimation. En particulier, deux scénarios sont envisagées. Dans le premier, nous supposons l'indépendance des coefficients d'activation par échantillon. Des résultats expérimentaux antérieurs ont montré que les dictionnaires appris via cette approche avaient tendance à régulariser de manière automatique le nombre de composantes ; une propriété avantageuse qui n'avait pas été expliquée alors. Dans le second, nous levons cette hypothèse habituelle, et considérons des structures de Markov, introduisant ainsi de la corrélation au sein du modèle, en vue d'analyser des séries temporelles.

Abstract

Non-negative matrix factorization (NMF) has become a popular dimensionality reduction technique, and has found applications in many different fields, such as audio signal processing, hyperspectral imaging, or recommender systems. In its simplest form, NMF aims at finding an approximation of a non-negative data matrix (i.e., with non-negative entries) as the product of two non-negative matrices, called the factors. One of these two matrices can be interpreted as a dictionary of characteristic patterns of the data, and the other one as activation coefficients of these patterns. This low-rank approximation is traditionally retrieved by optimizing a measure of fit between the data matrix and its approximation. As it turns out, for many choices of measures of fit, the problem can be shown to be equivalent to the joint maximum likelihood estimation of the factors under a certain statistical model describing the data. This leads us to an alternative paradigm for NMF, where the learning task revolves around probabilistic models whose observation density is parametrized by the product of non-negative factors. This general framework, coined probabilistic NMF, encompasses many well-known latent variable models of the literature, such as models for count data.

In this thesis, we consider specific probabilistic NMF models in which a prior distribution is assumed on the activation coefficients, but the dictionary remains a deterministic variable. The objective is then to maximize the marginal likelihood in these semi-Bayesian NMF models, i.e., the integrated joint likelihood over the activation coefficients. This amounts to learning the dictionary only; the activation coefficients may be inferred in a second step if necessary. We proceed to study in greater depth the properties of this estimation process. In particular, two scenarios are considered. In the first one, we assume the independence of the activation coefficients sample-wise. Previous experimental work showed that dictionaries learned with this approach exhibited a tendency to automatically regularize the number of components, a favorable property which was left unexplained. In the second one, we lift this standard assumption, and consider instead Markov structures to add statistical correlation to the model, in order to better analyze temporal data.

Contents

List of Figures 17

List of Tables 21

Notations and Acronyms 23

Usual Probability Distributions 25

1 Introduction 29
- 1.1 Matrix factorization . . . 30
  - 1.1.1 General introduction . . . 30
  - 1.1.2 Principal component analysis . . . 32
- 1.2 Non-negative matrix factorization . . . 32
  - 1.2.1 Problem statement . . . 32
  - 1.2.2 Limitations . . . 34
  - 1.2.3 Choice of the divergence . . . 35
  - 1.2.4 Standard algorithms . . . 37
  - 1.2.5 An example: the original experiment . . . 39
- 1.3 Probabilistic non-negative factorization . . . 40
  - 1.3.1 Definition . . . 40
  - 1.3.2 List of models . . . 41
  - 1.3.3 Model variants and learning problems . . . 43
- 1.4 Inference in semi-Bayesian NMF . . . 44
  - 1.4.1 Estimators . . . 44
  - 1.4.2 Related works to MMLE . . . 45
    - 1.4.2.1 Integrating out nuisance parameters . . . 45
    - 1.4.2.2 A note on LDA . . . 46
    - 1.4.2.3 A note on independent component analysis (ICA) . . . 46
  - 1.4.3 Categories of priors on the activation coefficients . . . 47
- 1.5 Structure of the manuscript and contributions . . . 47

Appendices to Chapter 1 49
- 1.A The divergences used in NMF . . . 49
  - 1.A.1 Parametrized families . . . 49

1.A.2 Families generated by a function . . . 50
1.A.2.1 Bregman divergences . . . 50
1.A.2.2 Csiszar divergences . . . 50
1.B The majorization-minimization framework . . . 50
1.C Exponential dispersion models and Tweedie distributions . . . 52

List of Publications **53**

2 Maximum Marginal Likelihood Estimation in the Gamma-Poisson Model **55**
2.1 Introduction . . . 56
2.2 The Gamma-Poisson model . . . 57
2.2.1 First formulation . . . 57
2.2.2 Composite structure of the model . . . 58
2.3 New formulations of GaP . . . 58
2.3.1 GaP as a composite negative multinomial model . . . 58
2.3.2 GaP as a composite multinomial model . . . 59
2.4 Closed-form marginal likelihood . . . 60
2.4.1 Analytical expression . . . 60
2.4.2 Self-regularization . . . 62
2.5 Optimization algorithms . . . 63
2.5.1 Expectation-Maximization . . . 63
2.5.2 Monte Carlo E-step . . . 64
2.5.3 M-step . . . 65
2.6 Experimental work . . . 67
2.6.1 Comparison of the algorithms . . . 67
2.6.1.1 Experiments with synthetic data . . . 67
2.6.1.2 Experiments with real data . . . 72
2.6.2 Examples of the self-regularization phenomenon . . . 74
2.6.2.1 On synthetic data . . . 74
2.6.2.2 On a real dataset . . . 76
2.7 Discussion . . . 76

Appendices to Chapter 2 **78**
2.A Probability distributions . . . 78
2.A.1 Negative binomial distribution . . . 78
2.A.2 Negative multinomial distribution . . . 79
2.B Stars and bars theorem . . . 80
2.C Gibbs sampling of the posterior distribution . . . 80
2.C.1 First conditional . . . 80
2.C.2 Second conditional . . . 81
2.D EM algorithms . . . 81
2.D.1 MCEM-CH . . . 81
2.D.2 MCEM-H . . . 82

2.D.3 MCEM-C . . . 83

3 Maximum Marginal Likelihood Estimation in the Multiplicative Exponential Model 87
3.1 Introduction . . . 88
3.2 Model . . . 89
3.2.1 Observation model . . . 89
3.2.2 Working with complex data . . . 90
3.2.3 Prior distribution . . . 90
3.2.4 Objective function . . . 91
3.3 Marginalization of **H** . . . 91
3.4 Marginal likelihood . . . 92
3.4.1 Analytical expression . . . 92
3.4.2 Self-regularization . . . 93
3.5 Optimization algorithms . . . 93
3.5.1 Expectation-Minimization . . . 93
3.5.2 Monte Carlo E-step . . . 94
3.5.3 Monte Carlo M-step . . . 95
3.6 Estimating audio sources . . . 97
3.7 Experimental work . . . 98
3.7.1 Experimental setup . . . 98
3.7.2 Results . . . 99
3.8 Discussion . . . 100

Appendices to Chapter 3 104
3.A Probability distributions . . . 104
3.A.1 Complex normal distribution . . . 104
3.A.2 Multivariate Student's t-distribution . . . 105
3.B Gibbs sampling of the posterior distribution . . . 105
3.B.1 First conditional . . . 106
3.B.2 Second conditional . . . 106
3.C EM algorithms . . . 107
3.C.1 MCEM-CH . . . 107
3.C.2 MCEM-H . . . 107
3.C.3 MCEM-C . . . 108

4 Temporal Non-Negative Matrix Factorization 109
4.1 Introduction . . . 110
4.2 Comparative study of Gamma Markov chains . . . 111
4.2.1 Direct chaining on the rate parameter . . . 112
4.2.1.1 Model . . . 112
4.2.1.2 Analysis . . . 112

4.2.2 Hierarchical chaining with an auxiliary variable . . . 113
4.2.2.1 Model . . . 113
4.2.2.2 Analysis . . . 114
4.2.3 Chaining on the shape parameter . . . 115
4.2.3.1 Model . . . 115
4.2.3.2 Analysis . . . 116
4.2.4 BGAR(1) . . . 116
4.2.4.1 Model . . . 117
4.2.4.2 Analysis . . . 117
4.3 The BGAR-NMF model . . . 119
4.4 Maximum marginal likelihood estimation . . . 121
4.4.1 Objective . . . 121
4.4.2 Sequential Monte Carlo methods . . . 121
4.4.2.1 Particle filtering . . . 122
4.4.2.2 Particle smoothing . . . 122
4.4.3 M-step . . . 123
4.4.4 Experimental work . . . 123
4.5 MAP estimation . . . 125
4.5.1 Problem setting . . . 125
4.5.2 Optimization . . . 126
4.5.3 Experimental work . . . 129
4.5.3.1 On synthetic data . . . 129
4.5.3.2 On a real dataset . . . 131
4.6 Discussion . . . 132

Appendices to Chapter 4 **133**
4.A Moments . . . 133
4.A.1 Product of independent random variables . . . 133
4.A.2 Laws of total expectation and variance . . . 134
4.B Beta prime distribution . . . 134
4.C BGAR(1) linear correlation . . . 134
4.D Bootstrap particle filter . . . 134
4.E MAP estimation in temporal NMF models . . . 135
4.E.1 Direct chaining on the rate parameter . . . 136
4.E.2 Hierarchical chaining with an auxiliary variable . . . 136
4.E.3 Chaining on the shape parameter . . . 137
4.F MAP estimation in the GaP model . . . 138

Conclusion **139**

A Résumé Substantiel en Français **143**

B Other works **151**
B.1 Bayesian Mean-Parameterized Non-Negative Binary Matrix Factorization . . 151
B.2 A Ranking Model Motivated by Nonnegative Matrix Factorization with Applications to Tennis Tournaments 153

Bibliography **155**

List of Figures

1.1 Illustrative matrix factorization. In this example, the observation matrix $\mathbf{V}$ can be exactly factorized as $\mathbf{WH}$ with $K = 3$. The activation coefficients are binary, with a black dot representing a one, and a white dot a zero. 31
1.2 The family of the β-divergences. Each curve represents the β-divergence as a function of y, with the value of x set to 1, for five different values of β. . . . 37
1.3 $K = 30$ atoms of the dictionary learned on the CBCL dataset. 39
1.4 Eight faces of the CBCL dataset chosen at random. Top: original images. Bottom: reconstruction with $K = 30$. 40
1.5 Probabilistic NMF models. From left to right. (a) Frequentist NMF models. Neither $\mathbf{W}$ nor $\mathbf{H}$ is assumed to be a random variable. (b) Bayesian NMF models. Both $\mathbf{W}$ and $\mathbf{H}$ are assumed to be random variables with a prior distribution. (c) Semi-Bayesian NMF models. $\mathbf{W}$ is assumed to be a parameter, whereas a prior distribution is assumed on $\mathbf{H}$. 44
1.6 Illustrative example of an MM algorithm. The function f to minimize is displayed in blue. Its minimum is achieved at $x = x^\star$. Assuming that $x^{(i)} = 0.75$, we obtain the majorizing function in red. Its minimization yields the next iterate $x^{(i+1)}$. 51

2.1 Graphical representations of the Gamma-Poisson model. Observed variables are in blue, while latent variables are in white. Deterministic parameters are represented as black dots. The observation $\mathbf{v}_n$ is a vector of size F. From left to right:(a) The standard model. The latent variable $\mathbf{h}_n$ is of size K; (b) The augmented model with variables $\mathbf{C}$. The latent variable $\mathbf{c}_{kn}$ is of size F, and h_{kn} is scalar. 57
2.2 Common non-convex penalties represented in the scalar case. The log-sum penalty $f(x) = \log(|x| + \beta) - \log(\beta)$ is displayed in blue for two values of β. The ℓ_q pseudo-norm, defined as $g(x) = |x|^q$, is displayed in green for two values of q. The SCAD penalty is displayed in red with $a = 3$. 63
2.3 $\mathcal{L}(\mathbf{W})$ w.r.t. CPU time in seconds for the three MCEM algorithms on toy dataset $\mathbf{V}_1$. From top to bottom:(a) $\beta = 1$; (b) $\beta = 100$. 69
2.4 Evolution of the norm of each of the $K = 3$ columns $\mathbf{w}_k$ of the dictionary w.r.t. CPU time in seconds for the three MCEM algorithms on toy dataset $\mathbf{V}_1$. From top to bottom:(a) $\beta = 1$; (b) $\beta = 100$. 70

2.5 Evolution of the norm of each of the $K = 3$ columns $\mathbf{w}_k$ of the dictionary w.r.t. CPU time in seconds for the three MCEM algorithms on toy dataset $\mathbf{V}_2$. From top to bottom:(a) $\beta = 1$; (b) $\beta = 0.01$. 71
2.6 Evolution of the norm of each of the $K = 10$ columns of the dictionaries w.r.t. CPU time in minutes for the three MCEM algorithms on the `Taste Profile` dataset. 73
2.7 $\mathcal{L}(\hat{\mathbf{W}})$ w.r.t. K. 74
2.8 (a) Ground truth dictionary $\mathbf{W}^\star$; (b)-(g) Estimated dictionaries $\hat{\mathbf{W}}$ by MMLE for K from 1 (b) to 8 (i). The color bar displayed in (a) is common to all subplots. 75
2.9 Norm of the $K = 100$ columns of the dictionary learned on the `Taste Profile` dataset. 76

3.1 The piano dataset. From top to bottom:(a) Time-domain recorded signal. (b) Log-power spectrogram $\mathbf{V} = |\mathbf{X}|^2$ 99
3.2 Evolution of the norm of each of the $K = 10$ columns $\mathbf{w}_k$ of the dictionary in $\log_{10}$ scale w.r.t. the number of EM iterations for MCEM-CH on the piano dataset. 100
3.3 Columns of $\mathbf{W}$ in $\log_{10}$ scale w.r.t. frequency bin f. From left to right :(a) With IS-NMF (in blue). (b) With MMLE (in red). For each method, the $K = 10$ components are sorted by the decreasing variance of the associated time-domain signal. 101
3.4 Reconstructed time-domain components. From left to right :(a) With IS-NMF (in blue). (b) With MMLE (in red). For each method, the $K = 10$ components are sorted by decreasing variance. 102

4.1 Realizations of the Markov chain defined in Eq. (4.4). The initial value h_1 is set to 1, and chains were simulated until $n = 50$. Each subplot contains ten independent realizations, with the value of the parameters (α, β) given at the top of the subplot. $\log_{10}(h_n)$ is displayed. 113
4.2 Realizations of the Markov chain defined in Eq. (4.9)-(4.10). The initial value h_1 is set to 1, and chains were simulated until $n = 50$. Each subplot contains ten independent realizations, with the value of the parameters $(\alpha_z, \beta_z, \alpha_h, \beta_h)$ given at the top of the subplot. $\log_{10}(h_n)$ is displayed. 115
4.3 Realizations of the Markov chain defined in Eq. (4.18). The initial value h_1 is set to 1, and chains were simulated until $n = 50$. Each subplot contains ten independent realizations, with the value of the parameters (α, β) given at the top of the subplot. $\log_{10}(h_n)$ is displayed. 116
4.4 Three realizations of the BGAR(1) process, with parameters fixed to $\alpha = 2$ and $\beta = 1$, and a different parameter ρ in each subplot. The mean of the process is displayed by a dashed red line. 118
4.5 The BGAR-NMF model. Observed variables are in blue, while latent variables are in white. In our setting, $\mathbf{h}_n$ is of size K, and $\mathbf{v}_n$ is of size F. 120

4.6 The augmented BGAR-NMF model. Observed variables are in blue, while latent variables are in white. In our setting, $\mathbf{b}_n$ is of size K, $\mathbf{h}_n$ is of size K, and $\mathbf{v}_n$ is of size F. The hyperparameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}$ have been omitted for enhanced readability. 120
4.7 Evolution of the norm of the $K = 5$ columns $\mathbf{w}_k$ of the dictionary w.r.t. the number of EM iterations on synthetic dataset $\mathbf{V}_s$. 124
4.8 Evolution of the norm of the $K = 10$ columns $\mathbf{w}_k$ of the dictionary w.r.t. the number of EM iterations on the `NIPS` dataset. 125
4.9 Hyperparameter values (in white) of the parameters α_k and ρ_k ensuring a well-posed MAP estimation in the BGAR-NMF model. 129
4.10 Example of the evolution of the cost function C w.r.t. the number of iterations on a synthetic dataset. The hyperparameters are here set to $\rho = 0.9$, $\alpha = 11$ and $\beta = 1$. 130
4.11 Evolution of the learned $\mathbf{H}$ w.r.t. the value of ρ on a synthetic dataset. For all subplots, the value of α is set to $(1 - \rho)^{-1} + 1$, and β to 1. 130

List of Tables

1.1 Typical examples of data available as matrices. 30

2.1 Characteristics of the three datasets considered for the experimetanl comparison of the three EM algorithms. $\mathbf{V}_1$ and $\mathbf{V}_2$ are synthetic datasets, whereas the `Taste Profile` is a real dataset. The over-dispersion ratio corresponds to the variance to mean ratio. 72

4.1 Prediction results on the `NIPS` dataset. Lower values are better. The mean and standard deviation of each error is reported over 100 runs (10 different splits with 10 different initializations). 131

4.2 Coefficients of the order-2 polynomial equation (Eq. (4.101)) w.r.t. k and n. . 136

Notations and Acronyms

Generic notations

a	Scalar a
$\mathbf{a}$	Column vector $\mathbf{a}$
a_i	i-th entry of $\mathbf{a}$
$\mathbf{A}$	Matrix A
a_{ij}	Entry (i, j) of the matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	Inverse matrix of $\mathbf{A}$
$\det(\mathbf{A})$	Determinant of the matrix $\mathbf{A}$
$(.)^{\mathrm{T}}$	Transpose
$(.)^{\mathrm{H}}$	Hermitian transpose
$\mathbf{a}_i$	i-th column of $\mathbf{A}$
$\underline{\mathbf{a}}_i$	i-th row of $\mathbf{A}$
$\mathbf{AB}$	Matrix product
$\mathbf{A} \odot \mathbf{B}$	Hadamard product
$\mathrm{diag}(\mathbf{a})$	Diagonal matrix based on $\mathbf{a}$
$\|\|\mathbf{a}\|\|_p$	p-norm of $\mathbf{a}$
$\|\|\mathbf{A}\|\|_F$	Frobenius norm of $\mathbf{A}$
$\mathbf{0}_K$	Column vector of size K full of zeros
$\mathbf{1}_K$	Column vector of size K full of ones
$\mathbf{I}_K$	Identity matrix of size K

Spaces

$\mathbb{R}$	The set of real numbers
$\mathbb{R}^K$	The set of real vectors of size K
$\mathbb{R}_+$	The set of non-negative numbers
$\mathbb{N}$	The set of natural numbers (0 included)
$\mathbb{C}$	The set of complex numbers

Probabilistic notations

X	Random variable
$\mathbb{E}(X)$	Mean of X
$\text{var}(X)$	Variance of X
$X \sim$	X is distributed w.r.t.
i.i.d.	Independent and identically distributed
$\hat{\theta}$	Point estimate of θ

Usual probability distributions as well as their associated notations are defined page 25.

Miscellaneous

$\Gamma(.)$	Gamma function
$\Psi(.)$	Digamma function
$B(.)$	Beta function
$\propto$	Proportional to
$\overset{c}{=}$	Equal up to a constant

Acronyms

AR	Autoregressive
EB	Empirical Bayes
EM	Expectation-Maximization
ICA	Independent Component Analysis
IS	Itakura-Saito
KL	Kullback-Leibler
LDA	Latent Dirichlet Allocation
MAP	Maximum A Posteriori
MF	Matrix Factorization
MJLE	Maximum Joint Likelihood Estimation
ML	Maximum Likelihood
MM	Majorization-Minimization
MMLE	Maximum Marginal Likelihood Estimation
NMF	Non-negative Matrix Factorization
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
STFT	Short Time Fourier Transform

Usual Probability Distributions

We here introduce the usual probability distributions used in this thesis. Specific, lesser-known distributions that arise in the course of a chapter are defined in the associated appendix to that chapter.

Discrete univariate distributions

Bernoulli distribution

The probability mass function (p.m.f.) of a Bernoulli random variable X, parametrized by a probability parameter $p \in [0, 1]$, is such that:

$$\mathbb{P}(X = 1; p) = p, \quad \mathbb{P}(X = 0; p) = 1 - p. \tag{0.1}$$

We write $X \sim \text{Bernoulli}(p)$. The p.m.f. may alternatively be written as, for $k \in \{0, 1\}$:

$$\mathbb{P}(X = k; p) = p^k (1 - p)^{1-k}. \tag{0.2}$$

We have

$$\mathbb{E}(X) = p, \quad \text{var}(X) = p(1 - p). \tag{0.3}$$

Poisson distribution

The p.m.f. of a Poisson random variable X, parametrized with a rate parameter $\lambda > 0$, is such that, for all $c \in \mathbb{N}$:

$$\mathbb{P}(X = c; \lambda) = \frac{\lambda^c}{c!} \exp(-\lambda). \tag{0.4}$$

We write $X \sim \text{Poisson}(\lambda)$.

We have

$$\mathbb{E}(X) = \lambda, \quad \text{var}(X) = \lambda. \tag{0.5}$$

Continuous univariate distributions

Normal distribution

The probability density function (p.d.f.) of a normal random variable X, parametrized by a mean parameter $\mu \in \mathbb{R}$ and a variance parameter $\sigma^2 > 0$, is such that, for all $x \in \mathbb{R}$:

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{0.6}$$

We write $X \sim \mathcal{N}(\mu, \sigma^2)$.

We have

$$\mathbb{E}(X) = \mu, \quad \text{var}(X) = \sigma^2. \tag{0.7}$$

Gamma distribution

The p.d.f. of a Gamma random variable X, parametrized by a shape parameter $\alpha > 0$ and a rate parameter $\beta > 0$, is such that, for all $x > 0$:

$$f(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x). \tag{0.8}$$

We write $X \sim \text{Gamma}(\alpha, \beta)$. Γ is the Gamma function, defined as, for all $z > 0$:

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} \exp(-t) \mathrm{d}t. \tag{0.9}$$

We have

$$\mathbb{E}(X) = \frac{\alpha}{\beta}, \quad \text{var}(X) = \frac{\alpha}{\beta^2}. \tag{0.10}$$

When $\alpha = 1$, the Gamma distribution reduces to the exponential distribution, whose p.d.f. is therefore, for all $x > 0$:

$$f(x; \beta) = \frac{1}{\beta} \exp(-\beta x). \tag{0.11}$$

Inverse Gamma distribution

Let X be a Gamma random variable with shape parameter α and rate parameter β. Then $Y = \frac{1}{X}$ follows an inverse Gamma distribution. For all $x > 0$, its p.d.f. is given by:

$$f(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{x}\right)^{\alpha+1} \exp\left(-\frac{\beta}{x}\right). \tag{0.12}$$

Note that β is a *scale* parameter of the distribution. We write $X \sim \mathcal{IG}(\alpha, \beta)$.

We have

$$\mathbb{E}(X) = \frac{\beta}{\alpha - 1} \qquad \text{for } \alpha > 1, \text{ undefined otherwise,} \tag{0.13}$$

$$\text{var}(X) = \frac{\beta^2}{(\alpha-1)^2(\alpha-2)} \qquad \text{for } \alpha > 2, \text{ undefined otherwise.} \tag{0.14}$$

Beta distribution

The p.d.f. of a Beta random variable X, parametrized by a two shape parameters $\alpha > 0$ and $\beta > 0$, is such that, for all $x \in [0, 1]$:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\text{B}(\alpha, \beta)}. \tag{0.15}$$

We write $X \sim \text{Beta}(\alpha, \beta)$. B is the Beta function, defined as, for all $x > 0, y > 0$:

$$\text{B}(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}. \tag{0.16}$$

We have

$$\mathbb{E}(X) = \frac{\alpha}{\alpha+\beta}, \quad \text{var}(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}. \tag{0.17}$$

Multivariate distributions

Multinomial distribution

The p.m.f. of a multinomial random vector $X = (X_1, \ldots, X_K)$ is parametrized by $n \in \mathbb{N}^\star$ (the number of trials), and $\mathbf{p} = [p_1, \ldots, p_K]^{\text{T}}$ such that $0 \leq p_i \leq 1$ and $\sum_i p_i = 1$ (the event probabilities). We have, for all $(x_1, \ldots, x_K) \in \{0, \ldots, n\}^K$ such that $\sum_i x_i = n$:

$$\mathbb{P}(X_1 = x_1, \ldots, X_K = x_K \,; n, \mathbf{p}) = \frac{n!}{x_1! \cdots x_K!} \prod_{i=1}^{K} p_i^{x_i}. \tag{0.18}$$

We write $X \sim \text{Mult}(n, \mathbf{p})$.

We have

$$\mathbb{E}(X_i) = np_i, \qquad \text{var}(X_i) = np_i(1-p_i). \tag{0.19}$$

Dirichlet distribution

The p.d.f. of a Dirichlet random vector $X = (X_1, \ldots, X_K)$, parametrized by $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]^{\text{T}}$ with $\alpha_i > 0$, is such that, for all $(x_1, \ldots, x_K) \in [0, 1]^K$ such that $\sum_i x_i = 1$:

$$f(x_1, \ldots, x_K; \boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}. \tag{0.20}$$

We write $X \sim \mathrm{Dir}(\boldsymbol{\alpha})$.

We have

$$\mathbb{E}(X_i) = \frac{\alpha_i}{\sum_{i=1}^K \alpha_i} = \tilde{\alpha}_i, \quad \mathrm{var}(X_i) = \frac{\tilde{\alpha}_i(1-\tilde{\alpha}_i)}{\sum_{i=1}^K \alpha_i + 1}. \tag{0.21}$$

Chapter 1

Introduction

This chapter introduces the general concept of matrix factorization, before focusing on the specific problem of non-negative matrix factorization and its probabilistic variants.

Contents

1.1 Matrix factorization . . . 30
1.1.1 General introduction . . . 30
1.1.2 Principal component analysis . . . 32
1.2 Non-negative matrix factorization . . . 32
1.2.1 Problem statement . . . 32
1.2.2 Limitations . . . 34
1.2.3 Choice of the divergence . . . 35
1.2.4 Standard algorithms . . . 37
1.2.5 An example: the original experiment . . . 39
1.3 Probabilistic non-negative factorization . . . 40
1.3.1 Definition . . . 40
1.3.2 List of models . . . 41
1.3.3 Model variants and learning problems . . . 43
1.4 Inference in semi-Bayesian NMF . . . 44
1.4.1 Estimators . . . 44
1.4.2 Related works to MMLE . . . 45
1.4.3 Categories of priors on the activation coefficients . . . 47
1.5 Structure of the manuscript and contributions . . . 47

1.1 Matrix factorization

1.1.1 General introduction

In many situations, data is available in matrix form. Indeed, consider a collection of N samples $\mathbf{v}_n$ ($n \in \{1, \dots, N\}$) belonging to $\mathbb{R}^F$ (i.e., described by F real features or attributes). These samples can be stored column-wise, yielding an $F \times N$ matrix, which we denote by $\mathbf{V}$. The matrix $\mathbf{V}$ is referred to as the observation matrix, or the data matrix. Several examples of such matrices are given in Table 1.1.

$\mathbf{V}$ represents	f	n	Typical F
A corpus of documents	Words	Documents	$10^4 - 10^5$
A collection of grayscale images	Pixels	Images	$10^4 - 10^6$
An audio signal	Frequencies	Time frames	$10^3 - 10^4$
Ratings	Items	Users	$10^6 - 10^8$

Table 1.1: Typical examples of data available as matrices.

Generally speaking, matrix factorization (MF) techniques aim at finding an approximation of $\mathbf{V}$ as the product of two matrices:

$$\mathbf{V} \simeq \mathbf{W}\mathbf{H}, \tag{1.1}$$

where $\mathbf{W}$ is of size $F \times K$, and $\mathbf{H}$ is of size $K \times N$. They are jointly referred to as the *factors*. The factorization rank K is usually chosen such that $K \ll \min(F, N)$, hence producing a low-rank approximation of the data matrix $\mathbf{V}$. In this case, matrix factorization is a linear dimensionality reduction technique, since every sample is approximated by a linear combination of K basis elements:

$$\mathbf{v}_n \simeq \sum_{k=1}^{K} h_{kn}\mathbf{w}_k. \tag{1.2}$$

More specifically, the columns of $\mathbf{W}$ (sometimes called the atoms) represent characteristic elements or recurring patterns of the data, and as such $\mathbf{W}$ is referred to as the dictionary, or the basis matrix. As for the columns of $\mathbf{H}$, they encode how much of each atom is needed to represent each sample, and are referred to as the activation coefficients, or the score matrix. An illustrative matrix factorization is displayed on Figure 1.1.

The choice of the factorization rank K is a challenging question. Most of the time, the value of K is set beforehand. One has to consider the trade-off between loss of information and computational efficiency. Moreover, in some settings, K has a physical meaning, such as the number of sources when considering audio signal processing, which makes the choice even more complex.

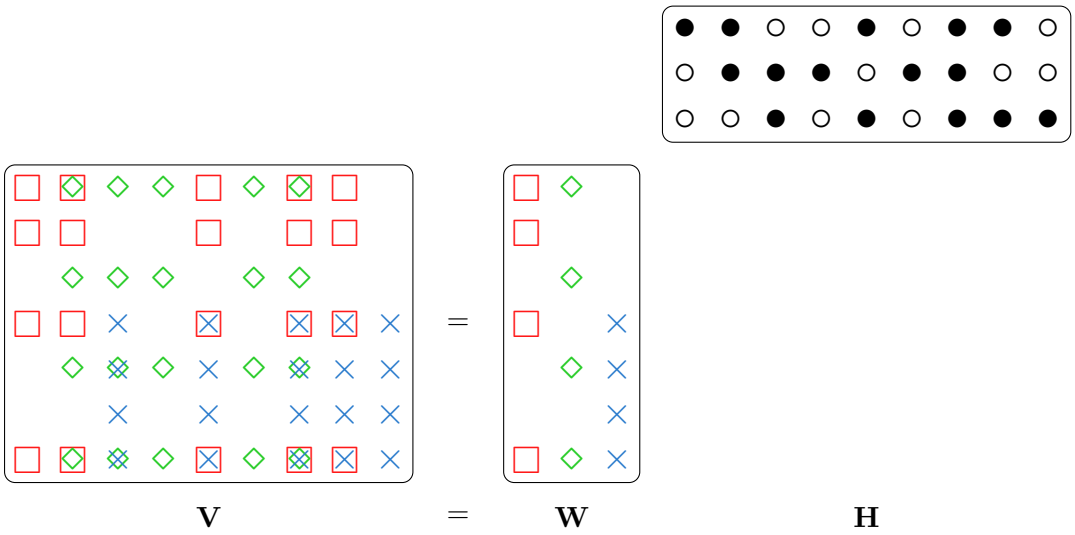


Figure 1.1: Illustrative matrix factorization. In this example, the observation matrix $\mathbf{V}$ can be exactly factorized as $\mathbf{WH}$ with $K = 3$. The activation coefficients are binary, with a black dot representing a one, and a white dot a zero.

The goals of matrix factorization techniques are therefore two-fold. Firstly, as previously explained, the dimensionality of the problem is linearly reduced, i.e., the data is projected in a low-dimensional subspace such that each sample is approximated as a linear combination of atoms. Secondly, characteristic patterns are extracted from the data. Indeed, matrix factorization methods automatically uncover some latent structure of the data. As such, they are part of a much broader family of learning methods called unsupervised learning.

Two important questions arise at this point. First of all, we must define a way to quantify the fit between $\mathbf{V}$ and its approximation $\mathbf{WH}$, i.e., we must choose a certain loss function D. The objective will be to minimize this function, hence casting matrix factorization as an optimization problem. The problem can be stated as:

$$\min_{\mathbf{W},\mathbf{H}} D(\mathbf{V}|\mathbf{WH}). \tag{1.3}$$

We emphasize that the notation $D(\mathbf{V}|\mathbf{WH})$ is to be understood as a function of $\mathbf{W}$ and $\mathbf{H}$. Secondly, additional constraints might be added on the factors $\mathbf{W}$ and $\mathbf{H}$ for interpretability reasons. Combinations of both elements (specific choices of a loss function and of constraints on the factors) lead to a variety of well-known problems from the data mining, machine learning, and signal processing communities. In particular, matrix factorization problems have received a great deal of attention under the name "dictionary learning" (Olshausen and Field, 1996), where it is assumed that the representation of $\mathbf{v}_n$ over the dictionary $\mathbf{W}$, namely $\mathbf{h}_n$, should be sparse (see Mairal et al. (2010) and references therein). We will present the most classical matrix factorization problem in the following subsection, before focusing on the specific case of non-negative matrix factorization in Section 1.2.

1.1.2 Principal component analysis

Principal component analysis (PCA) is probably the most well-known data analysis technique, whose origins can be dated to Pearson (1901) and Hotelling (1933). Basically, PCA amounts to sequentially finding a set of K orthogonal vectors, called the principal components, so that the variance of the projected data onto the subspace spanned by these principal components is maximized. As it turns out, PCA can be cast as a matrix factorization problem. Indeed, an equivalent formulation to this problem is

$$\min_{\mathbf{W},\mathbf{H}} ||\mathbf{V} - \mathbf{W}\mathbf{H}||_F^2, \quad \text{s.t. } \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}_K, \tag{1.4}$$

where the constraint ensures that $\mathbf{W}$ is an orthogonal matrix, and $||.||_F$ denotes the Frobenius matrix norm:

$$||\mathbf{V}||_F = \sqrt{\sum_{f,n} |v_{fn}|^2}. \tag{1.5}$$

This minimization problem can be solved exactly thanks to the singular value decomposition (SVD), a factorization that exists for any real matrix. When dealing with an $F \times N$ matrix, as is the matrix $\mathbf{V}$, the SVD writes as

$$\mathbf{V} = \tilde{\mathbf{U}}\mathbf{\Sigma}\tilde{\mathbf{V}}, \tag{1.6}$$

where

- $\tilde{\mathbf{U}} \in \mathbb{R}^{F\times F}$ is an orthogonal matrix ($\tilde{\mathbf{U}}^{\mathrm{T}}\tilde{\mathbf{U}} = \mathbf{I}_F$);
- $\mathbf{\Sigma} \in \mathbb{R}^{F\times N}$ is a matrix whose diagonal entries are non-negative and sorted in decreasing order (the so-called singular values σ_i), and zero elsewhere;
- $\tilde{\mathbf{V}} \in \mathbb{R}^{N\times N}$ is an orthogonal matrix ($\tilde{\mathbf{V}}^{\mathrm{T}}\tilde{\mathbf{V}} = \mathbf{I}_N$).

The Eckart-Young-Mirsky theorem states that the rank-K matrix $\mathbf{M}$ that minimizes $||\mathbf{V} - \mathbf{M}||_F^2$ is the truncated SVD of $\mathbf{V}$, that is:

$$\mathbf{M} = \sum_{k=1}^{K} \sigma_k \tilde{\mathbf{u}}_k \tilde{\mathbf{v}}_k^{\mathrm{T}}. \tag{1.7}$$

Thus, the PCA problem of Eq. (1.4) is solved by choosing $\mathbf{W}$ to be the first K columns of $\tilde{\mathbf{U}}$, and $\mathbf{H}$ to be the first K rows of $\mathbf{\Sigma}\tilde{\mathbf{V}}$.

1.2 Non-negative matrix factorization

1.2.1 Problem statement

A shortcoming of the aforementioned matrix factorization methods is that they produce blind representations to the support of the data. However, naturally non-negative data

arises in a wide variety of scenarios: physical measurements, counts of occurrences, pixel intensities... As a matter of fact, all examples of data presented in Table 1.1 are non-negative. How should the unconstrained principal components of PCA be interpreted in this setting? This is where non-negative matrix factorization (NMF) steps in, by adding non-negativity constraints on both factors to ensure interpretability.

We give a first formulation of the problem. Given a non-negative[1] matrix $\mathbf{V}$ of size $F \times N$, NMF aims at finding the best rank-K approximation of $\mathbf{V}$ as the product of two non-negative matrices $\mathbf{W}$ and $\mathbf{H}$. Once again, the word "best" refers to the minimization of a certain measure of fit D between $\mathbf{V}$ and its approximation $\mathbf{WH}$:

$$\min_{\mathbf{W}\geq 0,\ \mathbf{H}\geq 0} D(\mathbf{V}|\mathbf{WH}), \tag{1.8}$$

where the notation $\mathbf{A} \geq 0$ denotes the non-negativity of the entries of the matrix $\mathbf{A}$. The non-negativity constraints enable the following. First, the dictionary $\mathbf{W}$ lies in the same space as the data, and each of its columns can therefore be directly interpreted as a characteristic pattern of the data. Secondly, the non-negativity of $\mathbf{H}$ imposes *constructive* representations only, that is, each sample $\mathbf{v}_n$ is described as a weighted sum of the (non-negative) columns of $\mathbf{W}$. Subtractions are, by definition, unfeasible. This explains the ability of NMF to produce part-based representations.

As with general MF techniques, K is usually chosen such that $K \ll \min(F, N)$[2]. If K was set to F or N, the NMF could be solved exactly with trivial solutions. When this is not the case, one cannot expect NMF to produce an exact factorization, especially when working with real data, and therefore the factorization remains an approximate one.

Early work on NMF was conducted by researchers in the field of chemometrics in the early 90s under the name "positive matrix factorization" (Paatero and Tapper, 1994; Paatero, 1997). However, the approach was really popularized by the seminal papers of Lee and Seung (Lee and Seung, 1999, 2000), who coined its definitive name[3], highlighted the part-based representations (making connections with neurological processes), and proposed extremely simple and efficient algorithms.

NMF has found a wide variety of application fields. We detail three illustrative examples.

- In audio signal processing, $\mathbf{V}$ is the amplitude spectrogram of an audio signal. Each column of $\mathbf{V}$ corresponds to the squared module of the coefficients of the short-time Fourier transform of the signal over F frequency bins. As such, in a an NMF decomposition, $\mathbf{W}$ represents the spectra of K audio sources, and $\mathbf{H}$ the time frames activations. This has notably been used for automatic music transcription (Smaragdis and Brown, 2003), or blind source separation (Virtanen, 2007).

[1] We emphasize that the adjective "non-negative" is to be understood w.r.t. the entries of the matrix.

[2] Note that so-called *overcomplete* representations exist in dictionary learning (Lewicki and Sejnowski, 2000), i.e., $K > F$, leading to non-unique basis decompositions.

[3] Perhaps the choice of the word "positive", too ambiguous in a matrix context, prevented Paatero and its colleagues to be recognized as the true pioneers of NMF.

- In text information retrieval, $\mathbf{V}$ represents a corpus of N documents under the so-called bag-of-words representation. Given a vocabulary of F different words, the matrix entry v_{fn} is the occurrence count of word f in document n. In an NMF decomposition, $\mathbf{W}$ represents the word distributions of K topics, and $\mathbf{H}$ the proportion of each topic in a document (Xu et al., 2003; Shahnaz et al., 2006).
- In hyperspectral imaging, $\mathbf{V}$ represents an hyperspectral image. Each column of $\mathbf{V}$ represents the pixel intensities of the image over a broad range of spectral bands (i.e., not limited to the RGB bands). In an NMF decomposition, $\mathbf{W}$ represents the spectral signature of materials (called the endmembers), and $\mathbf{H}$ represents the relative proportions (called the abundances). This is referred to as hyperspectral unmixing (Berry et al., 2007; Bioucas-Dias et al., 2012).

Other application fields include, to cite only a few: image processing (Li et al., 2001; Guillamet et al., 2003), collaborative filtering (Zhang et al., 2006), computational biology (Devarajan, 2008), or community detection (Wang et al., 2011).

1.2.2 Limitations

Despite all the benefits brought by NMF when analyzing non-negative data, several limitations are to be discussed.

NMF is ill-posed

NMF is an inherently ill-posed problem, since there always is an infinite number of solutions[4]. Indeed, consider ($\mathbf{W}^\star$,$\mathbf{H}^\star$) solution of the problem (1.8). If there exists invertible square matrices $\mathbf{Q} \in \mathbb{R}^{K\times K}$ such that:

$$\begin{cases} \mathbf{W}^\star\mathbf{Q} \geq 0, \\ \mathbf{Q}^{-1}\mathbf{H}^\star \geq 0, \end{cases} \tag{1.9}$$

then ($\mathbf{W}^\star\mathbf{Q}$, $\mathbf{Q}^{-1}\mathbf{H}^\star$) is also a solution of the problem (1.8).

As it turns out, $\mathbf{Q}$ can always be chosen to be a monomial matrix, that is the product of a permutation matrix and a diagonal matrix with positive entries (it is in fact an equivalence if $\mathbf{Q}$ is constrained to be non-negative (Minc, 1988)). However, this is not a problem in practice, since all the solutions are equivalent up to a scale and permutation indeterminacy in this scenario. This can be fixed by adding additional regularization terms in the objective function, e.g., on the scale of one of the factors.

The real problem arises when there exists non-monomial matrices $\mathbf{Q}$ such that the couple ($\mathbf{W}^\star\mathbf{Q}$, $\mathbf{Q}^{-1}\mathbf{H}^\star$) is also a solution of the problem (1.8). The two solutions then lead to

[4] When admitting the existence of at least one solution to the problem (1.8).

completely different interpretations. Below is a simple illustrative example:

$$\begin{bmatrix} 1 & 0.5 \\ 2 & 0.5 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.25 \\ 1.5 & 0 \\ 1 & 0.5 \end{bmatrix} \frac{1}{3} \begin{bmatrix} 1 & 4 & 5 \\ 4 & 4 & 8 \end{bmatrix}. \tag{1.10}$$

For more detailed considerations about uniqueness in NMF, the interested reader is referred to Donoho and Stodden (2003); Laurberg et al. (2008); Gillis (2012); Huang et al. (2014).

NMF is non-convex

NMF as in Eq. (1.8) is not a convex problem w.r.t. both $\mathbf{W}$ and $\mathbf{H}$. However, for certain choices of cost functions, the problem will be convex w.r.t. one variable, when keeping the other fixed. This gives the intuition behind most of the existing optimization algorithms to solve the problem.

NMF is NP-hard

In the scenario where $\mathbf{V}$ can be exactly factorized as $\mathbf{WH}$, NMF is NP-hard in general (Vavasis, 2009). This is in contrast with algorithms retrieving the SVD of a matrix, which have $\mathcal{O}(NF^2 + N^3)$ complexity.

1.2.3 Choice of the divergence

In this subsection, we detail some of the different measures of fit that have been used in the NMF literature. First of all, we assume the separability of the cost function w.r.t. F and N, that is

$$D(\mathbf{V}|\mathbf{WH}) = \sum_{f,n} d(v_{fn}|[\mathbf{WH}]_{fn}), \tag{1.11}$$

where d is a scalar divergence. The word *divergence*, commonly used in NMF, is to be understood as a mathematical object more general than a distance. In particular, a divergence almost never enforces two of the three conditions necessary to define a distance, namely the symmetry condition and the triangular inequality. Basically, a divergence must at least respect the following conditions:

- $d(x|y) \geq 0$,
- $d(x|y) = 0 \Leftrightarrow x = y$.

The seminal papers of Lee and Seung proposed the use of two different divergences, which to this day remain very popular in NMF:

- The (squared) Euclidian distance

$$d(x|y) = (x - y)^2. \tag{1.12}$$

Note that in this case $D(\mathbf{V}|\mathbf{WH})$ is equivalently the the squared Frobenius norm $||\mathbf{V} - \mathbf{WH}||_F^2$.

- The (generalized) Kullback-Leibler (KL) divergence[5]

$$d(x|y) = x \log\left(\frac{x}{y}\right) - x + y. \tag{1.13}$$

Several works have proposed alternative cost functions. Most of these works focus on the study of a parametrized family of divergences. We mention the Kompass family of divergences (Kompass, 2007), which interpolates between the KL divergence and the Euclidian distance with a parameter between 0 and 1. The family of the α-divergences was also considered (Cichocki et al., 2008), as well as the family of the β-divergences (Cichocki and Amari, 2010; Févotte and Idier, 2011), where both α or β are real parameters.

Even broader families have been considered, based on generating functions. We mention the well-known Bregman divergences (Sra and Dhillon, 2005), which can generate the β-divergences, and the Csiszar divergences (Cichocki et al., 2006), which can generate the α-divergences.

The links between all these families are recapped and further discussed in Appendix 1.A. We will however focus on the special case of the β-divergences family, which has received a particular attention in Févotte and Idier (2011).

The special case of the β-divergences family

The family of the β-divergences, parametrized by $\beta \in \mathbb{R}$, can be defined as:

$$d_\beta(x|y) = \begin{cases} \frac{1}{\beta(\beta-1)}\left(x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}\right) & \text{if } \beta \in \mathbb{R} \setminus \{0,1\}, \\ x \log\left(\frac{x}{y}\right) - x + y & \text{if } \beta = 1, \\ \frac{x}{y} - \log\left(\frac{x}{y}\right) - 1 & \text{if } \beta = 0. \end{cases} \tag{1.14}$$

This family takes as particular cases the Euclidian distance ($\beta = 2$), the KL divergence ($\beta = 1$), as well as the Itakura-Saito divergence ($\beta = 0$), a divergence broadly used in audio signal processing. Therefore, the β-divergences family continuously generalizes the most commonly used divergences in NMF. Figure 1.2 displays the β-divergence for various values of β.

An interesting property of the β-divergence is the following:

$$\forall \lambda > 0,\ d_\beta(\lambda x|\lambda y) = \lambda^\beta d_\beta(x|y). \tag{1.15}$$

Consequently, the largest values will have more importance in the cost function with $\beta > 0$, whereas with $\beta < 0$, the smallest values will have more importance. The only scale-invariant

[5] Of course, the KL divergence does not represent a divergence between probability distributions in this context, hence the word "generalized".

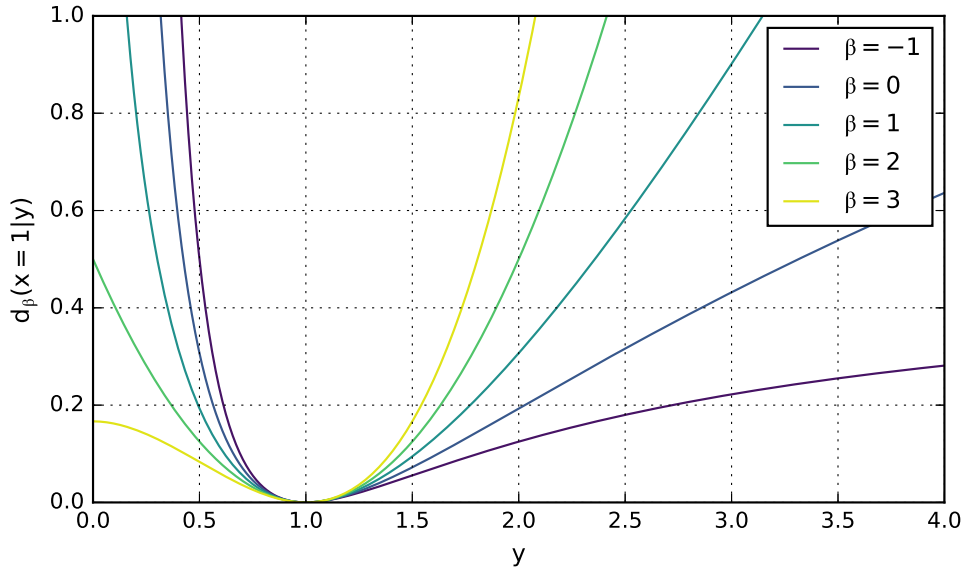


Figure 1.2: The family of the β-divergences. Each curve represents the β-divergence as a function of y, with the value of x set to 1, for five different values of β.

distribution is obtained for $\beta = 0$, i.e., the Itakura-Saito (IS) divergence. This gives us some insight regarding the choice of the divergence to use. For example, the scale invariance property of the IS divergence is one of the main reasons for its advocacy in audio signal processing (Févotte et al., 2009).

Finally, note that the first-order and second-order derivatives are continuous in β, which allows for unified optimization techniques for the whole family, as shall be discussed in the following subsection.

1.2.4 Standard algorithms

The very large majority of algorithms designed to solve the NMF problem described in Eq. (1.8) makes use of a block coordinate descent scheme. More precisely, the algorithm consists in alternatively updating one of the factors, while keeping the other fixed. The general framework of such algorithms is outlined in Algorithm 1.

The algorithm therefore boils down to the update of $\mathbf{W}$ given $\mathbf{H}$, and that of $\mathbf{H}$ given $\mathbf{W}$. As it turns out, these two steps are symmetric, since we have by transposition

$$\mathbf{V} \simeq \mathbf{WH} \Leftrightarrow \mathbf{V}^{\mathrm{T}} \simeq \mathbf{H}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}. \tag{1.16}$$

As such, we can focus on how to solve one of these two tasks only, for instance

$$\min_{\mathbf{W} \geq 0} D(\mathbf{V}|\mathbf{WH}^{(i-1)}). \tag{1.17}$$

Algorithm 1: Standard block coordinate descend (BCD) algorithm for NMF

Input: Non-negative matrix $\mathbf{V}$, factorization rank K

Generate random initial non-negative matrices $\mathbf{W}^{(0)}$ and $\mathbf{H}^{(0)}$

for $i = 1, \dots, N_{iter}$ **do**

  update $\mathbf{W}$ such that $D(\mathbf{V}|\mathbf{W}^{(i)}\mathbf{H}^{(i-1)}) \leq D(\mathbf{V}|\mathbf{W}^{(i-1)}\mathbf{H}^{(i-1)})$

  update $\mathbf{H}$ such that $D(\mathbf{V}|\mathbf{W}^{(i)}\mathbf{H}^{(i)}) \leq D(\mathbf{V}|\mathbf{W}^{(i)}\mathbf{H}^{(i-1)})$

end

Output: $\mathbf{WH}$, rank-K approximation of $\mathbf{V}$

The best case scenario arises when the divergence D is convex w.r.t. $\mathbf{W}$ (while keeping $\mathbf{H}$ fixed). This is for example the case for the β-divergences family when $\beta \in [1, 2]$. As such, the optimal solution to the sub-problem (1.17) is usually sought after. This has been extensively studied when D is chosen to be the squared Euclidian distance. We mention the projected gradient descent (Lin, 2007), as well as the numerous works on the "alternating non-negative least squares" family developed by Haesun Park and colleagues (Kim and Park, 2008, 2011). For a more thorough survey of these methods, as well as comparative considerations, the interested reader is referred to Gillis and Glineur (2012) and Kim et al. (2014).

In the general case, we may resort to majorization-minimization (MM) (Hunter and Lange, 2004). MM techniques consist in majorizing the objective function (by a so-called auxiliary function), and then optimizing the auxiliary function instead. The auxiliary function is usually chosen such that its optimization is easier than the optimization of the original objective function, e.g., by choosing a convex auxiliary function. More details about the MM framework can be found in Appendix 1.B. By construction, we retrieve a $\mathbf{W}^{(i)}$ that decreases the objective function. MM-based algorithms have been studied for the β-divergences family in Févotte and Idier (2011). It results in very simple and elegant multiplicative updates, which corresponds to the original heuristic update rules proposed by Lee and Seung for the KL divergence and Euclidian distance:

$$\mathbf{W} \leftarrow \mathbf{W} \odot \left(\frac{[(\mathbf{WH})^{\beta-2} \odot \mathbf{V}]\mathbf{H}^{\mathrm{T}}}{(\mathbf{WH})^{\beta-1}\mathbf{H}^{\mathrm{T}}} \right)^{\gamma(\beta)}, \tag{1.18}$$

$$\mathbf{H} \leftarrow \mathbf{H} \odot \left(\frac{\mathbf{W}^{\mathrm{T}}[(\mathbf{WH})^{\beta-2} \odot \mathbf{V}]}{\mathbf{W}^{\mathrm{T}}(\mathbf{WH})^{\beta-1}} \right)^{\gamma(\beta)}, \tag{1.19}$$

where $\odot$ denotes the Hadamard product; division and power are taken entry-wise. The exponent $\gamma(\beta)$ is such that

$$\gamma(\beta) = \begin{cases} \frac{1}{2-\beta} & \text{if } \beta < 1, \\ 1 & \text{if } 1 \leq \beta \leq 2, \\ \frac{1}{\beta-1} & \text{if } \beta > 2. \end{cases} \tag{1.20}$$

For considerations regarding the convergence of these MM-based algorithms (i.e., whether

Figure 1.3: $K = 30$ atoms of the dictionary learned on the CBCL dataset.

the sequence of iterates $\{\mathbf{W}^{(i)}, \mathbf{H}^{(i)}\}_{i\geq 0}$ converges to a stationary point satisfying the Karush-Kuhn-Tucker (KKT) conditions), we refer the reader to Zhao and Tan (2018).

1.2.5 An example: the original experiment

We here present for illustrations purposes the experiment described in the seminal paper of Lee and Seung (Lee and Seung, 1999). We similarly consider the CBCL[6] dataset, which contains 2429 grayscale images of faces of size 19×19. This yields an observation matrix $\mathbf{V}$ of size 361×2429.

We then seek to find an NMF of $\mathbf{V}$ with $K = 30$. The Kullback-Leibler divergence was chosen to be the measure of fit, and we use the standard multiplicative updates (i.e., Eqs (1.18)-(1.19) with $\beta = 1$). The algorithm is run for 5000 iterations.

Each image of Figure 1.3 displays one column of the dictionary $\mathbf{W}$. As we can see, they highlight different parts of faces. Figure 1.4 displays the reconstructed image for eight samples chosen at random.

[6]Retrieved from http://www.ai.mit.edu/courses/6.899/lectures/faces.tar.gz.

Figure 1.4: Eight faces of the CBCL dataset chosen at random. Top: original images. Bottom: reconstruction with $K = 30$.

1.3 Probabilistic non-negative factorization

1.3.1 Definition

In the previous section, we have described NMF as an optimization problem, with one of the main questions being the choice of the measure of fit to assess the dissimilarity between $\mathbf{V}$ and its approximation $\mathbf{WH}$. We will now turn to an alternative paradigm, namely *probabilistic* non-negative matrix factorization.

As it turns out, for many usual cost functions, the minimization problem described in Eq. (1.8) can be shown to be equivalent to the joint maximum likelihood estimation of the factors $\mathbf{W}$ and $\mathbf{H}$ under a specific statistical model, that is

$$\max_{\mathbf{W},\mathbf{H}} \log p(\mathbf{V}; \mathbf{W}, \mathbf{H}). \tag{1.21}$$

We give the following illustrative example. Consider the following statistical model (assuming independence of the v_{fn}):

$$v_{fn} \sim \text{Poisson}([\mathbf{WH}]_{fn}), \tag{1.22}$$

with $\mathbf{W}, \mathbf{H} \geq 0$. We have

$$-\log p(\mathbf{V}; \mathbf{W}, \mathbf{H}) = -\sum_{f,n} \left(v_{fn} \log([\mathbf{WH}]_{fn}) - [\mathbf{WH}]_{fn} - \log(v_{fn}!)\right) \tag{1.23}$$

$$\stackrel{\text{c}}{=} \sum_{f,n} \left(v_{fn} \log \frac{1}{[\mathbf{WH}]_{fn}} + [\mathbf{WH}]_{fn}\right) \tag{1.24}$$

$$\stackrel{\text{c}}{=} D_{\text{KL}}(\mathbf{V}|\mathbf{WH}). \tag{1.25}$$

Therefore, maximizing the log-likelihood w.r.t. $\mathbf{W}$ and $\mathbf{H}$ in the model of Eq. (1.22) is equivalent to minimizing the KL-divergence between $\mathbf{V}$ and $\mathbf{WH}$[7].

[7] Algorithmic equivalences also exist between KL-NMF and probabilistic latent semantic analysis (PLSA), a document clustering model (Gaussier and Goutte, 2005).

This leads the way to so-called *probabilistic* NMF, i.e., learning (in a broad sense, meaning either estimation or inference problems) in probabilistic models whose observation density (likelihood) can be written as

$$\mathbf{v}_n \sim p(. \,; \mathbf{W}\mathbf{h}_n, \mathbf{\Psi}), \quad \mathbf{W} \geq 0, \quad \mathbf{h}_n \geq 0, \tag{1.26}$$

that is to say that the distribution of $\mathbf{v}_n$ is parametrized by the dot product between $\mathbf{W}$ and $\mathbf{h}_n$; any other parameters that could act on the distribution are generically denoted by $\mathbf{\Psi}$. Such models are therefore latent variable models.

This general framework encompasses many well-known latent variable models from the NMF literature. A tentatively exhaustive list is presented in the following subsection.

1.3.2 List of models

In the following list of models, the independence of the v_{fn} or of the $\mathbf{v}_n$ is implied and is not recalled in the equations.

- Models based on a Gaussian likelihood. Schmidt et al. (2009) assume the following likelihood for non-negative data

$$v_{fn} \sim \mathcal{N}([\mathbf{WH}]_{fn}, \sigma^2). \tag{1.27}$$

  The joint maximum likelihood estimation of the factors in this model amounts to minimizing the squared Euclidian distance between $\mathbf{V}$ and $\mathbf{WH}$. As such, it is the probabilistic counterpart to the classical quadratic loss choice. However, this model can be criticized, most notably because it can give rise to negative data, which contradicts the nature of the data at hand. Moreover, it non-negatively factorizes the mean of a Gaussian distribution, an unconstrained parameter, which may be unnatural in some cases. More general models with zero-mean noise have been considered in Alquier and Guedj (2017).

- Models based on an exponential likelihood (Févotte et al., 2009; Hoffman et al., 2010). They assume the following likelihood for non-negative data

$$v_{fn} \sim \text{Exp}\left(\frac{1}{[\mathbf{WH}]_{fn}}\right). \tag{1.28}$$

  The joint maximum likelihood of the factors in this model is equivalent to minimizing the Itakura-Saito divergence between $\mathbf{V}$ and $\mathbf{WH}$. As a matter of fact, this model can equivalently be rewritten as a composite model with complex Gaussian components, therefore principled when considering the STFT of an audio signal. These links will be further developed in Chapter 3.

- Models based on the Poisson distribution (Canny, 2004; Cemgil, 2009; Zhou et al., 2012; Gopalan et al., 2015). These models are sometimes generically referred to as "Poisson factorization", or "Poisson factor analysis". They assume

$$v_{fn} \sim \text{Poisson}([\mathbf{WH}]_{fn}). \tag{1.29}$$

As detailed in the previous subsection, the joint maximum likelihood estimation of the factors is tantamount to the minimization of the KL divergence between $\mathbf{V}$ and $\mathbf{WH}$.

- Models based on compound Poisson distributions. As it turns out, the use of the Poisson distribution does not restrict us to integer data, when considering the larger family of the compound Poisson distributions. Such models can be written as

$$L_{fn} \sim \text{Poisson}([\mathbf{WH}]_{fn}), \tag{1.30}$$

$$x_{lfn} \overset{\text{i.i.d.}}{\sim} p(.\,; \mathbf{\Psi}) \quad \forall l \in \{1, \dots, L_{fn}\}, \tag{1.31}$$

$$v_{fn} = \sum_{l=1}^{L_{fn}} x_{lfn}. \tag{1.32}$$

  In this case, p is called the element distribution, and depending on the choice of this distribution, gives rise to various supports for v_{fn} ($\mathbb{N}, \mathbb{R}_+, \mathbb{R}$...). These models (Şimşekli et al., 2013; Basbug and Engelhardt, 2016; Gouvert et al., 2019) are examples of hierarchical models in which the distribution of v_{fn} is only known conditionally to another random variable (in this case, L). It falls into the framework of Eq. (1.26) once this variable has been marginalized, which is most of the time not analytically possible.

- Models with more restrictive constraints on the factors. We mention the ubiquitous latent Dirichlet allocation (LDA) model (Blei et al., 2003), which assume

$$\mathbf{v}_n \sim \text{Mult}(L, \mathbf{Wh}_n), \tag{1.33}$$

  where Mult denotes the multinomial distribution. Therefore, we must have $\sum_f \mathbf{Wh}_n = 1$, which can be achieved by assuming that both the columns of $\mathbf{W}$ and $\mathbf{H}$ sum to 1. We also mention models for binary data, based on a Bernoulli likelihood (Kabán and Bingham, 2008; Bingham et al., 2009; Lumbreras et al., 2018)

$$v_{fn} \sim \text{Bernoulli}([\mathbf{WH}]_{fn}). \tag{1.34}$$

  Similarly, to ensure a valid Bernoulli parameter, $\mathbf{W}$ and $\mathbf{H}$ must be such that $\sum_k w_{fk} h_{kn} \in [0, 1]$.

- Models based on distributions with heavy tails. We mention for count data the negative binomial distribution (Gouvert et al., 2018; Zhou, 2018), and for continuous non-negative data the Student-t distribution (Yoshii et al., 2016) or the Lévy distribution (Magron et al., 2017).

Exponential dispersion models and Tweedie distributions

The exponential dispersion models (EDM) (Jørgensen, 1987; Jørgensen, 1997) are a two-parameter family of distributions, whose distribution can be written as

$$p(x; \theta, \varphi) = h(x, \varphi) \exp\left(\frac{1}{\varphi}(\theta x - \kappa(\theta))\right), \tag{1.35}$$

where $\theta \in \mathcal{D} \subset \mathbb{R}$ is the natural parameter, and $\varphi > 0$ is the dispersion parameter. The function h is called the base function, and κ is called the cumulant function. As it turns out, we have $\mu = \mathbb{E}(X) = \kappa'(\theta)$ and $\text{var}(X) = \varphi\kappa''(\theta)$ (see Appendix 1.C). The mapping between θ and μ being invertible, we may as well write $\text{var}(X) = \varphi V(\mu)$, where $V(\mu)$ is called the variance function. An EDM is characterized by its variance function.

The Tweedie family of distributions (Tweedie, 1984) assume that the variance function of the EDM is a power variance function, that is

$$V(\mu) = \mu^p. \tag{1.36}$$

It can be shown that such EDMs exist for $p \in \mathbb{R}\backslash]0, 1[$. Special cases are the normal distribution ($p = 0$), the Poisson distribution ($p = 1$), the Gamma distribution ($p = 2$), and the inverse Gaussian distribution ($p = 3$). More precisely, apart from the four aforementioned cases, a closed-form analytical expression of the distribution does not exist. The cumulant function κ can be derived in this case, but this does not imply a closed-form expression of h[8].

Setting $\beta = 2-p$, and assuming that the v_{fn} are independent and distributed as a Tweedie with mean $[\mathbf{WH}]_{fn}$, it can be shown that (Yılmaz and Cemgil, 2012; Tan and Févotte, 2013)

$$-\log p(\mathbf{V}; \mathbf{W}, \mathbf{H}) \overset{c}{=} \frac{1}{\varphi} \sum_{f,n} d_\beta(v_{fn}|[\mathbf{WH}]_{fn}). \tag{1.37}$$

Thus, the choice of β when considering NMF with the β-divergence (described in Section 1.2.3) underlies the choice of a noise model that can be described with Tweedie distributions, except for interval $\beta \in]1, 2[$ where no such model exists.

1.3.3 Model variants and learning problems

As explained in Section 1.3.1, a probabilistic NMF model is only defined by its associated observation model (the distribution of $\mathbf{v}_n$, Eq. (1.26)). As such, several variants can be considered

1. Frequentist NMF models. Graphical model described in Figure 1.5-a. The factors $\mathbf{W}$ and $\mathbf{H}$ are treated as deterministic parameters. Learning tasks in these models therefore correspond to maximum likelihood estimation. As explained previously, this amounts to optimizing a certain divergence between $\mathbf{V}$ and $\mathbf{WH}$. However, casting the problem as a maximum likelihood problem may enable us to use specific tools to solve this optimization task, namely the EM algorithm.
2. Bayesian NMF models. Graphical model described in Figure 1.5-b. In this case, the factors $\mathbf{W}$ and $\mathbf{H}$ are treated as random variables with prior distributions. As a matter of fact, the vast majority of the aforementioned works consider Bayesian NMF models. Inference resolves here around the joint posterior distribution $p(\mathbf{W}, \mathbf{H}|\mathbf{V})$.

[8] Numerical schemes exist, see for example Dunn and Smyth (2005).

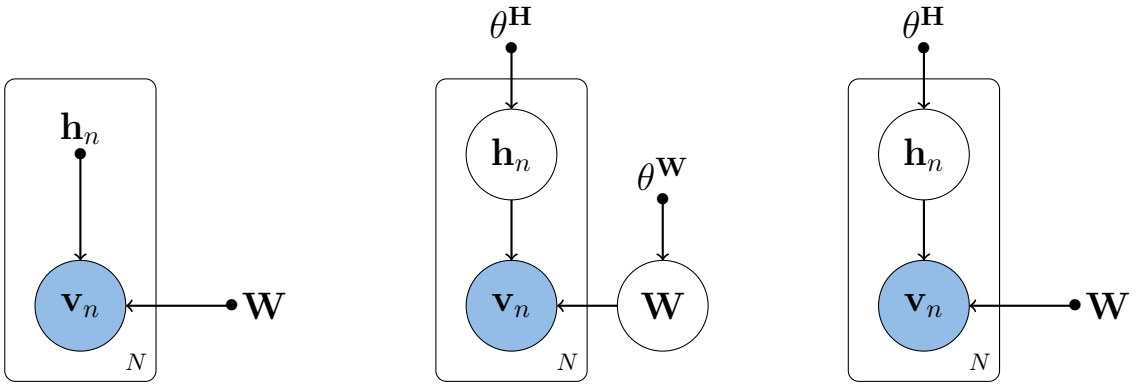


Figure 1.5: Probabilistic NMF models. From left to right. (a) Frequentist NMF models. Neither $\mathbf{W}$ nor $\mathbf{H}$ is assumed to be a random variable. (b) Bayesian NMF models. Both $\mathbf{W}$ and $\mathbf{H}$ are assumed to be random variables with a prior distribution. (c) Semi-Bayesian NMF models. $\mathbf{W}$ is assumed to be a parameter, whereas a prior distribution is assumed on $\mathbf{H}$.

3. Semi-Bayesian NMF models. Graphical model described in Figure 1.5-c. A third and last class of models, which we coin semi-Bayesian NMF models, can be considered. In these models, a prior distribution is assumed on $\mathbf{H}$, but $\mathbf{W}$ remains a deterministic variable. These models will be the specific focus of this thesis.

1.4 Inference in semi-Bayesian NMF

1.4.1 Estimators

We consider from now on semi-Bayesian NMF models, that is models defined by a prior distribution for $\mathbf{H}$, and an observation density of $\mathbf{v_n}$ given $\mathbf{h_n}$ (as in Eq. (1.26)). They have been considered in Dikmen and Févotte (2011) for an exponential likelihood (Eq. (1.28)) and in Dikmen and Févotte (2012) for a Poisson likelihood (Eq. (1.29)).

Denote by $\theta^{\mathbf{H}}$ the hyperparameters of the prior distribution on $\mathbf{H}$, $p(\mathbf{H};\theta^{\mathbf{H}})$. In these two papers, two estimation approaches were compared:

1. Maximizing the joint likelihood $p(\mathbf{V},\mathbf{H};\mathbf{W},\theta^{\mathbf{H}})$, that is:

$$\max_{\mathbf{W},\mathbf{H},\theta^{\mathbf{H}}} \log p(\mathbf{V},\mathbf{H};\mathbf{W},\theta^{\mathbf{H}}) = \log p(\mathbf{V}|\mathbf{H};\mathbf{W}) + \log p(\mathbf{H};\theta^{\mathbf{H}}). \tag{1.38}$$

   We refer to this process as maximum joint likelihood estimation (MJLE).

2. Maximizing the marginal likelihood $p(\mathbf{V};\mathbf{W})$, that is when $\mathbf{H}$ has been integrated out of the joint likelihood:

$$\max_{\mathbf{W},\theta^{\mathbf{H}}} \log p(\mathbf{V};\mathbf{W},\theta^{\mathbf{H}}) = \log \int_{\mathbf{H}} p(\mathbf{V}|\mathbf{H};\mathbf{W}) p(\mathbf{H};\theta^{\mathbf{H}}) \mathrm{d}\mathbf{H}. \tag{1.39}$$

We refer to this process as maximum marginal likelihood estimation (MMLE). Note that we may turn to the inference the posterior distribution $p(\mathbf{H}|\mathbf{V};\hat{\mathbf{W}})$ in a second step if necessary, where $\hat{\mathbf{W}}$ is the maximum marginal likelihood estimate. More generally, the methodology consisting first in estimating the hyperparameters (i.e., all non-random variables, meaning $\mathbf{W}$ and $\theta^{\mathbf{H}}$ in our setting) by maximizing the marginal likelihood (e.g., with an EM algorithm), and then in inferring the posterior distributions (e.g., by sampling) is referred to as *empirical Bayes* (EB) (Morris, 1983).

MMLE clearly constitutes a better-posed approach than MJLE from a statistical point of view. Assume that the hyperparameter $\theta^{\mathbf{H}}$ is of size L. In MJLE, the estimated variables are $\mathbf{W}, \mathbf{H}$ and $\theta^{\mathbf{H}}$. This represents a total number of $FK + KN + L$ estimated parameters, which grows with the number samples N. As such, little can be said about the statistical optimality the maximum likelihood estimator, which requires a fixed numbers of parameters w.r.t. the number of samples. In particular, this can lead to overfitting issues. This is in contrast with MMLE, where the number of estimated parameters is $FK + L$ (since only $\mathbf{W}$ and $\theta^{\mathbf{H}}$ are estimated), i.e., constant w.r.t. N.

In Dikmen and Févotte (2011, 2012), the comparison of these two methods was tackled empirically. In particular, algorithms to optimize both functions were conceived. In their experiments, on both synthetic (i.e., generated from the considered models) and real datasets, they consistently found that MMLE had a tendency to automatically regularize the factorization rank K. More precisely, the dictionaries estimated by MMLE have a tendency to exhibit columns with negligible norm, in contrast with those estimated by MJLE, which always make use of all the K columns.

This favorable behavior of MMLE was left unexplained. In particular, we would like to answer the following questions:

- Can we give an explanation to the observed self-regularization phenomenon in the specific settings of Dikmen and Févotte (2011, 2012)?
- Can we exhibit general conditions in which this phenomenon is bound to happen?
- Can we quantify the phenomenon?

1.4.2 Related works to MMLE

In this section, we make connections with other works of the literature, which maximize the marginal likelihood in different contexts.

1.4.2.1 Integrating out nuisance parameters

The methodology of MMLE bears strong links with the problem of eliminating nuisance parameters, a well-studied problem in statistics. A nuisance parameter is a parameter which takes part in the generative process, but whose value or distribution is not of immediate interest, or even of no interest at all. One might argue that in our NMF setting, the dictio-

nary $\mathbf{W}$ is the variable of interest, whereas the activation coefficients are not of immediate interest (recall that we may tackle the inference of $\mathbf{H}$ in a second step if necessary).

Berger et al. (1999) discusses the advantages and limitations of the elimination of nuisance parameters through integration (i.e., dealing with the marginal likelihood) versus methods that do not involve integration, especially methods that make use of the so-called profile likelihood.

1.4.2.2 A note on LDA

In the initial formulation of LDA (Blei et al., 2003), a Dirichlet prior is assumed on $\mathbf{h}_n$, while the dictionary $\mathbf{W}$ remains a deterministic variable. The authors then proposed an EB methodology, based on variational inference (Jordan et al., 1999; Blei et al., 2017) for both the estimation of $\mathbf{W}$ and $\theta^{\mathbf{H}}$ and the posterior inference. The inference in LDA is therefore MMLE.

Note that, as already proposed in Blei et al. (2003), LDA is now most of the time presented as a fully Bayesian model, by adding an additional Dirichlet prior on the columns of $\mathbf{W}$ (in this context, a column of $\mathbf{W}$ represents a distribution of words over a certain vocabulary, therefore must sum to one). In this case, the first step of the EB methodology amounts to estimating $\theta^{\mathbf{W}}$ and $\theta^{\mathbf{H}}$, the parameters of the two Dirichlet priors, by MMLE. See also a discussion of the importance of the hyperparameters in LDA in Wallach et al. (2009a).

1.4.2.3 A note on independent component analysis (ICA)

The concept of maximizing the marginal likelihood in semi-Bayesian NMF models is shared in spirit by the standard approach adopted in independent component analysis (ICA) (Comon, 1994; Hyvärinen et al., 2001).

Consider a collection of M signals of length T, stored into a matrix $\mathbf{X}$ of size $M \times T$. ICA assumes that these signals are an instantaneous linear mixture of M sources. More precisely, we can write for all time t

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t, \tag{1.40}$$

where $\mathbf{A} \in \mathbb{R}^{M \times M}$ is the mixture matrix. Moreover, the M sources are assumed to be independent

$$p(\mathbf{s}_t) = \prod_{m=1}^{M} p_m(s_{mt}), \tag{1.41}$$

where p_m denotes the distribution of source m. No assumption is made on the mixing matrix. The goal is then to estimate both the sources and the mixture matrix.

The standard way to solve the ICA problem is to maximize the marginal likelihood $p(\mathbf{X}; \mathbf{A})$, which can be easily obtained since $\mathbf{A}$ is square. The sources are then recovered by inverting the linear system of Eq. (1.40) using the maximum likelihood estimate of

$\mathbf{A}$. However, we would like to emphasize that in standard ICA, no noise model is assumed, meaning that $p(\mathbf{x}_t|\mathbf{s}_t)$ is a Dirac distribution, hence the ease to obtain the marginal likelihood (the marginal distribution of $\mathbf{x}_t$ is simply a change of variable). In the context of noisy ICA (i.e., where Eq. (1.40) becomes $\mathbf{x}_t = \mathbf{A}\mathbf{s}_t + \boldsymbol{\epsilon}_t$), maximizing the marginal likelihood has most notably been addressed in Moulines et al. (1997), under Gaussian assumptions.

1.4.3 Categories of priors on the activation coefficients

We conclude this section by taking a closer look at the prior distributions on $\mathbf{H}$ that will be considered in this thesis. We can sketch up two classes of priors. The first class corresponds to the standard independence assumption of the $\mathbf{h}_n$:

$$p(\mathbf{H}) = \prod_{n=1}^{N} p(\mathbf{h}_n). \tag{1.42}$$

The factors being non-negative, a standard choice for $p(\mathbf{h}_n)$ is for instance independent Gamma distributions, which can be sparsity-inducing if the shape parameter is lower than one. The inverse Gamma distribution has also been considered. The models tackled in Chapter 2 and 3 make use of this class of prior on $\mathbf{H}$.

One might also be interested in adding statistical correlation in the model, i.e., when the columns of $\mathbf{V}$ cannot be treated as exchangeable. Such a scenario arises in particular when the columns of $\mathbf{V}$ describe the evolution of a process over time (such matrices $\mathbf{V}$ are sometimes referred to as dynamic matrices). This is usually achieved by lifting the independence assumption of Eq. (1.42) in order to introduce correlation between successive columns of $\mathbf{H}$. In particular, we consider a Markov structure on the columns of $\mathbf{H}$:

$$p(\mathbf{H}) = p(\mathbf{h}_1) \prod_{n \geq 2} p(\mathbf{h}_n|\mathbf{h}_{n-1}). \tag{1.43}$$

This corresponds to the second class of priors studied in this thesis, and are the main concern of the models addressed in Chapter 4.

1.5 Structure of the manuscript and contributions

Chapter 2 tackles maximum marginal likelihood estimation in the Gamma-Poisson matrix factorization model. In particular, we derive a closed-form expression of the marginal likelihood, which gives us some insight into the self-regularization phenomenon empirically observed in Dikmen and Févotte (2012). Moreover, an experimental comparison of three EM algorithms is carried out.

Chapter 3 deals with maximum marginal likelihood estimation in a semi-Bayesian NMF model based on an exponential likelihood. This model is a special case of the models studied in Dikmen and Févotte (2011). Unlike the previous chapter, we are unable to derive

a closed-form expression of the marginal likelihood in this setting. Nonetheless, we derive three novel EM algorithms and apply them to a real audio decomposition example.

Chapter 4 addresses maximum marginal likelihood estimation in semi-Bayesian NMF models in which the prior distribution on $\mathbf{H}$ is as described in Eq. (1.43). We begin by thoroughly reviewing the literature on non-negative Markov chains. We then propose a novel NMF model, as well as its associated inference.

The concluding chapter, page 139, presents conclusions and discusses some perspectives of our work.

The Appendix A provides a substantial abstract of this thesis in French.

The Appendix B describes the results of two collaborations that were conducted concurrently to the work presented in this thesis. The first one deals with Bayesian mean-parametrized NMF models for binary data. The second one studies a ranking model combined with NMF, with applications to data from tennis tournaments.

Appendices to Chapter 1

Contents

1.A The divergences used in NMF **49**
1.A.1 Parametrized families 49
1.A.2 Families generated by a function 50
1.B The majorization-minimization framework **50**
1.C Exponential dispersion models and Tweedie distributions **52**

1.A The divergences used in NMF

1.A.1 Parametrized families

In this subsection, we discuss the classical parametrized families of divergences that have been used in NMF. We begin by mentioning the family of the α-divergences (Cichocki et al., 2008)

$$d_\alpha^a(x|y) = \begin{cases} \frac{1}{\alpha(1-\alpha)}\left(\alpha x + (1-\alpha)y - x^\alpha y^{1-\alpha}\right) & \text{if } \alpha \in \mathbb{R}\setminus\{0,1\} \\ y\log\left(\frac{y}{x}\right) + x - y & \text{if } \alpha = 0 \\ x\log\left(\frac{x}{y}\right) - x + y & \text{if } \alpha = 1 \end{cases} \tag{1.44}$$

as well as the family of the β-divergences (Cichocki and Amari, 2010; Févotte and Idier, 2011)

$$d_\beta^b(x|y) = \begin{cases} \frac{1}{\beta(\beta-1)}\left(x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}\right) & \text{if } \beta \in \mathbb{R}\setminus\{0,1\} \\ \frac{x}{y} - \log\left(\frac{x}{y}\right) - 1 & \text{if } \beta = 0 \\ x\log\left(\frac{x}{y}\right) - x + y & \text{if } \beta = 1 \end{cases} \tag{1.45}$$

The two families are distinct. However, they are connected with the simple relation

$$d_\beta^b(x|y) = y^{\beta-1} d_\beta^a(x|y). \tag{1.46}$$

A parametric family of divergences was independently studied by Kompass (Kompass,

2007)

$$d_\lambda^K(x|y) = \begin{cases} \frac{x}{\lambda}(x^\lambda - y^\lambda) + y^\lambda(y-x) & \text{if } \lambda \in]0,1] \\ \lambda \log\left(\frac{x}{y}\right) + x - y & \text{if } \lambda = 0 \end{cases} \tag{1.47}$$

As it turns out, this family corresponds to the family of the β-divergences up to a multiplicating factor:

$$d_\lambda^K(x|y) = \lambda d_{\lambda+1}^b(x|y). \tag{1.48}$$

Note that the families of the α-divergences and β-divergences can further be generalized to the so-called α, β-family of divergences (Cichocki et al., 2011).

1.A.2 Families generated by a function

In this subsection, we discuss families of divergences used in NMF that are based on a generating function.

1.A.2.1 Bregman divergences

Let $F : \mathbb{R}_+ \to \mathbb{R}$ be a continuously-differentiable, convex function. The Bregman divergence associated to F is defined as

$$d(x|y) = F(x) - F(y) + F'(y)(x-y). \tag{1.49}$$

By taking

$$F_\beta(x) = \frac{x^\beta}{\beta(\beta-1)} - \frac{x}{\beta-1} + \frac{1}{\beta}, \tag{1.50}$$

and the associated limit cases when $\beta \to 0$ or 1, we retrieve the family of the β-divergences.

1.A.2.2 Csiszar divergences

Let $\varphi : \mathbb{R}_+ \to \mathbb{R}$ be a convex function, continuous in zero.

$$d(x|y) = x\varphi\left(\frac{y}{x}\right). \tag{1.51}$$

An appropriate choice of φ can give rise to the family of the α-divergences.

1.B The majorization-minimization framework

We describe in this section the key elements of the majorization-minimization (MM) framework. Consider a real-valued function f that we aim to minimize. The direct optimization of f is assumed to be difficult.

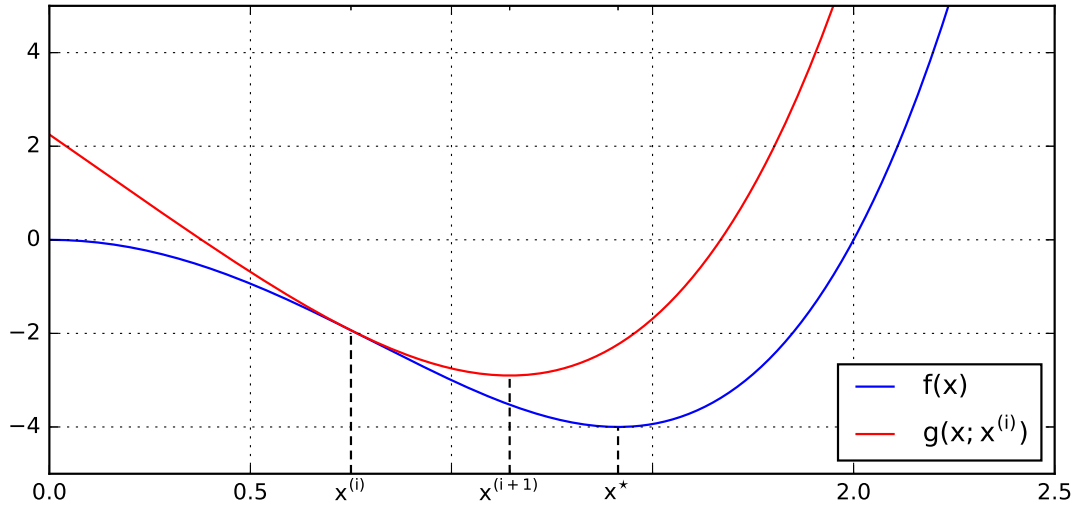


Figure 1.6: Illustrative example of an MM algorithm. The function f to minimize is displayed in blue. Its minimum is achieved at $x = x^\star$. Assuming that $x^{(i)} = 0.75$, we obtain the majorizing function in red. Its minimization yields the next iterate $x^{(i+1)}$.

MM algorithms iteratively construct a majorizing function of f, which we denote by g (the *majorization* step), and then proceed to the minimization of g (the *minimization* step). The function g, also called the auxiliary function, is constructed such that its minimization is easy, e.g. can be carried out in closed form.

Denote by $g(.;\tilde{x})$ a function whose shape depend on $\tilde{x}$. The function $g(.;\tilde{x})$ is said to be an auxiliary function to the function f if the two following properties hold

1. $\forall x,\ g(x;\tilde{x}) \geq f(x)$,
2. $g(\tilde{x};\tilde{x}) = f(\tilde{x})$.

In other words, $g(.;\tilde{x})$ is tangent to f at the point $x = \tilde{x}$, and above f elsewhere.

Consider now that the value of x at iteration i is $x^{(i)}$. The auxiliary function is constructed to be tangent at the point $x = x^{(i)}$, i.e., is denoted by $g(.;x^{(i)})$. Set $x^{(i+1)} = \operatorname{argmin} g(x;x^{(i)})$. Then we have

$$f(x^{(i+1)}) \leq g(x^{(i+1)};x^{(i)}) \tag{1.52}$$
$$\leq g(x^{(i)};x^{(i)}) = f(x^{(i)}). \tag{1.53}$$

Hence f is non-decreasing under the proposed procedure.

An illustrative example is given on Figure 1.6. The function to minimize is given by $f(x) = x^4 - 4x^2$. It is the sum of a convex function and a concave function. We majorize the concave part by its tangent at $x^{(i)}$ (the current iterate). Assuming that $x^{(i)} = 0.75$, we

obtain the auxiliary function $g(x) = x^4 - 6x + 0.75$, which we minimize to obtain the next iterate $x^{(i+1)}$.

1.C Exponential dispersion models and Tweedie distributions

We recall the distribution of an EDM

$$p(x;\theta,\varphi) = h(x,\varphi)\exp\left(\frac{1}{\varphi}(\theta x - \kappa(\theta))\right). \tag{1.54}$$

Integrating w.r.t. x and deriving w.r.t. θ we obtain

$$\int \frac{1}{\varphi}(x - \kappa'(\theta))p(x;\theta,\varphi)\mathrm{d}x = 0, \tag{1.55}$$

$$\frac{1}{\varphi}\left(\mathbb{E}(X) - \kappa'(\theta)\right) = 0. \tag{1.56}$$

Hence $\mu \overset{\text{def}}{=} \mathbb{E}(X) = \kappa'(\theta)$. Deriving Eq. (1.55) one more time w.r.t. θ yields

$$\int \frac{1}{\varphi}\left(-\kappa''(\theta)p(x;\theta,\varphi) + \frac{1}{\varphi}(x - \kappa'(\theta))^2 p(x;\theta,\varphi)\right)\mathrm{d}x = 0, \tag{1.57}$$

$$\frac{1}{\varphi}\left(-\kappa''(\theta) + \frac{1}{\varphi}\mathbb{E}((X - \mathbb{E}(X)^2)\right) = 0. \tag{1.58}$$

Hence $\mathrm{var}(X) = \phi\kappa''(\theta)$, and since μ and θ are in one-to-one mapping (Jørgensen, 1987), we may as well write $\mathrm{var}(X) = \varphi V(\mu)$, where V is the variance function.

In the case of a Tweedie EDM, we have $\mathrm{var}(X) = \mu^p$. Then, setting arbitrary integration constant to zero, we obtain

$$\theta = \begin{cases} \frac{\mu^{1-p}}{1-p} & \text{if } p \neq 1 \\ \log\mu & \text{if } p = 1, \end{cases} \tag{1.59}$$

and

$$\kappa(\theta) = \begin{cases} \frac{\mu^{2-p}}{2-p} & \text{if } p \neq 2 \\ \log\mu & \text{if } p = 2. \end{cases} \tag{1.60}$$

Note that $0 < p < 1$ do not correspond to an EDM (Jørgensen, 1987). Finally, substituting $\beta = 2 - p$, for all $\beta \neq 1, 2$ we obtain

$$p(x;\mu,\varphi) = h(x,\varphi)\exp\left(\frac{1}{\varphi}\left(x\frac{\mu^{\beta-1}}{\beta-1} - \frac{\mu^\beta}{\beta}\right)\right). \tag{1.61}$$

This is the expression given in Tan and Févotte (2013) and suffices to show the equivalence with the β-divergence (by taking the log of the likelihood ratio at $x = \mu$ and x).

List of Publications

International conference papers

- Filstroff et al. (2018)
  Filstroff, L., Lumbreras, A., and Févotte, C. (2018). Closed-form Marginal Likelihood in Gamma-Poisson Matrix Factorization. In *Proceedings of the International Conference of Machine Learning (ICML)*.
  Available on arXiv: https://arxiv.org/pdf/1801.01799.pdf

- Xia et al. (2019)
  Xia, R., Tan, V.Y.F., Filstroff, L., and Févotte, C. (2019). A Ranking Model Motivated by Nonnegative Matrix Factorization with Applications to Tennis Tournaments. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*.
  Available on arXiv: https://arxiv.org/pdf/1903.06500.pdf

Submitted journal papers

- Lumbreras et al. (2018)
  Lumbreras, A., Filstroff, L., and Févotte, C. (2018). Bayesian mean-parameterized nonnegative binary matrix factorization. *Submitted to Data Mining and Knowledge Discovery*.
  Preprint available on arXiv: https://arxiv.org/pdf/1812.06866.pdf

Chapter 2

Maximum Marginal Likelihood Estimation in the Gamma-Poisson Model

This chapter has been adapted from Filstroff et al. (2018)

Contents

2.1 Introduction **56**
2.2 The Gamma-Poisson model **57**
2.2.1 First formulation 57
2.2.2 Composite structure of the model 58
2.3 New formulations of GaP **58**
2.3.1 GaP as a composite negative multinomial model 58
2.3.2 GaP as a composite multinomial model 59
2.4 Closed-form marginal likelihood **60**
2.4.1 Analytical expression 60
2.4.2 Self-regularization 62
2.5 Optimization algorithms **63**
2.5.1 Expectation-Maximization 63
2.5.2 Monte Carlo E-step 64
2.5.3 M-step 65
2.6 Experimental work **67**
2.6.1 Comparison of the algorithms 67
2.6.2 Examples of the self-regularization phenomenon 74
2.7 Discussion **76**

2.1 Introduction

A wide variety of probabilistic models are based on the following observation model

$$v_{fn} \sim \text{Poisson}([\mathbf{WH}]_{fn}), \tag{2.1}$$

with $\mathbf{W} \geq 0$, $\mathbf{H} \geq 0$. Such models are called Poisson factorization models, or sometimes Poisson factor analysis. As discussed in Section 1.3, we may classify these models in two categories:

- Semi-Bayesian models. When independent Gamma priors are assumed on $\mathbf{H}$, and $\mathbf{W}$ is treated as a deterministic variable, we retrieve the so-called Gamma-Poisson model of Canny (2004) and Buntine and Jakulin (2006);
- Bayesian models, which assume prior distributions on both $\mathbf{W}$ and $\mathbf{H}$. Typical prior distributions include independent Gamma distributions, or column-wise Dirichlet distributions. These Bayesian models have found applications in image processing (Cemgil, 2009), as well as in recommender systems (Ma et al., 2011; Gopalan et al., 2015), where they have received a particular attention. Several works have also been devoted to non-parametric approaches to alleviate the problem of setting the factorization rank K (Zhou et al., 2012; Gopalan et al., 2014; Zhou and Carin, 2015).

In this chapter, we focus on maximum marginal likelihood estimation in the Gamma-Poisson model. This problem has been first been addressed in Dikmen and Févotte (2012). In their experiments, they found this estimation process to be robust to over-specified values of K; an intriguing behavior that was left unexplained. We provide the following contributions:

- We provide a closed-form expression of the marginal likelihood. This expression is tedious to compute for large F and K, as it involves combinatorial operations, but is workable for problems in small dimension.
- We show that the proposed closed-form expression reveals a penalization term on the columns of $\mathbf{W}$ that explains the "self-regularization" effect observed in Dikmen and Févotte (2012).
- We compare three variants of the EM algorithm, and experimentally show that the one based on the marginalization of $\mathbf{H}$ has favorable properties.

The rest of the chapter is organized as follows. Section 2.2 introduces the Gamma-Poisson model. In Section 2.3, we propose two novel formulations of the model in which $\mathbf{H}$ has been marginalized out. This leads to a closed-form expression of the marginal likelihood, discussed in Section 2.4. Three optimization algorithms are presented in Section 2.5, and experimental work is conducted in Section 2.6. We conclude by a general discussion in Section 2.7.

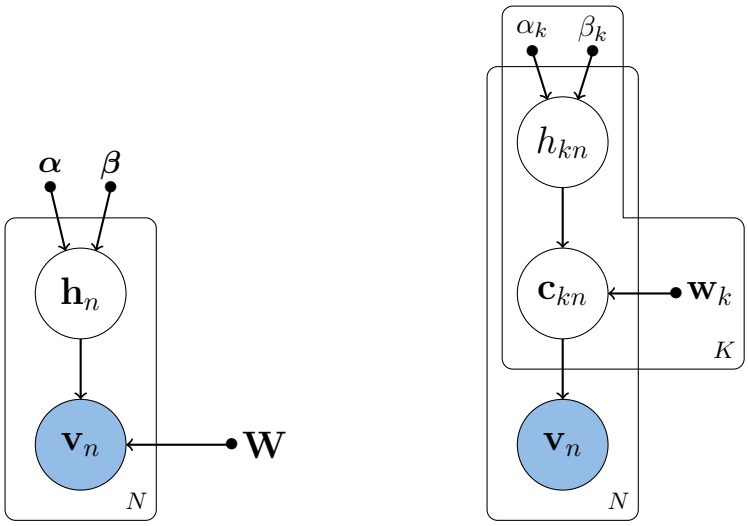


Figure 2.1: Graphical representations of the Gamma-Poisson model. Observed variables are in blue, while latent variables are in white. Deterministic parameters are represented as black dots. The observation $\mathbf{v}_n$ is a vector of size F. From left to right:(a) The standard model. The latent variable $\mathbf{h}_n$ is of size K; (b) The augmented model with variables $\mathbf{C}$. The latent variable $\mathbf{c}_{kn}$ is of size F, and h_{kn} is scalar.

2.2 The Gamma-Poisson model

2.2.1 First formulation

The Gamma-Poisson (GaP) model is a probabilistic matrix factorization model which was introduced in the field of text information retrieval (Canny, 2004; Buntine and Jakulin, 2006). In this field, a corpus of text documents is typically represented by an integer-valued matrix $\mathbf{V}$ of size $F \times N$, where each column $\mathbf{v}_n$ represents a document as a so-called "bag of words". Given a vocabulary of F words (or in practice semantic stems), the matrix entry v_{fn} is the number of occurrences of word f in the document n. GaP is a generative model described by a dictionary of "topics" or "patterns" $\mathbf{W}$ (a non-negative matrix of size $F \times K$) and a non-negative "activation" or "score" matrix $\mathbf{H}$ (of size $K \times N$), as follows:

$$h_{kn} \sim \text{Gamma}(\alpha_k, \beta_k), \tag{2.2}$$

$$v_{fn}|\mathbf{h}_n \sim \text{Poisson}\left([\mathbf{WH}]_{fn}\right), \tag{2.3}$$

where we use the shape and rate parametrization of the Gamma distribution, see its p.d.f. in Equation (0.8). The dictionary $\mathbf{W}$ is treated as a free deterministic variable.

We consider maximum marginal likelihood estimation (MMLE), in which $\mathbf{H}$ is treated as a latent variable over which the joint likelihood is integrated. In other words, MMLE relies on the minimization of

$$\mathcal{L}(\mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \stackrel{\text{def}}{=} -\log p(\mathbf{V}; \mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{2.4}$$

$$= -\log \int_{\mathbf{H}} p(\mathbf{V}|\mathbf{H}; \mathbf{W}) p(\mathbf{H}; \boldsymbol{\alpha}, \boldsymbol{\beta}) \mathrm{d}\mathbf{H}, \tag{2.5}$$

where $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_K\}$ and $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_K\}$. We emphasize that the dictionary $\mathbf{W}$ is treated as a free deterministic variable.

2.2.2 Composite structure of the model

GaP can be augmented with auxiliary variables $\mathbf{C}$, yielding a composite model, thanks to the superposition property of the Poisson distribution (Févotte et al., 2009):

$$h_{kn} \sim \text{Gamma}(\alpha_k, \beta_k), \tag{2.6}$$

$$c_{fkn}|h_{kn} \sim \text{Poisson}(w_{fk} h_{kn}), \tag{2.7}$$

$$v_{fn} = \sum_k c_{fkn}. \tag{2.8}$$

In the remainder, the vectors $\mathbf{c}_{kn} = [c_{1kn}, \ldots, c_{Fkn}]^{\mathrm{T}}$ of size F and which sum up to $\mathbf{v}_n$ will be referred to as *components*. $\mathbf{C}$ will denote the $F \times K \times N$ tensor with coefficients c_{fkn}. Figure 2.1 displays graphical representations of the Gamma-Poisson model, both in its initial form (on the left) and augmented form (on the right).

2.3 New formulations of GaP

We now show how GaP can be rewritten free of the latent variables $\mathbf{H}$ in two different ways.

2.3.1 GaP as a composite negative multinomial model

As it turns out, h_{kn} can be integrated out from Eqs. (2.6)-(2.7), thanks to the conjugacy between the Poisson and the Gamma distributions. That is to say that the marginal

distribution of $\mathbf{c}_{kn}$ can be determined in closed form. Indeed, we have

$$p(\mathbf{c}_{kn}; \mathbf{w}_k, \alpha_k, \beta_k) = \int_{\mathbb{R}_+} p(c_{1kn}, \ldots, c_{Fkn}|h_{kn}; \mathbf{w}_k) p(h_{kn}; \alpha_k, \beta_k) \mathrm{d}h_{kn} \tag{2.9}$$

$$= \int_{\mathbb{R}_+} \left(\prod_f p(c_{fkn}|h_{kn}; w_{fk}) \right) p(h_{kn}; \alpha_k, \beta_k) \mathrm{d}h_{kn} \tag{2.10}$$

$$= \int_{\mathbb{R}_+} \left(\prod_f \frac{(w_{fk}h_{kn})^{c_{fkn}}}{c_{fkn}!} \exp(-w_{fk}h_{kn}) \right) \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} h_{kn}^{\alpha_k - 1} \exp(-\beta_k h_{kn}) \mathrm{d}h_{kn} \tag{2.11}$$

$$= \left(\prod_f \frac{w_{fk}^{c_{fkn}}}{c_{fkn}!} \right) \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} \frac{\Gamma(\alpha_k + \sum_f c_{fkn})}{\left(\sum_f w_{fk} + \beta_k\right)^{\alpha_k + \sum_f c_{fkn}}} \tag{2.12}$$

$$= \frac{\Gamma(\alpha_k + \sum_f c_{fkn})}{\Gamma(\alpha_k) \prod_f c_{fkn}!} \left(\frac{\beta_k}{\sum_f w_{fk} + \beta_k} \right)^{\alpha_k} \prod_f \left(\frac{w_{fk}}{\sum_f w_{fk} + \beta_k} \right)^{c_{fkn}}. \tag{2.13}$$

More precisely, the distribution of $\mathbf{c}_{kn}$ can be identified to be a so-called negative multinomial (NM) distribution of dimension F, with shape parameter α_k and event probabilities $[p_{1k}, \ldots, p_{Fk}]^{\mathrm{T}}$, where

$$p_{fk} = \frac{w_{fk}}{\sum_{f'} w_{f'k} + \beta_k}. \tag{2.14}$$

The NM distribution is the multivariate extension of the perhaps more well-known negative binomial (NB) distribution, which arises in a scalar Gamma-Poisson mixture. The reader is referred to Appendix 2.A. for the definitions related to these probability distributions.

We therefore immediately obtain the following result:

Theorem 2.1. GaP can be rewritten as follows:

$$\mathbf{c}_{kn} \sim \mathrm{NM}\left(\alpha_k, \left[\frac{w_{1k}}{\sum_f w_{fk} + \beta_k}, \ldots, \frac{w_{Fk}}{\sum_f w_{fk} + \beta_k} \right]^{\mathrm{T}} \right), \tag{2.15}$$

$$\mathbf{v}_n = \sum_{k=1}^{K} \mathbf{c}_{kn}. \tag{2.16}$$

GaP may thus be interpreted as a composite model in which the k^{th} component has a NM distribution with parameters governed by $\mathbf{w}_k$ (the k^{th} column of $\mathbf{W}$), α_k and β_k.

2.3.2 GaP as a composite multinomial model

The NM distribution possesses an alternative characterization, namely a multinomial distribution whose number of trials is random. This immediately leads to the following

result:

Theorem 2.2. GaP can be rewritten as follows:

$$L_{kn} \sim \text{NB}\left(\alpha_k, \frac{\sum_f w_{fk}}{\sum_f w_{fk} + \beta_k}\right), \tag{2.17}$$

$$\mathbf{c}_{kn}|L_{kn} \sim \text{Mult}\left(L_{kn}, \left[\frac{w_{1k}}{\sum_f w_{fk}}, \ldots, \frac{w_{Fk}}{\sum_f w_{fk}}\right]^{\text{T}}\right), \tag{2.18}$$

$$\mathbf{v}_n = \sum_{k=1}^{K} \mathbf{c}_{kn}, \tag{2.19}$$

where "NB" denotes the negative binomial distribution, and "Mult" refers to the multinomial distribution.

Proof. See Appendix 2.A. □

Theorem 2.2 states that another interpretation of GaP consists in modeling the data as a sum of K independent multinomial distributions, governed individually by $\mathbf{w}_k$ and whose number of trials is random, following a NB distribution governed by $\mathbf{w}_k$, α_k and β_k.

A special case of the reformulation of GaP offered by Theorem 2.2 is given by Buntine and Jakulin (2006) using a different reasoning, when it is assumed that $\sum_f w_{fk} = 1$ (a common assumption in the field of text information retrieval, where the columns of $\mathbf{W}$ are interpreted as discrete probability distributions). Theorem 2.2 provides a more general result as it applies to any non-negative matrix $\mathbf{W}$.

2.4 Closed-form marginal likelihood

2.4.1 Analytical expression

Until now, it was assumed that the marginal likelihood in the GaP model was not available analytically. However, the new parametrization offered by Theorem 2.1 allows to obtain a computable analytical expression of the marginal likelihood $\mathcal{L}$. Denote by $\mathcal{C}$ the set of all "admissible" components, i.e.,

$$\mathcal{C} = \{\mathbf{C} \in \mathbb{N}^{F\times K\times N} \,|\, \forall (f,n), \sum_k c_{fkn} = v_{fn}\}. \tag{2.20}$$

By marginalization of $\mathbf{C}$, we may write

$$p(\mathbf{V}; \mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{\mathbf{C}\in\mathcal{C}} p(\mathbf{C}; \mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{2.21}$$

$$= \sum_{\mathbf{C}\in\mathcal{C}} \prod_{k,n} p(\mathbf{c}_{kn}; \mathbf{w}_k, \alpha_k, \beta_k). \tag{2.22}$$

As stated previously, $p(\mathbf{c}_{kn}; \mathbf{w}_k, \alpha_k, \beta_k)$ is now known in closed form, see Eq. (2.13). Replacing it with its expression, we obtain

$$p(\mathbf{V}; \mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{\mathbf{C} \in \mathcal{C}} \prod_{k,n} \left[\frac{\Gamma(\sum_f c_{fkn} + \alpha_k)}{\Gamma(\alpha_k) \prod_f c_{fkn}!} \left(\frac{\beta_k}{\sum_f w_{fk} + \beta_k} \right)^{\alpha_k} \prod_f \left(\frac{w_{fk}}{\sum_f w_{fk} + \beta_k} \right)^{c_{fkn}} \right]. \tag{2.23}$$

Introducing the notation

$$\Omega_{\boldsymbol{\alpha}}(\mathbf{C}) = \prod_{k,n} \frac{\Gamma(\sum_f c_{fkn} + \alpha_k)}{\Gamma(\alpha_k) \prod_f c_{fkn}!}, \tag{2.24}$$

and reusing the event probabilities p_{fk} defined in Eq. (2.14), we may rewrite Eq. (2.23) as

$$p(\mathbf{V}; \mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \left[\prod_k (1 - \sum_f p_{fk})^{N\alpha_k} \right] \times \sum_{\mathbf{C} \in \mathcal{C}} \left[\Omega_{\boldsymbol{\alpha}}(\mathbf{C}) \prod_{f,k} p_{fk}^{\sum_n c_{fkn}} \right]. \tag{2.25}$$

Equation (2.25) is a computable closed-form expression of the marginal likelihood. It is free of $\mathbf{H}$ and in particular of the integral that appears in Equation (2.5). However the expression (2.25) is still semi-explicit because it involves a sum over the set of all admissible components $\mathbf{C}$. $\mathcal{C}$ is a finite set with cardinality

$$\mathrm{card}(\mathcal{C}) = \prod_{f,n} \binom{v_{fn} + K - 1}{K - 1}, \tag{2.26}$$

see the result given in Appendix 2.B. It is straightforward to construct but challenging to compute in large dimension, as well as for large values of v_{fn}.

The sum over all the matrices in the set $\mathcal{C}$ expresses the convolution of the (discrete) probability distributions of the K components. Unfortunately, the distribution of the sum of independent negative multinomial variables of different event probabilities is not available in closed form.

As already stated in Dikmen and Févotte (2012), the value of the marginal likelihood is unchanged when the scales of the columns of $\mathbf{W}$ and the rates $\boldsymbol{\beta}$ are changed accordingly. Let $\boldsymbol{\Lambda}$ be a non-negative diagonal matrix of size K. It can easily be derived from Equation (2.25) that

$$p(\mathbf{V}; \mathbf{W}\boldsymbol{\Lambda}, \boldsymbol{\alpha}, \boldsymbol{\beta}\boldsymbol{\Lambda}) = p(\mathbf{V}; \mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}). \tag{2.27}$$

We therefore have a scaling invariance between $\mathbf{W}$ and $\boldsymbol{\beta}$, and as such, we may fix $\boldsymbol{\beta}$ to arbitrary values and leave $\mathbf{W}$ free. Thus, we will treat $\boldsymbol{\beta}$ as a constant in the following and drop it from the arguments of $\mathcal{L}$.

2.4.2 Self-regularization

Dikmen and Févotte (2012) empirically studied the properties of MMLE. In particular, they observed the self-ability of the estimator to regularize the number of columns of $\mathbf{W}$. For example, one experiment (reproduced in Section 2.6.2.1) consisted in generating synthetic data according to the GaP model, with a ground-truth number of components $K^\star$. MMLE was run with $K > K^\star$ and they noticed that the estimated $\mathbf{W}$ contained $K - K^\star$ empty columns. As such, the estimator was able to recover the ground-truth dimensionality. In contrast, MJLE used all K dimensions and overfit the data. They were unable to give a theoretical justification of the observed phenomenon, but provided a first insight thanks to a Laplace approximation of $p(\mathbf{V}; \mathbf{W}, \boldsymbol{\alpha})$. The closed-form expression (2.25) offers a deeper understanding of this phenomenon, as explained next.

Using Equations (2.25) and (2.14) and treating $\boldsymbol{\beta}$ as a constant, the negative log-likelihood can be expressed as

$$-\frac{1}{N}\mathcal{L}(\mathbf{W}, \boldsymbol{\alpha}) = -\frac{1}{N}\log\left(\sum_{\mathbf{C}\in\mathcal{C}} \Omega_{\boldsymbol{\alpha}}(\mathbf{C}) \prod_{f,k} p_{fk}^{\sum_n c_{fkn}}\right) \tag{2.28}$$

$$+ \sum_k \alpha_k \log(||\mathbf{w}_k||_1 + \beta_k) + \text{cst}, \tag{2.29}$$

where $\text{cst} = -\sum_k \alpha_k \log \beta_k$.

The negative log-likelihood reveals two terms. The first term, Equation (2.28), captures the interaction between data $\mathbf{V}$ (through $\mathbf{C}$) and the parameter $\mathbf{W}$ (through the event probabilities $p_{fk} = w_{fk}/(\|\mathbf{w}_k\|_1 + \beta_k)$). The second term, Equation (2.29), only depends on the parameter $\mathbf{W}$ and can be interpreted as a group-regularization term. The non-convex and sharply peaked function $f(\mathbf{x}) = \sum_k \log(|x_k| + \beta)$ is known to be sparsity-inducing (Candès et al., 2008). As such, the term (2.29) will promote sparsity of the norms of the columns of $\mathbf{W}$. When a norm $||\mathbf{w}_k||_1$ is set to zero for some k, the whole column $\mathbf{w}_k$ is set to zero because of the non-negativity constraint. This gives a formal explanation of the ability of MMLE to automatically prune columns of $\mathbf{W}$, without any explicit sparsity-inducing prior at the modeling stage (recall that $\mathbf{W}$ is a deterministic parameter without a prior).

The discussed penalty term, sometimes referred to as the "log-sum" penalty is displayed[1] on Figure 2.2, along with other common non-convex penalties (the ℓ_q pseudo-norm with $0 < q < 1$, and the smoothly clipped absolute deviation (SCAD) penalty (Fan and Li, 2001)).

[1] On Figure 2.2, a constant term $-\log(\beta)$ is added to the penalty to ensure $f(0) = 0$, as is the case with the other displayed penalties.

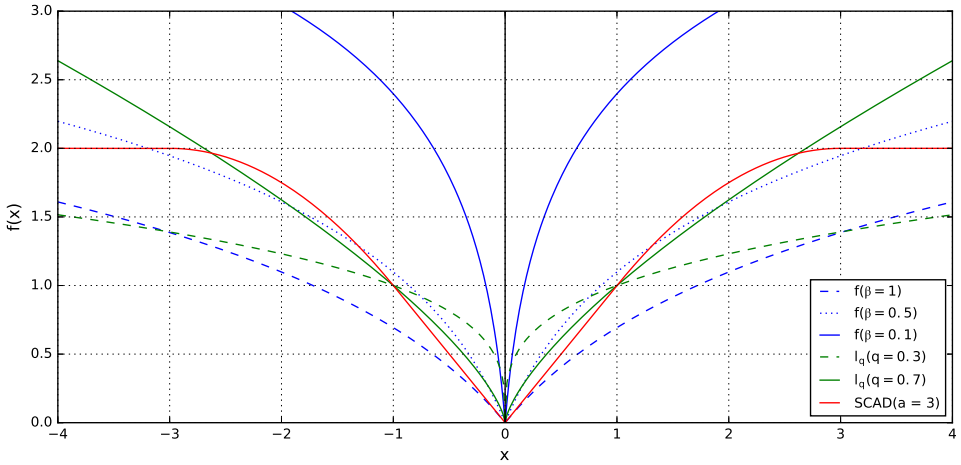


Figure 2.2: Common non-convex penalties represented in the scalar case. The log-sum penalty $f(x) = \log(|x| + \beta) - \log(\beta)$ is displayed in blue for two values of β. The ℓ_q pseudo-norm, defined as $g(x) = |x|^q$, is displayed in green for two values of q. The SCAD penalty is displayed in red with $a = 3$.

2.5 Optimization algorithms

2.5.1 Expectation-Maximization

We now turn to the problem of optimizing Equation (2.5) by leveraging on the results of Section 2.4. Despite obtaining a closed-form expression, the direct optimization of the marginal likelihood remains difficult. However, the structure of GaP makes Expectation-Maximization (EM) algorithms a natural option (Dempster et al., 1977). Indeed, the GaP model involves observed variables $\mathbf{V}$ and latent variables $\mathbf{C}$ and $\mathbf{H}$. As such, we can derive several EM algorithms based on various choices of the complete set. More precisely, we consider three possible choices that each define a different algorithm. In the following, we use the notation $\boldsymbol{\theta} = \{\mathbf{W}, \boldsymbol{\alpha}\}$.

EM-CH. The complete set is $\{\mathbf{C}, \mathbf{H}\}$ and EM consists in the iterative minimization w.r.t. $\boldsymbol{\theta}$ of the functional defined by

$$Q_{\mathrm{CH}}(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}) = -\int_{\mathbf{C},\mathbf{H}} \log p(\mathbf{C}, \mathbf{H}; \boldsymbol{\theta}) p(\mathbf{C}, \mathbf{H} | \mathbf{V}; \tilde{\boldsymbol{\theta}}) \mathrm{d}\mathbf{C}\mathrm{d}\mathbf{H}, \tag{2.30}$$

where $\tilde{\boldsymbol{\theta}} = \{\tilde{\mathbf{W}}, \tilde{\boldsymbol{\alpha}}\}$ is the current estimate. Note that $\mathbf{V}$ does not need to be included in the complete set because we have $\mathbf{V} = \sum_k \mathbf{C}_k$. This corresponds to the general formulation of EM in which the relation between the complete set and the data is a many-to-one mapping and slightly differs from the more usual one where the complete set is formed by the union of data and a hidden set (Dempster et al., 1977).

EM-H. The complete set is $\{\mathbf{V}, \mathbf{H}\}$ and EM consists in the iterative minimization of

$$Q_{\mathrm{H}}(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}) = -\int_{\mathbf{H}} \log p(\mathbf{V}, \mathbf{H}; \boldsymbol{\theta}) p(\mathbf{H}|\mathbf{V}; \tilde{\boldsymbol{\theta}}) \mathrm{d}\mathbf{H}. \tag{2.31}$$

EM-C. The complete set is merely $\{\mathbf{C}\}$ and EM consists in the iterative minimization of

$$Q_{\mathrm{C}}(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}) = -\int_{\mathbf{C}} \log p(\mathbf{C}; \boldsymbol{\theta}) p(\mathbf{C}|\mathbf{V}; \tilde{\boldsymbol{\theta}}) \mathrm{d}\mathbf{C}. \tag{2.32}$$

EM-CH and EM-H have been considered in Dikmen and Févotte (2012). EM-C is a new proposal that exploits the results of Section 2.4. In all three cases, the posteriors of the latent variables involved – $p(\mathbf{C}, \mathbf{H}|\mathbf{V}; \tilde{\boldsymbol{\theta}})$, $p(\mathbf{H}|\mathbf{V}; \tilde{\boldsymbol{\theta}})$ and $p(\mathbf{C}|\mathbf{V}; \tilde{\boldsymbol{\theta}})$ – are intractable and neither are the integrals involved in Equations (2.30), (2.31) and (2.32). To overcome this problem, we resort to Monte Carlo EM (MCEM) (Wei and Tanner, 1990) as described in the next section.

2.5.2 Monte Carlo E-step

MCEM consists in using a Monte Carlo (MC) approximation of the integrals in Equations (2.30), (2.31) and (2.32) based on samples drawn from the posterior distributions $p(\mathbf{C}, \mathbf{H}|\mathbf{V}; \tilde{\boldsymbol{\theta}})$, $p(\mathbf{H}|\mathbf{V}; \tilde{\boldsymbol{\theta}})$ and $p(\mathbf{C}|\mathbf{V}; \tilde{\boldsymbol{\theta}})$. These can be obtained by Gibbs sampling of the joint posterior $p(\mathbf{C}, \mathbf{H}|\mathbf{V}; \tilde{\boldsymbol{\theta}})$, which also returns samples from the marginals $p(\mathbf{H}|\mathbf{V}; \tilde{\boldsymbol{\theta}})$ and $p(\mathbf{C}|\mathbf{V}; \tilde{\boldsymbol{\theta}})$ at convergence. This Gibbs sampler can easily be derived because the conditional distributions $p(\mathbf{C}|\mathbf{H}, \mathbf{V}; \tilde{\boldsymbol{\theta}})$ and $p(\mathbf{H}|\mathbf{C}, \mathbf{V}; \tilde{\boldsymbol{\theta}}) = p(\mathbf{H}|\mathbf{C}; \tilde{\boldsymbol{\theta}})$ are available in closed form.

In particular, denoting $\underline{\mathbf{c}}_{fn}$ the vector $[c_{f1n}, \ldots, c_{fKn}]^{\mathrm{T}}$ of size K, we have for the first conditional[2]

$$p(\mathbf{C}|\mathbf{H}, \mathbf{V}; \tilde{\boldsymbol{\theta}}) = \prod_{f,n} p(\underline{\mathbf{c}}_{fn}|\mathbf{h}_n, v_{fn}; \underline{\tilde{\mathbf{w}}}_f), \tag{2.33}$$

where

$$p(\underline{\mathbf{c}}_{fn}|\mathbf{h}_n, v_{fn}; \underline{\tilde{\mathbf{w}}}_f) = \mathrm{Mult}\left(v_{fn}, [\tilde{\rho}_{f1n}, \ \ldots, \tilde{\rho}_{fKn}]^{\mathrm{T}}\right), \tag{2.34}$$

with the notation

$$\tilde{\rho}_{fkn} = \frac{\tilde{w}_{fk} h_{kn}}{[\tilde{\mathbf{W}}\mathbf{H}]_{fn}}. \tag{2.35}$$

For the second conditional, we have

$$p(\mathbf{H}|\mathbf{C}; \tilde{\boldsymbol{\theta}}) = \prod_{k,n} p(h_{kn}|\mathbf{c}_{kn}; \tilde{\alpha}_k, \tilde{\mathbf{w}}_k), \tag{2.36}$$

[2] We recall that the notation $\mathbf{h}_n$ refers to the n-th column of $\mathbf{H}$, whereas the notation $\underline{\mathbf{w}}_f$ refers to the f-th row of $\mathbf{W}$.

where

$$p(h_{kn}|\mathbf{c}_{kn};\tilde{\alpha}_k,\tilde{\mathbf{w}}_k) = \text{Gamma}\left(\tilde{\alpha}_k + \sum_f c_{fkn},\ \beta_k + \sum_f \tilde{w}_{fk}\right). \tag{2.37}$$

See Appendix 2.C for derivation details. The Gibbs sampler is thus summarized in Algorithm 2 below.

Algorithm 2: Gibbs sampler in the Gamma-Poisson model

Input: Integer matrix $\mathbf{V}$, current values of the parameters $\tilde{\mathbf{W}}$ and $\tilde{\boldsymbol{\alpha}}$

Generate initial state $\mathbf{H}^{(0)}$
for $j = 1, ..., J$ **do**
  # Sample $\mathbf{C}$ given $\mathbf{H}$
  for $f = 1, ..., F$ **do**
    for $n = 1, ..., N$ **do**
      $\underline{\mathbf{c}}_{fn}^{(j)} \sim \text{Mult}\left(v_{fn}, \left[\tilde{\rho}_{f1n}^{(j-1)}, \ \ldots, \tilde{\rho}_{fKn}^{(j-1)}\right]^{\mathrm{T}}\right)$
    end
  end
  # Sample $\mathbf{H}$ given $\mathbf{C}$
  for $k = 1, ..., K$ **do**
    for $n = 1, ..., N$ **do**
      $h_{kn}^{(j)} \sim \text{Gamma}\left(\tilde{\alpha}_k + \sum_f c_{fkn}^{(j)},\ \beta_k + \sum_f \tilde{w}_{fk}\right)$
    end
  end
end

Output: J samples asymptotically drawn from the joint posterior $p(\mathbf{C}, \mathbf{H}|\mathbf{V};\tilde{\boldsymbol{\theta}})$

Note that $\underline{\mathbf{c}}_{fn}^{(j+1)}$ only needs to be sampled when $v_{fn} \neq 0$, since $\underline{\mathbf{c}}_{fn}^{(j+1)} = [0, \ldots, 0]^{\mathrm{T}}$ when $v_{fn} = 0$.

2.5.3 M-step

The M-step consists in minimizing the MC approximation of the different functionals. We first discuss the optimization of $\mathbf{W}$, then the optimization of $\boldsymbol{\alpha}$. Details regarding derivation of the update rules can be found in Appendix 2.D.

Optimizing W. Given a set of J samples $\{\mathbf{C}^{(j)}, \mathbf{H}^{(j)}\}_j$ returned by the Gibbs sampler (after burn-in), minimization of the MC approximation of Q_{CH} in Eq. (2.30) w.r.t. $\mathbf{W}$ yields the closed-form update

$$w_{fk}^{\text{MCEM-CH}} = \frac{\sum_{j,n} c_{fkn}^{(j)}}{\sum_{j,n} h_{kn}^{(j)}}, \tag{2.38}$$

as shown by Dikmen and Févotte (2012). They also showed that the following multiplicative update decreases the MC approximation of Q_{H} in Eq. (2.31) at every iteration

$$w_{fk}^{\text{MCEM-H}} = \tilde{w}_{fk} \frac{\sum_{j,n} h_{kn}^{(j)} v_{fn} [\tilde{\mathbf{W}} \mathbf{H}^{(j)}]_{fn}^{-1}}{\sum_{j,n} h_{kn}^{(j)}}. \tag{2.39}$$

We now derive the novel update for EM-C. The MC approximation of Q_{C} in Eq. (2.32) is given by:

$$\hat{Q}_{\mathrm{C}}(\mathbf{W}) \overset{\text{def}}{=} -\frac{1}{J} \sum_{j=1}^{J} \log p(\mathbf{C}^{(j)}; \mathbf{W}). \tag{2.40}$$

Replacing $p(\mathbf{C}^{(j)}; \mathbf{W})$ by its expression given by Equation (2.13), we obtain:

$$\hat{Q}_{\mathrm{C}}(\mathbf{W}) \overset{\text{c}}{=} \frac{1}{J} \sum_{j,k,n} \left[\alpha_k \log \left(\sum_f w_{fk} + \beta_k \right) + \sum_f c_{fkn}^{(j)} \log \left(\frac{\sum_f w_{fk} + \beta_k}{w_{fk}} \right) \right]. \tag{2.41}$$

The minimization of $\hat{Q}_{\mathrm{C}}$ w.r.t. $\mathbf{W}$ leads to K linear systems of equations that we need to solve for each column $\mathbf{w}_k$:

$$\mathbf{A}_k \mathbf{w}_k = \mathbf{b}_k. \tag{2.42}$$

The matrix $\mathbf{A}_k \in \mathbb{R}^{F \times F}$ is defined by:

$$a_{fg} = \left(JN\alpha_k + \sum_{j,f,n} c_{fkn}^{(j)} \right) \delta_{fg} - \sum_{j,n} c_{fkn}^{(j)}, \tag{2.43}$$

where δ_{fg} is the Kronecker symbol, i.e., $\delta_{fg} = 1$ if and only if $f = g$, and zero otherwise. The vector $\mathbf{b}_k \in \mathbb{R}^{F \times 1}$ is defined by:

$$b_{fk} = \beta_k \sum_{j,n} c_{fkn}^{(j)}. \tag{2.44}$$

The matrix $\mathbf{A}_k$ appears to be the sum of a diagonal matrix with a rank-1 matrix and can be inverted analytically thanks to the Sherman-Morrison formula (Sherman and Morrison, 1950). This results in the closed-form update

$$w_{fk}^{\text{MCEM-C}} = \frac{1}{JN} \frac{\beta_k}{\alpha_k} \sum_{j,n} c_{fkn}^{(j)}. \tag{2.45}$$

Optimizing $\boldsymbol{\alpha}$. Deriving the MC approximations $\hat{Q}_{\mathrm{CH}}$ and $\hat{Q}_{\mathrm{H}}$ w.r.t. $\boldsymbol{\alpha}$ and setting this expression to zero yields the same equation that we need to solve for α_k:

$$NJ \left(\log(\beta_k) - \Psi(\alpha_k) \right) + \sum_{j,n} \log h_{kn}^{(j)} = 0, \tag{2.46}$$

where Ψ denotes the so-called digamma function, defined as the logarithmic derivative of the Gamma function. Such an equation cannot be solved analytically. However, it can easily be tackled numerically thanks to Newton's method, as in Cemgil (2009). This results in the following update

$$\alpha_k = \tilde{\alpha}_k - \frac{NJ\left(\log(\beta_k) - \Psi(\tilde{\alpha}_k)\right) + \sum_{j,n} \log h_{kn}^{(j)}}{-NJ\Psi'(\tilde{\alpha}_k)}. \tag{2.47}$$

As for the optimization of the MC approximation $\hat{Q}_{\mathrm{C}}$, we follow the same reasoning, and obtain the following equation that we need to solve for α_k:

$$\sum_{j,n}\left(\Psi\left(\alpha_k + \sum_f c_{fkn}^{(j)}\right) - \Psi(\alpha_k) + \log\left(\frac{\beta_k}{\sum_f w_{fk} + \beta_k}\right)\right) = 0. \tag{2.48}$$

Once again, we resort to Newton's method (with the updated value of $\mathbf{W}$).

Note that, for all three algorithms, this scheme can yield negative values of α_k. When this occurs, we simply set $\alpha_k = \frac{1}{2}\alpha_k$.

2.6 Experimental work

Though we have discussed a way of optimizing $\boldsymbol{\alpha}$ in the last subsection, all the experimental work presented in this thesis will consider $\boldsymbol{\alpha}$ to be fixed hyperparameters. Python implementations of the three MCEM algorithms are available on GitHub.

2.6.1 Comparison of the algorithms

We begin this section by comparing the three MCEM algorithms proposed for MMLE in the GaP model, using both synthetic toy datasets and real-world data.

2.6.1.1 Experiments with synthetic data

We generate a dataset of $N = 100$ samples according to the GaP model, with the following parameters:

$$\mathbf{W}_1^\star = \begin{bmatrix} 0.638 & 0.075 \\ 0.009 & 0.568 \\ 0.045 & 0.126 \\ 0.308 & 0.231 \end{bmatrix}, \quad \boldsymbol{\alpha}^\star = \boldsymbol{\beta}^\star = \mathbf{1}_K. \tag{2.49}$$

The columns of $\mathbf{W}_1^\star$ have been generated from a Dirichlet distribution (with parameters $\mathbf{1}_F$). The generated dataset (of size 4×100) is denoted by $\mathbf{V}_1$.

We proceed to estimate the dictionary $\mathbf{W}$ using hyperparemeters $K = K^\star + 1 = 3$, $\alpha_k = \beta_k = 1$ with MCEM-C, MCEM-H and MCEM-CH. The algorithms are run for 500

iterations. 300 Gibbs samples are generated at each iteration, with the first 100 samples being discarded for burn-in (this proves to be enough in practice), leading to $J = 200$. The Gibbs sampler at EM iteration $i+1$ is initialized with the last sample obtained at EM iteration i (warm restart). Finally, the algorithms are initialized from the same deterministic starting point given by

$$w_{fk} = \frac{1}{K}\frac{\beta_k}{\alpha_k}\overline{v}_f, \tag{2.50}$$

where $\overline{v}_f$ denotes the mean of the f-th row of $\mathbf{V}$.

Figure 2.3-(a) displays the negative log-likelihood $\mathcal{L}(\mathbf{W})$ w.r.t. CPU time in seconds, exactly computed thanks to the derivations of Section 2.4, and Figure 2.4-(a) displays the norm of the three columns of the iterates, also w.r.t. CPU time in seconds. The three algorithms have almost identical computation times, since most of the computational burden resides in the Gibbs sampling procedure that is common to the three algorithms. Moreover, the three algorithms converge to the same point, with MCEM-C converging marginally faster than the other two in this case.

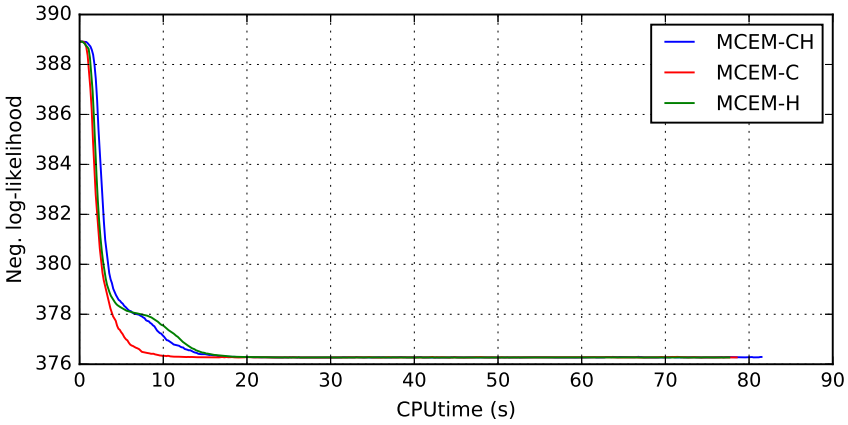
We then proceed to generate a second dataset $\mathbf{V}_2$ according to the GaP model, with now $\mathbf{W}_2^\star = 100 \times \mathbf{W}_1^\star$. Over-dispersion ratios, defined as the ratio between the variance and the mean, as well as the proportion of zeros, are given in the first two rows of Table 2.1. As we can see, the values of $\mathbf{V}_2$ are way more dispersed than those of $\mathbf{V}_1$. Moreover, almost all values of $\mathbf{V}_2$ are non-zero, unlike the values of $\mathbf{V}_1$, which are roughly 65% zero.

We apply the exact same experimental protocol to $\mathbf{V}_2$ as we did for $\mathbf{V}_1$, except that the algorithms are now run for a larger number of 1000 iterations. The large values of $\mathbf{V}_2$ have a huge impact on $\text{card}(\mathcal{C})$, and as such it is now impossible to compute the likelihood in reasonable time. The norms of the columns of the iterates are displayed on Figure 2.5-(a). As we can see, MCEM-C clearly outperforms the other two algorithms in this scenario. This behavior has been consistently found when estimating dictionaries from datasets with sufficiently large values.

In order to check whether this phenomenon is imputed to the scale of the dictionary to be estimated, or to the conditioning of the data matrix itself, we repeat the previously described experimental protocol on $\mathbf{V}_1$ and $\mathbf{V}_2$, except for the value of β_k which is changed to alter the scale of the learned dictionaries. In particular, for $\mathbf{V}_1$ we now set $\beta_k = 100$ (so that the values of the learned dictionary will be 100 times larger), and for $\mathbf{V}_2$ we set $\beta_k = 0.01$ (so that the values of the learned dictionary will be 100 times smaller).

The norms of the columns of the iterates are displayed on Figure 2.4-(b) for $\mathbf{V}_1$ (and Figure 2.3-(b) displays the associated negative log-likelihood $\mathcal{L}(\mathbf{W})$), and on Figure 2.5-(b) for $\mathbf{V}_2$, for the three algorithms. As can be observed, changing the value of β_k does not impact at all the behavior of the algorithms. We thus conclude that the drastic difference in convergence observed on the dataset $\mathbf{V}_2$ has to do with the conditioning of the data matrix. We further investigate this phenomenon on a real dataset in the next sub-subsection.

(a) Dataset $\mathbf{V}_1$ – $\beta = 1$



(b) Dataset $\mathbf{V}_1$ – $\beta = 100$

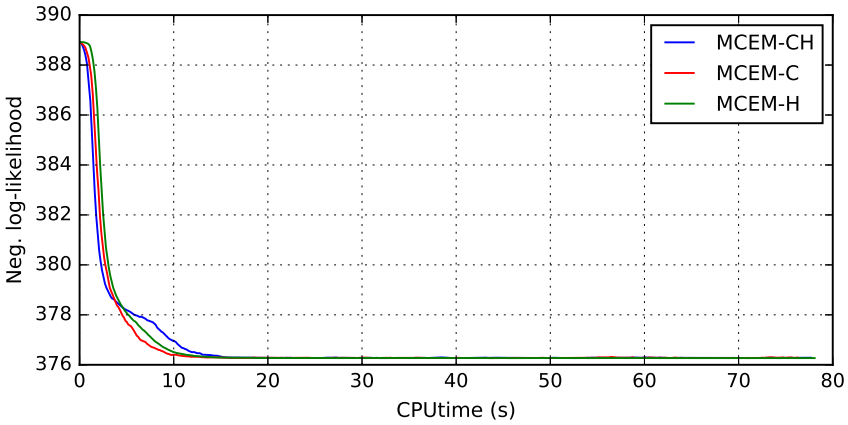


Figure 2.3: $\mathcal{L}(\mathbf{W})$ w.r.t. CPU time in seconds for the three MCEM algorithms on toy dataset $\mathbf{V}_1$. From top to bottom:(a) $\beta = 1$; (b) $\beta = 100$.

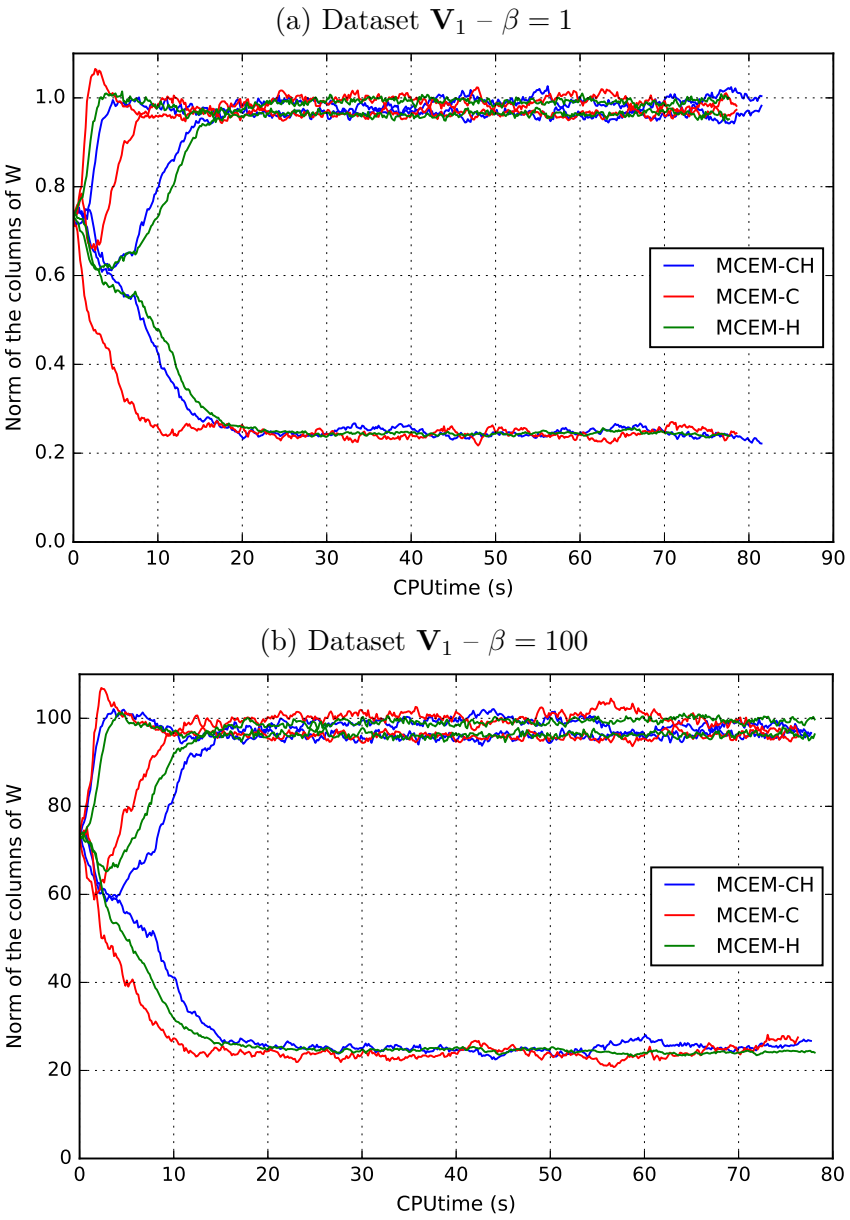
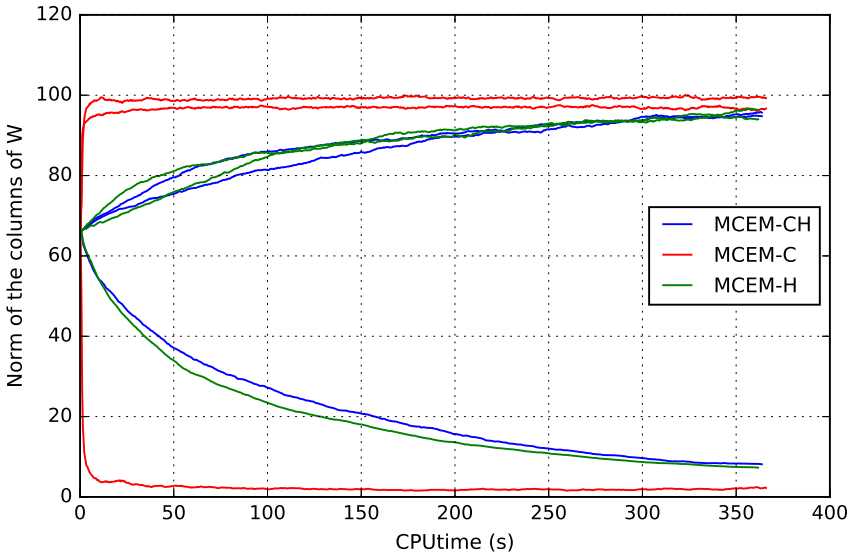


Figure 2.4: Evolution of the norm of each of the $K = 3$ columns $\mathbf{w}_k$ of the dictionary w.r.t. CPU time in seconds for the three MCEM algorithms on toy dataset $\mathbf{V}_1$. From top to bottom:(a) $\beta = 1$; (b) $\beta = 100$.

(a) Dataset $\mathbf{V}_2$ – $\beta = 1$



(b) Dataset $\mathbf{V}_2$ – $\beta = 0.01$

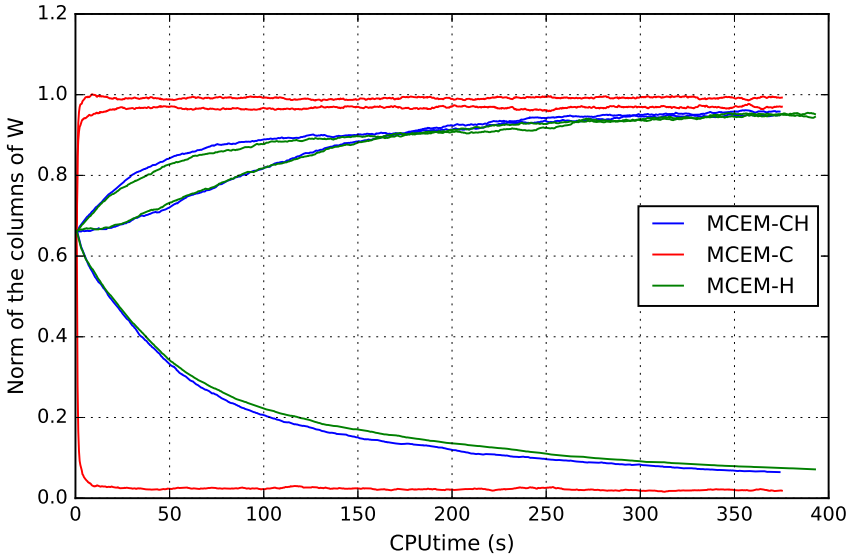


Figure 2.5: Evolution of the norm of each of the $K = 3$ columns $\mathbf{w}_k$ of the dictionary w.r.t. CPU time in seconds for the three MCEM algorithms on toy dataset $\mathbf{V}_2$. From top to bottom:(a) $\beta = 1$; (b) $\beta = 0.01$.

Dataset	F	N	Mean	Var	Ratio	% of zeros
$\mathbf{V}_1$	4	100	0.55	0.99	1.80	65.3 %
$\mathbf{V}_2$	4	100	49.54	3017.01	60.90	0.5 %
`Taste Profile`	1509	805	0.15	2.00	13.00	94.7 %

Table 2.1: Characteristics of the three datasets considered for the experimetanl comparison of the three EM algorithms. $\mathbf{V}_1$ and $\mathbf{V}_2$ are synthetic datasets, whereas the `Taste Profile` is a real dataset. The over-dispersion ratio corresponds to the variance to mean ratio.

2.6.1.2 Experiments with real data

Finally, we consider the `Taste Profile` dataset (Bertin-Mahieux et al., 2011). This dataset contains the listening history of users in the form of song play counts. We use a subset of the original dataset, as in Gouvert et al. (2018), leading to a dataset of $F = 1509$ users and $N = 805$ songs. The matrix $\mathbf{V}$ is quite sparse as 94.7% of its coefficients are zeros. This saves a large amount of computational effort, because we only need to sample $\underline{\mathbf{c}}_{fn}$ for pairs (f, n) such that v_{fn} is non-zero. Moreover, the count values range from 0 to 421, with an over-dispersion ratio of roughly 13 (see the last row of Table 2.1).

We apply the three algorithms with $K = 10$ and $\alpha_k = \beta_k = 1$. The algorithms are run for 1000 iterations. 150 Gibbs samples are generated in each iteration with the first 50 being discarded for burn-in (i.e., $J = 100$). The Gibbs sampler at iteration $i+1$ is again initialized with warm restart. The algorithms are initialized in the same fashion as before.

Figure 2.6 shows the column norms of the iterates w.r.t. CPU time in minutes for the first 800 iterations. The difference in convergence speed between MCEM-C and the other two algorithms is again striking. MCEM-C, the algorithm proposed in this chapter, efficiently explores the parameter space in the first iterations and converges dramatically faster than MCEM-CH or MCEM-H. The algorithms here converge to different solutions, which confirms the non-convexity of $\mathcal{L}(\mathbf{W})$. Other runs confirmed that MCEM-C is consistently faster.

As such, we conjecture that the favorable properties of MCEM-C are linked to the over-dispersion of the data, but not to the sparsity (as it has been observed on both **V2** and the `Taste Profile` dataset). We are unable to provide a more detailed explanation at this stage.

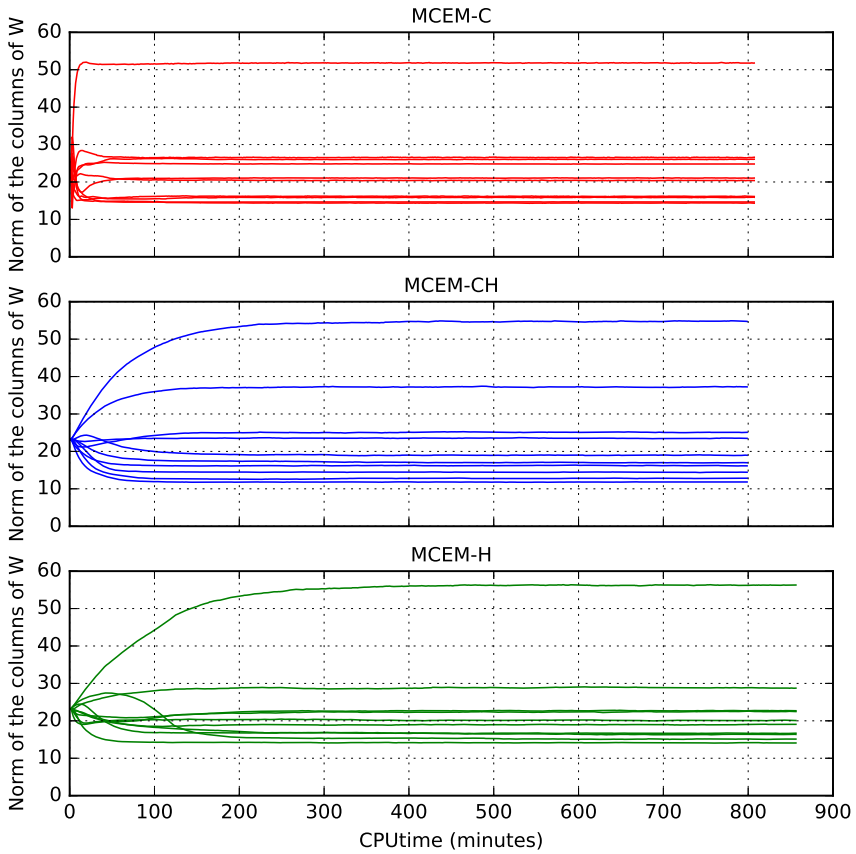


Figure 2.6: Evolution of the norm of each of the $K = 10$ columns of the dictionaries w.r.t. CPU time in minutes for the three MCEM algorithms on the `Taste Profile` dataset.

2.6.2 Examples of the self-regularization phenomenon

We conclude this section by showing examples of the self-regularization phenomenon on both a synthetic dataset and a real dataset.

2.6.2.1 On synthetic data

We generate a toy dataset of size 8×100 according to the GaP model with $\alpha_k^\star = 0.05$, $\beta_k^\star = 1$, and a dictionary $\mathbf{W}^\star$ of size 8×3 graphically displayed on Figure 2.8-(a). Each column of $\mathbf{W}^\star$ has been generated from a Dirichlet distribution, rescaled to sum up to 8.

We then proceed to estimate a dictionary $\hat{\mathbf{W}}$ of size $8 \times K$, for all values of K between 1 and F, with the algorithm MCEM-C. In each of these experiments, the hyperparameters are set to $\boldsymbol{\alpha} = \mathbf{1}_K$ and $\boldsymbol{\beta} = \mathbf{1}_K$, and the algorithm is run for 500 iterations. 300 Gibbs samples are generated at each iteration with a burn-in of the first 100 samples, leading to $J = 200$. We then retrieve the iterate with the smallest negative log-likelihood $\mathcal{L}(\mathbf{W})$, which we are able to compute exactly in this small-dimensioned problem.

All the subplots (b) to (i) of Figure 2.8 display the estimated $\hat{\mathbf{W}}$ for all values of K between 1 and 8. Note that the color bar of Figure 2.8-(a) is shared by all these subplots, so that direct visual comparison may be carried out. As we can see, when $K > 4$, the additional columns have very small norm, or are full of zeros, hence illustrating the behavior first observed in Dikmen and Févotte (2012) and explained in this chapter. This is further confirmed on Figure 2.7, which displays the associated negative log-likelihoods $\mathcal{L}(\hat{\mathbf{W}})$ for all K. The likelihood reaches a plateau for $K \geq 4$.

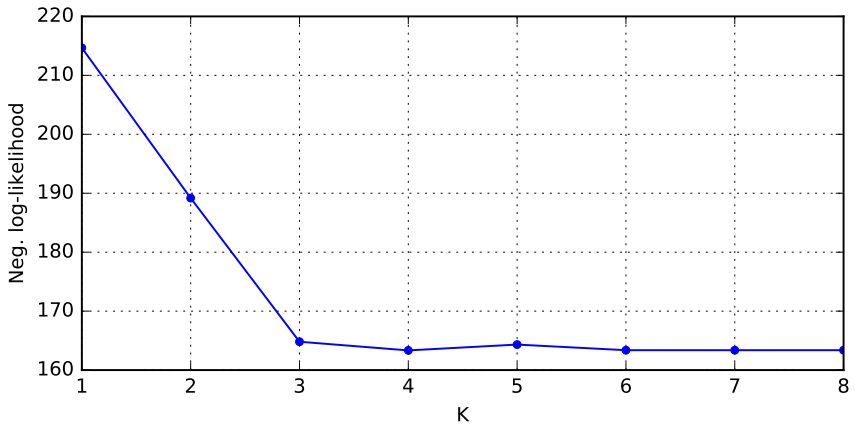


Figure 2.7: $\mathcal{L}(\hat{\mathbf{W}})$ w.r.t. K.

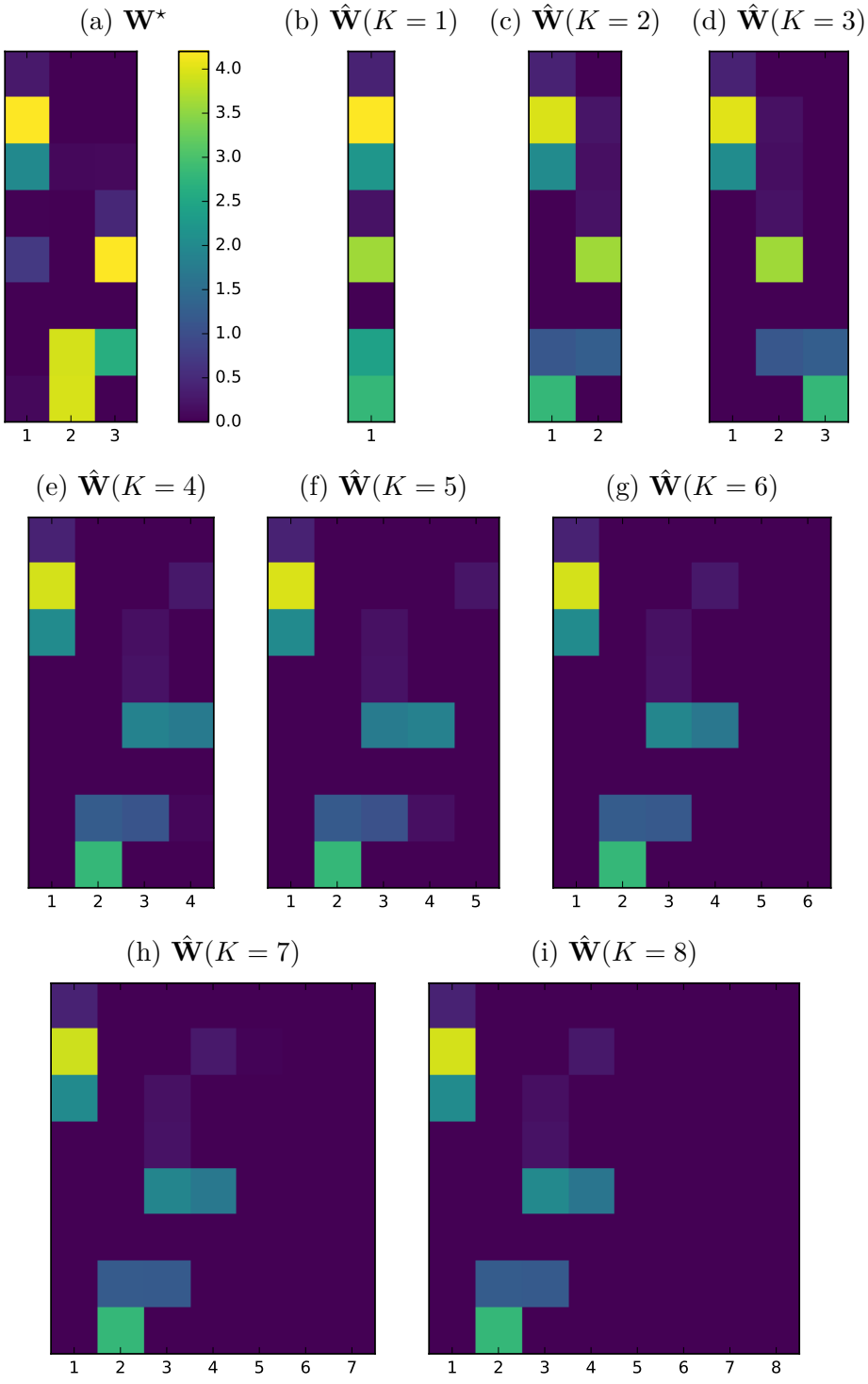


Figure 2.8: (a) Ground truth dictionary $\mathbf{W}^\star$; (b)-(g) Estimated dictionaries $\hat{\mathbf{W}}$ by MMLE for K from 1 (b) to 8 (i). The color bar displayed in (a) is common to all subplots.

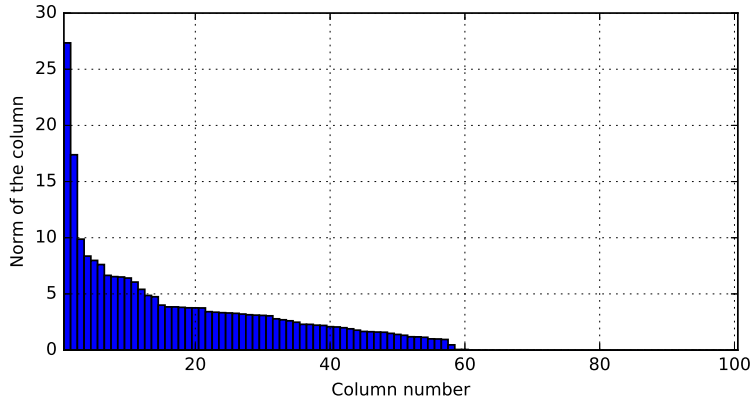


Figure 2.9: Norm of the $K = 100$ columns of the dictionary learned on the `Taste Profile` dataset.

2.6.2.2 On a real dataset

We run MCEM-C on the `Taste Profile` dataset, setting $K = 100$, and the hyperparameters to $\boldsymbol{\alpha} = \mathbf{1}_K$ and $\boldsymbol{\beta} = \mathbf{1}_K$. The algorithm is run for 1000 iterations. 150 Gibbs samples are generated at each iteration with a burn-in of the first 50, leading to $J = 100$. On Figure 2.9 is displayed the norm of each of the $K = 100$ columns of the estimated $\hat{\mathbf{W}}$. Exactly 40 columns are comprised only of zeros.

2.7 Discussion

We conclude this chapter by recalling the obtained results, and by discussing some limitations and open questions.

One of the main results showcased in this chapter is the semi-explicit closed-form expression of the marginal likelihood in the Gamma-Poisson matrix factorization model. As a matter of fact, it was the conjugacy between the Gamma distribution and the Poisson distribution which enabled us to derive such an expression (giving rise here to the negative multinomial distribution). If another prior distribution had been assumed on $\mathbf{H}$ (i.e., not Gamma), we would not have been able to derive a similar analysis, since the marginalization of $\mathbf{H}$ would be intractable. The extension of our analysis to more general models with any kind of prior on $\mathbf{H}$ is not straightforward, and the occurrence of analogous phenomena in different settings remains an open question at this point.

The analytical expression of the marginal likelihood led in turn to a formal explanation of the "self-regularization" phenomenon described in Dikmen and Févotte (2012). The key argument was the rewriting of the marginal likelihood as a regularization term, known to be sparsity-inducing, and a data-fitting term. We would however like to point out two

elements that were not taken into account in this analysis. The first one is the nature of the data-fitting term. As it turns out, we were unable to give a meaningful description of the interaction between this term and the regularization term. The second one is the influence of the hyperparameters, namely $\boldsymbol{\alpha}$ (since $\boldsymbol{\beta}$ is merely a scale parameter), which could not be quantified either.

As for the optimization of the marginal likelihood, we proposed a novel EM algorithm (EM-C), and compared the three variants of the EM algorithm on both synthetic and real datasets. Our experimental work demonstrated the favorable properties of EM-C. EM-based algorithms, by nature, assume the marginal likelihood to be intractable. An exciting perspective would therefore be to break out of EM-based schemes; in particular to design algorithms taking advantage of the expression of the marginal likelihood for direct optimization.

As such, we still have no alternative to the MCEM algorithms presented in Section 2.5 at this point. MCEM algorithms are extremely computationally intensive, since they require sampling from a target distribution which changes at each EM iteration, and these samples cannot be recycled[3]. One might consider resorting to variational inference, but fundamentally provides an approximate solution to the problem. Variants of the EM algorithm, such as SAEM (Delyon et al., 1999; Kuhn and Lavielle, 2004) or on-line EM (Cappé and Moulines, 2009) have been considered, but did not lead to any major improvement in our case.

We conclude this discussion by mentioning a very recent work, which leveraged on our closed-form expression of the marginal likelihood for the numerical evaluation of $\mathcal{L}(\mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ (Capdevila et al., 2018). This is a task of interest in text information retrieval, to assess the likelihood a previously unseen document for instance. More precisely, the authors developed a so-called left-to-right algorithm (Wallach et al., 2009b) for this task.

[3]Note that these EM algorithms exploit the Poisson-Gamma conjugacy, since it provides a Gibbs sampler for the posterior of the latent variables where all the conditionals are known. In a non-conjugate model, we would have to resort to additional Metropolis-Hastings steps, for example.

Appendices to Chapter 2

Contents

2.A Probability distributions **78**
2.A.1 Negative binomial distribution 78
2.A.2 Negative multinomial distribution 79
2.B Stars and bars theorem **80**
2.C Gibbs sampling of the posterior distribution **80**
2.C.1 First conditional 80
2.C.2 Second conditional 81
2.D EM algorithms **81**
2.D.1 MCEM-CH 81
2.D.2 MCEM-H 82
2.D.3 MCEM-C 83

2.A Probability distributions

2.A.1 Negative binomial distribution

A discrete random variable X is said to have a negative binomial (NB) distribution with parameters $\alpha > 0$ (the dispersion or shape parameter) and $p \in [0, 1]$ if, for all $c \in \mathbb{N}$, the p.m.f. of X is given by:

$$\mathbb{P}(X = c; \alpha, p) = \frac{\Gamma(\alpha + c)}{\Gamma(\alpha)\, c!}(1 - p)^{\alpha} p^{c}. \tag{2.51}$$

We have

$$\mathbb{E}(X) = \frac{\alpha p}{1 - p}, \qquad \mathrm{var}(X) = \frac{\alpha p}{(1 - p)^2}. \tag{2.52}$$

The variance of X is larger than its mean. The NB distribution is therefore suitable to model over-dispersed count data. Indeed, it offers more flexibility than the Poisson

distribution where the variance and the mean are equal. The NB distribution may be obtained via a Gamma-Poisson mixture. Consider the following hierarchical model:

$$\lambda \sim \text{Gamma}(\alpha, \beta), \tag{2.53}$$
$$X|\lambda \sim \text{Poisson}(\lambda). \tag{2.54}$$

Then, for all $c \in \mathbb{N}$:

$$\mathbb{P}(X = c\,;\alpha, p) = \int_{\mathbb{R}_+} \mathbb{P}(X = c|\lambda)p(\lambda)\mathrm{d}\lambda \tag{2.55}$$
$$= \int_{\mathbb{R}_+} \frac{\lambda^c}{c!}\exp(-\lambda)\frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}\exp(-\beta\lambda) \tag{2.56}$$
$$= \frac{\Gamma(\alpha+c)}{\Gamma(\alpha)c!}\frac{\beta^\alpha}{(\beta+1)^{c+\alpha}} = \text{NB}\left(\alpha, \frac{1}{\beta+1}\right). \tag{2.57}$$

2.A.2 Negative multinomial distribution

The negative multinomial (NM) distribution (Sibuya et al., 1964) is the multivariate generalization of the NB distribution. It is parametrized by a dispersion parameter $\alpha > 0$ and a vector of event probabilities $\mathbf{p} = [p_1, \ldots, p_F]^{\mathrm{T}}$, where $0 \leq p_f \leq 1$ and $\sum_f p_f \leq 1$. Denoting $p_0 = 1 - \sum_f p_f$, and for all $(c_1, \ldots, c_F) \in \mathbb{N}^F$, the p.m.f. of the NM distribution is given by:

$$\mathbb{P}(X_1 = c_1, \ldots, X_F = c_F) = \frac{\Gamma(\alpha + \sum_f c_f)}{\Gamma(\alpha)\prod_f c_f!}p_0^\alpha \prod_f p_f^{c_f}. \tag{2.58}$$

We have

$$\mathbb{E}(X_f) = \alpha\frac{p_f}{p_0}, \quad \text{var}(X_f) = \alpha\frac{p_f(p_f + p_0)}{p_0^2}. \tag{2.59}$$

Proposition 2.1. An equivalent characterization of the NM distribution is given by the marginal distribution of X in the following generative process.

$$L \sim \text{NB}\left(\alpha, \sum_f p_f\right), \tag{2.60}$$
$$X|L \sim \text{Mult}\left(L, \left[\frac{p_1}{\sum_f p_f}, \ldots, \frac{p_F}{\sum_f p_f}\right]^{\mathrm{T}}\right). \tag{2.61}$$

Proof.

$$\mathbb{P}(X = [c_1, \ldots, c_F]^{\mathrm{T}}) = \mathbb{P}(X = [c_1, \ldots, c_F]^{\mathrm{T}}|L) \times \mathbb{P}(L) \tag{2.62}$$
$$= \frac{L!}{\prod_f c_f!}\prod_f \left(\frac{p_f}{\sum_f p_f}\right)^{c_f} \frac{\Gamma(\alpha + L)}{\Gamma(\alpha)L!}\left(1 - \sum_f p_f\right)^\alpha \left(\sum_f p_f\right)^L. \tag{2.63}$$

Noting that $L = \sum_f c_f$ completes the proof. □

2.B Stars and bars theorem

The so-called "stars and bars" theorems refer to two elementary results in combinatorics. They take their name from a visual representation with stars and bars, which helps proving the theorems. We only state here the theorem of interest.

Proposition 2.2. For any pair of positive integers n and k, the number of k-tuples of non-negative integers whose sum is n is given by

$$\binom{n+k-1}{k-1} = \binom{n+k-1}{n}. \tag{2.64}$$

2.C Gibbs sampling of the posterior distribution

Given the current values of the parameters, namely $\tilde{\mathbf{W}}$ and $\tilde{\boldsymbol{\alpha}}$, we would like to be able to sample from the posterior of the latent variables $p(\mathbf{C}, \mathbf{H}|\mathbf{V}; \tilde{\mathbf{W}}, \tilde{\boldsymbol{\alpha}})$. Let us denote $\tilde{\boldsymbol{\theta}} = \{\tilde{\mathbf{W}}, \tilde{\boldsymbol{\alpha}}\}$. In a Gibbs sampler, we are interested in the conditionals $p(\mathbf{C}|\mathbf{H}, \mathbf{V}; \tilde{\boldsymbol{\theta}})$ and $p(\mathbf{H}|\mathbf{C}, \mathbf{V}; \tilde{\boldsymbol{\theta}})$.

2.C.1 First conditional

We recall the following result.

Proposition 2.3. Let $X_1, ...X_K$ be independent Poisson random variables with rates λ_k. Let $V = \sum_k X_k$. Then the conditional distribution $\mathbb{P}(X_1, \dots, X_K|V)$ is multinomial with number of trials V and event probabilities

$$\left[\frac{\lambda_1}{\sum_k \lambda_k}, \dots, \frac{\lambda_K}{\sum_k \lambda_k}\right]^{\mathrm{T}}. \tag{2.65}$$

Proof. Consider $(X_1, \dots, X_K)$ such that $\sum_k X_k = V$. Then we have

$$\mathbb{P}(X_1, \dots, X_K|V) = \frac{\mathbb{P}(V|X_1, ..., X_K)\mathbb{P}(X_1, ..., X_K)}{\mathbb{P}(V)} \tag{2.66}$$

$$= \frac{\prod_k \frac{\lambda_k^{x_k}}{x_k!} \exp(-\lambda_k)}{\frac{(\sum_k \lambda_k)^v}{v!} \exp(-\sum_k \lambda_k)} \tag{2.67}$$

$$= \frac{v!}{\prod_k x_k!} \prod_k \left(\frac{\lambda_k}{\sum_k \lambda_k}\right)^{x_k}. \tag{2.68}$$

□

Therefore in our case, we can easily derive that

$$p(\mathbf{C}|\mathbf{H},\mathbf{V};\tilde{\boldsymbol{\theta}}) = \prod_{f,n} p(\underline{\mathbf{c}}_{fn}|\mathbf{h}_n, v_{fn}; \tilde{\mathbf{w}}_f), \tag{2.69}$$

with

$$p(\underline{\mathbf{c}}_{fn}|\mathbf{h}_n, v_{fn}; \tilde{\mathbf{w}}_f) = \text{Mult}\left(v_{fn}, \left[\frac{\tilde{w}_{f1}h_{1n}}{\sum_k \tilde{w}_{fk}h_{kn}}, \ldots, \frac{\tilde{w}_{fK}h_{Kn}}{\sum_k \tilde{w}_{fk}h_{kn}}\right]^{\mathrm{T}}\right). \tag{2.70}$$

2.C.2 Second conditional

We have

$$\begin{aligned}
p(\mathbf{H}|\mathbf{C},\mathbf{V};\tilde{\boldsymbol{\theta}}) &= p(\mathbf{H}|\mathbf{C};\tilde{\boldsymbol{\theta}}) && (2.71)\\
&\propto p(\mathbf{C}|\mathbf{H};\tilde{\mathbf{W}})p(\mathbf{H};\tilde{\boldsymbol{\alpha}}) && (2.72)\\
&\propto \prod_{f,k,n} p(c_{fkn}|h_{kn};\tilde{w}_{fk}) \prod_{k,n} p(h_{kn};\tilde{\alpha}_k) && (2.73)\\
&\propto \prod_{f,k,n} h_{kn}^{c_{fkn}} \exp(-w_{fk}h_{kn}) \prod_{k,n} h_{kn}^{\alpha_k-1} \exp(-\beta_k h_{kn}) && (2.74)\\
&\propto \prod_{k,n} h_{kn}^{\sum_f c_{fkn}+\alpha_k-1} \exp\left(-\left(\sum_f w_{fk}+\beta_k\right)h_{kn}\right) && (2.75)\\
&= \text{Gamma}\left(\alpha_k + \sum_f c_{fkn},\ \beta_k + \sum_f w_{fk}\right). && (2.76)
\end{aligned}$$

2.D EM algorithms

2.D.1 MCEM-CH

The MC approximation of Q_{CH} writes

$$\begin{aligned}
\hat{Q}_{\text{CH}}(\boldsymbol{\theta}) &= \frac{1}{J}\sum_j \log p(\mathbf{C}^{(j)}, \mathbf{H}^{(j)}; \boldsymbol{\theta}) && (2.77)\\
&= \frac{1}{J}\sum_j \left(\log p(\mathbf{C}^{(j)}|\mathbf{H}^{(j)};\mathbf{W}) + \log p(\mathbf{H}^{(j)};\boldsymbol{\alpha})\right) && (2.78)\\
&= \frac{1}{J}\sum_j \left(\sum_{f,k,n} \log p(c_{fkn}^{(j)}|h_{kn}^{(j)}; w_{fk}) + \sum_{k,n} \log p(h_{kn}^{(j)};\alpha_k)\right) && (2.79)\\
&= \frac{1}{J}\left(\sum_{j,f,k,n} \log\left(\frac{(w_{fk}h_{kn}^{(j)})^{c_{fkn}^{(j)}}}{c_{fkn}^{(j)}!}\exp(-w_{fk}h_{kn})\right) + \sum_{j,k,n} \log\left(\frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)}(h_{kn}^{(j)})^{\alpha_k-1}\exp(-\beta_k h_{kn})\right)\right). && (2.80)
\end{aligned}$$

Deriving w.r.t. a specific w_{fk} yields

$$\frac{1}{J}\sum_{j,n}\left(\frac{c_{fkn}^{(j)}}{w_{fk}} - h_{kn}^{(j)}\right) = 0, \tag{2.81}$$

which can be easily solved as

$$w_{fk} = \frac{\sum_{j,n} c_{fkn}^{(j)}}{\sum_{j,n} h_{kn}^{(j)}}. \tag{2.82}$$

Deriving w.r.t. a specific α_k yields

$$\frac{1}{J}\sum_{j,n}\Big(\log(\beta_k) - \Psi(\alpha_k) + \log(h_{kn}^{(j)})\Big) = 0, \tag{2.83}$$

which we numerically solve with Newton's method.

2.D.2 MCEM-H

The MC approximation of Q_{H} writes

$$\hat{Q}_{\mathrm{H}}(\boldsymbol{\theta}) = \frac{1}{J}\sum_{j}\log p(\mathbf{V}, \mathbf{H}^{(j)}; \boldsymbol{\theta}) \tag{2.84}$$

$$= \frac{1}{J}\sum_{j}\Big(\log p(\mathbf{V}|\mathbf{H}^{(j)}; \mathbf{W}) + \log p(\mathbf{H}^{(j)}; \boldsymbol{\alpha})\Big) \tag{2.85}$$

Regarding $\mathbf{W}$, we can write

$$\hat{Q}_{\mathrm{H}}(\mathbf{W}) = \frac{1}{J}\sum_{j} D_{\mathrm{KL}}(\mathbf{V}|\mathbf{W}\mathbf{H}^{(j)}) + \mathrm{cst}. \tag{2.86}$$

Noting the parallel with the standard KL-NMF problem, this can be tackled with an MM algorithm, see Dikmen and Févotte (2012). Regarding $\boldsymbol{\alpha}$, we note that $\hat{Q}_{\mathrm{H}}(\boldsymbol{\alpha}) = \hat{Q}_{\mathrm{CH}}(\boldsymbol{\alpha})$, therefore yielding Eq. (2.83) again.

2.D.3 MCEM-C

Optimizing W. The MC approximation of Q_{C} writes

$$\hat{Q}_{\mathrm{C}}(\boldsymbol{\theta}) = \frac{1}{J}\sum_{j} \log p(\mathbf{C}^{(j)};\boldsymbol{\theta}) \tag{2.87}$$

$$= \frac{1}{J}\sum_{j,k,n} \log p(\mathbf{c}_{kn}^{(j)};\boldsymbol{\theta}) \tag{2.88}$$

$$= \frac{1}{J}\sum_{j,k,n} \left(\log\left(\frac{\Gamma(\alpha_k + \sum_f c_{fkn}^{(j)})}{\Gamma(\alpha_k)}\right) + \alpha_k \log\left(\frac{\beta_k}{\sum_f w_{fk} + \beta_k}\right) + \sum_f c_{fkn}^{(j)} \log\left(\frac{w_{fk}}{\sum_f w_{fk} + \beta_k}\right)\right). \tag{2.89}$$

Deriving Eq. (2.89) w.r.t. to a specific w_{fk}, and setting it to zero, we have:

$$\frac{1}{J}\sum_{j,n}\left(\frac{c_{fkn}^{(j)}}{w_{fk}} - \frac{\alpha_k + \sum_f c_{fkn}^{(j)}}{\sum_{f'} w_{f'k} + \beta_k}\right) = 0 \tag{2.90}$$

$$\left(\frac{\sum_{j,n} c_{fkn}^{(j)}}{w_{fk}} - \frac{NJ\alpha_k + \sum_{j,n,f} c_{fkn}^{(j)}}{\sum_{f'} w_{f'k} + \beta_k}\right) = 0, \tag{2.91}$$

that is

$$\left(NJ\alpha_k + \sum_{j,n,f} c_{fkn}^{(j)}\right) w_{fk} = \left(\sum_{j,n} c_{fkn}^{(j)}\right)\left(\sum_f w_{fk} + \beta_k\right) \tag{2.92}$$

$$\left(NJ\alpha_k + \sum_{j,n,f} c_{fkn}^{(j)}\right) w_{fk} - \left(\sum_{j,n} c_{fkn}^{(j)}\right)\sum_f w_{fk} = \beta_k \sum_{j,n} c_{fkn}^{(j)}. \tag{2.93}$$

We see that the variables at hand are $w_{1k}, ..., w_{Fk}$, which is the k-th column of $\mathbf{W}$. We therefore obtain a linear system of F equations with F variables, which we rewrite in a matricial form as:

$$\mathbf{A}_k \mathbf{w}_k = \mathbf{b}_k. \tag{2.94}$$

Introducing the notations

$$\lambda_{fk} = \sum_{j,n} c_{fkn}^{(j)} \tag{2.95}$$

$$\eta_k = NJ\alpha_k, \tag{2.96}$$

$\mathbf{A}_k \in \mathbb{R}^{F\times F}$ and $\mathbf{b}_k \in \mathbb{R}^{F\times 1}$ are defined as:

$$\mathbf{A}_k = \left(\eta_k + \sum_f \lambda_{fk}\right)\mathbf{I}_F - \boldsymbol{\lambda}_k \mathbf{1}_F^{\mathrm{T}} \tag{2.97}$$

$$\mathbf{b}_k = \beta_k \boldsymbol{\lambda}_k, \tag{2.98}$$

where $\mathbf{I}_F$ denotes the identity matrix of size F, $\mathbf{1}_F$ denotes the vector of ones of size F, and $\boldsymbol{\lambda}_k = [\lambda_{1k}, \ldots, \lambda_{Fk}]^{\mathrm{T}}$. Therefore, to update $\mathbf{W}$, we need to solve K independent linear systems, one for each column of $\mathbf{W}$:

$$\mathbf{w}_k = \mathbf{A}_k^{-1}\mathbf{b}_k. \tag{2.99}$$

Let us consider Eq. (2.99) for a certain k. The index k will be dropped from now on for enhanced readability. We want to check if $\mathbf{w}$ as the solution of Eq. (2.99) is non-negative. Since $\mathbf{b}$ is non-negative, a sufficient condition is to prove that $\mathbf{A}^{-1}$ is non-negative. The matrix $\mathbf{A}$ has a particular structure. Indeed, it is the sum of a diagonal matrix and a rank-one matrix. As such, its inverse can be described thanks to the Sherman-Morrison formula (Sherman and Morrison, 1950).

Theorem 2.3. Suppose $\mathbf{X} \in \mathbb{R}^{n\times n}$ is an invertible square matrix. Then, for all $\mathbf{u}$ and $\mathbf{v} \in \mathbb{R}^{n\times 1}$, $\mathbf{X} + \mathbf{u}\mathbf{v}^{\mathrm{T}}$ is invertible if and only if $1 + \mathbf{v}^{\mathrm{T}}\mathbf{X}^{-1}\mathbf{u} \neq 0$. If $\mathbf{X} + \mathbf{u}\mathbf{v}^{\mathrm{T}}$ is invertible, its inverse is given by :

$$(\mathbf{X} + \mathbf{u}\mathbf{v}^{\mathrm{T}})^{-1} = \mathbf{X}^{-1} - \frac{\mathbf{X}^{-1}\mathbf{u}\mathbf{v}^{\mathrm{T}}\mathbf{X}^{-1}}{1 + \mathbf{v}^{\mathrm{T}}\mathbf{X}^{-1}\mathbf{u}}. \tag{2.100}$$

In our problem, we identify

$$\mathbf{X} = \left(\eta + \sum_f \lambda_f\right)\mathbf{I}_F, \quad \mathbf{u} = \boldsymbol{\lambda}_k, \quad \mathbf{v} = -\mathbf{1}_F. \tag{2.101}$$

$\mathbf{X}$ is a diagonal matrix and therefore invertible. We have

$$1 + \mathbf{v}^{\mathrm{T}}\mathbf{X}^{-1}\mathbf{u} = 1 - \frac{\sum_f \lambda_f}{\eta + \sum_f \lambda_f} = \frac{\eta}{\eta + \sum_f \lambda_f} > 0, \tag{2.102}$$

because $\eta > 0$. Therefore $\mathbf{A}$ is always invertible, and its inverse is given by

$$\mathbf{A}^{-1} = \frac{1}{\left(\eta + \sum_f \lambda_f\right)}\mathbf{I}_F - \frac{\eta + \sum_f \lambda_f}{\eta}\left(-\frac{1}{(\eta + \sum_f \lambda_f)^2}\right)\boldsymbol{\lambda}_k \mathbf{1}_F^{\mathrm{T}} \tag{2.103}$$

$$= \frac{1}{\left(\eta + \sum_f \lambda_f\right)}\mathbf{I}_F + \frac{1}{\eta(\eta + \sum_f \lambda_f)}\boldsymbol{\lambda}_k \mathbf{1}_F^{\mathrm{T}}. \tag{2.104}$$

Therefore $\mathbf{A}^{-1}$ is always non-negative, hence the non-negativity of $\mathbf{w}$. Moreover a closed-form expression of w is available:

$$w_f = \beta \left(\frac{\lambda_f}{\eta + \sum_{f'} \lambda_{f'}} + \frac{\lambda_f \sum_{f'} \lambda_{f'}}{\eta(\eta + \sum_{f'} \lambda_{f'})} \right) \tag{2.105}$$

$$= \beta \frac{\lambda_f(\eta + \sum_{f'} \lambda_{f'})}{\eta(\eta + \sum_{f'} \lambda_{f'})} = \beta \frac{\lambda_f}{\eta}. \tag{2.106}$$

Back to the initial problem, we obtain the following update rule

$$w_{fk} = \frac{\beta_k}{NJ\alpha_k} \sum_{j,n} c_{fkn}^{(j)}. \tag{2.107}$$

Optimizing α. Deriving Eq. (2.89) w.r.t. to a specific α_k, and setting it to zero yields

$$\frac{1}{J} \sum_{j,n} \left(\Psi \left(\alpha_k + \sum_f c_{fkn}^{(j)} \right) - \Psi(\alpha_k) + \log \left(\frac{\beta_k}{\sum_f w_{fk} + \beta_k} \right) \right) = 0, \tag{2.108}$$

which, once again, cannot be solved analytically in closed form. We resort to the Newton's method, as previously described, yielding the update

$$\alpha_k = \alpha_k - \frac{\sum_{j,n} \left(\Psi \left(\alpha_k + \sum_f c_{fkn}^{(j)} \right) - \Psi(\alpha_k) + \log \left(\frac{\beta_k}{\sum_f w_{fk} + \beta_k} \right) \right)}{NJ \left(\Psi' \left(\alpha_k + \sum_f c_{fkn}^{(j)} \right) - \Psi'(\alpha_k) \right)}. \tag{2.109}$$

Chapter 3

Maximum Marginal Likelihood Estimation in the Multiplicative Exponential Model

The contents of this chapter have not been submitted for publication. This is joint work with Cédric Févotte.

Contents

3.1 Introduction **88**
3.2 Model **89**
3.2.1 Observation model 89
3.2.2 Working with complex data 90
3.2.3 Prior distribution 90
3.2.4 Objective function 91
3.3 Marginalization of H **91**
3.4 Marginal likelihood **92**
3.4.1 Analytical expression 92
3.4.2 Self-regularization 93
3.5 Optimization algorithms **93**
3.5.1 Expectation-Minimization 93
3.5.2 Monte Carlo E-step 94
3.5.3 Monte Carlo M-step 95
3.6 Estimating audio sources **97**
3.7 Experimental work **98**
3.7.1 Experimental setup 98
3.7.2 Results 99
3.8 Discussion **100**

3.1 Introduction

In the previous chapter, we studied a probabilistic NMF model for integer data, based on the Poisson distribution. We will now turn to a model for continuous non-negative data. We will assume the following observation model

$$v_{fn} \sim \text{Exp}\left(\frac{1}{[\mathbf{WH}]_{fn}}\right), \tag{3.1}$$

where "Exp" denotes the exponential distribution parametrized by its rate. As already stated in Section 1.3.2, this model is equivalent to minimizing the Itakura-Saito divergence between $\mathbf{V}$ and $\mathbf{WH}$, and we will discuss in Section 3.2 why this generative model is particularly well-suited for the analysis of power spectrograms. The use of such a likelihood was first proposed in Févotte et al. (2009), and has since become a classical model in audio signal processing. The authors considered the plain maximum likelihood estimation of $\mathbf{W}$ and $\mathbf{H}$, as well as a MAP estimation in a Bayesian variant. This model has also been tackled in Hoffman et al. (2010) in a Bayesian non-parametric setting to alleviate the choice of the factorization rank K.

Dikmen and Févotte (2011) considered a semi-Bayesian setting with generalized Gamma priors on $\mathbf{H}$ and tackled maximum marginal likelihood estimation. They observed a similar self-regularization phenomenon to the one described in the Gamma-Poisson model. However, it should be noted that they only proposed variational inference schemes. As such, we may wonder whether the observed self-regularization phenomenon is a by-product of the inference method in this model.

In this chapter, for reasons that will be apparent later, we tackle MMLE in the specific semi-Bayesian model with independent inverse Gamma priors on $\mathbf{H}$, i.e., a special case of the prior considered in Dikmen and Févotte (2011). We aim at carrying out the same analysis as the one done for the GaP model. We provide the following contributions:

- We show how the generative model that will be referred to as IGCN can be rewritten free of $\mathbf{H}$;
- We obtain an expression for the marginal likelihood with an intractable integral, but which still reveals a penalty term on $\mathbf{W}$;
- We provide three novel MCEM algorithms for the optimization of the marginal likelihood;
- We conduct experimental work on a real audio decomposition task. However, the learned dictionaries do not exhibit sparsity. Furthermore, we observe that MMLE behaves similarly to IS-NMF, i.e., plain joint maximum likelihood estimation in the frequentist model, making its benefits less striking in this case.

The rest of the chapter is organized as follows. Section 3.2 discusses model considerations, and introduces the so-called IGCN model. Section 3.3 proposes a new formulation of IGCN in which $\mathbf{H}$ has been marginalized out. This leads to an expression of the marginal likelihood

discussed in Section 3.4. Novel MCEM algorithms are derived in Section 3.5, and considerations regarding the estimation of audio sources are described in Section 3.6. Experimental work is conducted in Section 3.7, and we conclude by a discussion in Section 3.8.

3.2 Model

3.2.1 Observation model

As it turns out, the generative model of Eq. (3.1) is equivalent to

$$x_{fn} \sim \mathcal{CN}(0, [\mathbf{WH}]_{fn}), \tag{3.2}$$
$$v_{fn} = |x_{fn}|^2, \tag{3.3}$$

where $\mathcal{CN}$ denotes the *circularly-symmetric*. This amounts to saying that the real and imaginary parts of x_{fn} are independent and distributed as

$$\mathrm{Re}(x_{fn}) \sim \mathcal{N}(0, \frac{1}{2}[\mathbf{WH}]_{fn}), \quad \mathrm{Im}(x_{fn}) \sim \mathcal{N}(0, \frac{1}{2}[\mathbf{WH}]_{fn}). \tag{3.4}$$

For more detailed considerations about complex normal distributions, we refer the reader to Appendix 3.A. The equivalence between the two generative models is obtained by remembering that the sum of two independent, squared, standard normal distributions is a chi-squared distribution with two degrees of freedom, i.e., an exponential distribution.

Note that this model also has a composite structure, thanks to the superposition property of the normal distribution. We may further augment the model as

$$c_{fkn} \sim \mathcal{CN}(0, w_{fk}h_{kn}), \tag{3.5}$$
$$x_{fn} = \sum_k c_{fkn}, \tag{3.6}$$
$$v_{fn} = |x_{fn}|^2. \tag{3.7}$$

In audio signal processing, we are interested in the short-time Fourier transform (STFT) of the signal, that is when the signal has been divided in short, overlapping segments of equal lengths over which the Fourier transform is computed. This results in a complex matrix, $\mathbf{X}$. An exponential likelihood for the power spectrogram ($\mathbf{V} = |\mathbf{X}|^2$) therefore underlies a Gaussian composite model for the complex spectrogram. More precisely, it is modeled as a pure (i.e., without noise) sum of K zero-mean components, whose variances are rank-one. These are reasonable assumptions from a physical point of view. Finally, note that this vindicates the choice of using the power spectrogram as the non-negative observation matrix rather than the magnitude spectrogram ($\mathbf{V} = |\mathbf{X}|$).

3.2.2 Working with complex data

Assuming that we have chosen a prior distribution on $\mathbf{H}$ (see the following subsection which discusses this choice), our goal is to maximize the marginal likelihood

$$p(\mathbf{V};\mathbf{W}) = \int_{\mathbf{H}} p(\mathbf{V}|\mathbf{H};\mathbf{W})p(\mathbf{H})\mathrm{d}\mathbf{H}. \tag{3.8}$$

However, as can be easily derived, we have the following proportional relation

$$p(\mathbf{V};\mathbf{W}) \propto p(\mathbf{X};\mathbf{W}), \tag{3.9}$$

and as such, we will rather focus on the study of $p(\mathbf{X};\mathbf{W})$, since the composite structure brought by the complex normal likelihood, identical to the one with a Poisson likelihood, facilitates derivations.

Indeed, we have for the p.d.f. of x_{fn} in Eq. (3.2)

$$f(x_{fn}) = \frac{1}{\pi[\mathbf{WH}]_{fn}} \exp\left(-\frac{|x_{fn}|^2}{[\mathbf{WH}]_{fn}}\right), \tag{3.10}$$

and as such, we can write

$$p(\mathbf{X};\mathbf{W},\mathbf{H}) = \prod_{f,n} \frac{1}{\pi[\mathbf{WH}]_{fn}} \exp\left(-\frac{|x_{fn}|^2}{[\mathbf{WH}]_{fn}}\right) \tag{3.11}$$

$$= \pi^{-FN} \prod_{f,n} \frac{1}{[\mathbf{WH}]_{fn}} \exp\left(-\frac{v_{fn}}{[\mathbf{WH}]_{fn}}\right) \tag{3.12}$$

$$= \pi^{-FN} p(\mathbf{V};\mathbf{W},\mathbf{H}). \tag{3.13}$$

Since Eq. (3.13) holds, by multiplication and integration, we straightforwardly obtain Eq. (3.9).

3.2.3 Prior distribution

We consider independent inverse Gamma priors on $\mathbf{H}$. As such, the studied model becomes

$$h_{kn} \sim \mathcal{IG}(\alpha_k, \beta_k), \tag{3.14}$$

$$c_{fkn}|h_{kn} \sim \mathcal{CN}(0, w_{fk}h_{kn}), \tag{3.15}$$

$$x_{fn} = \sum_k c_{fkn}, \tag{3.16}$$

where $\mathcal{IG}$ denotes the inverse Gamma distribution parametrized with its shape and scale. This choice is mainly motivated by the fact that the inverse Gamma distribution is conjugate with the normal distribution with known mean and unknown variance. The p.d.f. and moments of the inverse Gamma distribution are recalled in Appendix 3.A. We shall refer to the generative model defined by Eqs. (3.14)-(3.15)-(3.16) as the "IGCN" (standing for inverse Gamma complex normal) model. In the following, $\boldsymbol{\alpha}$ denotes the set $\{\alpha_1,\ldots,\alpha_K\}$ and $\boldsymbol{\beta}$ denotes the set $\{\beta_1,\ldots,\beta_K\}$.

3.2.4 Objective function

Treating $\boldsymbol{\alpha}$ as fixed hyperparameters, and noting a similar scale-invariance for $\boldsymbol{\beta}$ to the one described at the end of Section 2.4.1[1], MMLE in the IGCN model amounts to the minimization of

$$\mathcal{L}(\mathbf{W}) \overset{\text{def}}{=} -\log p(\mathbf{X};\mathbf{W}) \tag{3.17}$$

$$= -\log \int_{\mathbf{H}} p(\mathbf{X}|\mathbf{H};\mathbf{W})p(\mathbf{H})\mathrm{d}\mathbf{H}. \tag{3.18}$$

3.3 Marginalization of H

We now show how the IGCN model can be rewritten free of the variables $\mathbf{H}$. Similarly to the previous chapter, h_{kn} can be marginalized out from Eqs. (3.14)-(3.15), thanks to the conjugacy between distributions, leading to a closed-form distribution for the components $\mathbf{c}_{kn}$. More precisely, we have

$$p(\mathbf{c}_{kn};\mathbf{w}_k,\alpha_k,\beta_k)$$

$$= \int_{\mathbb{R}_+} p(c_{1kn},\dots,c_{Fkn}|h_{kn})p(h_{kn})\mathrm{d}h_{kn} \tag{3.19}$$

$$= \int_{\mathbb{R}_+} \left(\prod_f p(c_{fkn}|h_{kn})\right) p(h_{kn})\mathrm{d}h_{kn} \tag{3.20}$$

$$= \int_{\mathbb{R}_+} \left(\prod_f \frac{1}{\pi w_{fk}h_{kn}} \exp\left(-\frac{|c_{fkn}|^2}{w_{fk}h_{kn}}\right)\right) \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} \left(\frac{1}{h_{kn}}\right)^{\alpha_k+1} \exp\left(-\frac{\beta_k}{h_{kn}}\right) \mathrm{d}h_{kn} \tag{3.21}$$

$$= \frac{1}{\pi^F \prod_f w_{fk}} \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} \frac{\Gamma(\alpha_k+F)}{\left(\beta_k + \sum_f \frac{|c_{fkn}|^2}{w_{fk}}\right)^{\alpha_k+F}}. \tag{3.22}$$

This distribution can be identified as a complex multivariate Student's t-distribution[2] of dimension F (Yoshii et al., 2016), with parameters

$$\nu = 2\alpha_k,\ \boldsymbol{\mu} = \mathbf{0}_F,\ \boldsymbol{\Sigma} = \mathrm{Diag}(\boldsymbol{\lambda}_k), \tag{3.23}$$

where $\boldsymbol{\lambda}_k \in \mathbb{R}_+^F = \frac{\beta_k}{\alpha_k}\mathbf{w}_k$, and $\mathbf{0}_F$ denotes a vector of zeros of size F. More details about the distribution can be found in Appendix 3.A.

Therefore, we immediately obtain the following result:

[1] As it turns out, in this Chapter $\boldsymbol{\beta}$ is a *scale* parameter (it was a rate parameter in Chapter 2), so we would write instead $p(\mathbf{V};\mathbf{W}\boldsymbol{\Lambda}^{-1},\boldsymbol{\alpha},\boldsymbol{\beta}\boldsymbol{\Lambda}) = p(\mathbf{V};\mathbf{W},\boldsymbol{\alpha},\boldsymbol{\beta})$.

[2] To have a better intuition about where this comes from, we recall that the scalar real-valued Student's t-distribution arises when the variance of a normal distribution is assumed to be inverse Gamma distributed.

Theorem 3.1. IGCN can be rewritten as follows

$$\mathbf{c}_{kn} \sim \mathcal{CT}_{2\alpha_k}\left(\mathbf{0}_F, \mathrm{Diag}(\boldsymbol{\lambda}_k)\right), \tag{3.24}$$

$$\mathbf{v}_n = \sum_k \mathbf{c}_{kn}, \tag{3.25}$$

where $\mathcal{CT}_\nu\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ denotes the multivariate complex Student's t-distribution with degrees of freedom ν, location parameter $\boldsymbol{\mu}$ and scale matrix $\boldsymbol{\Sigma}$.

IGCN may therefore be interpreted as a composite complex Student's t model, in which each source is parametrized by $\mathbf{w}_k$, α_k and β_k.

3.4 Marginal likelihood

3.4.1 Analytical expression

We now focus on obtaining an expression for the marginal likelihood. Defining a set $\mathcal{C}$ of admissible components,

$$\mathcal{C} = \{\mathbf{C} \in \mathbb{C}^{F\times K\times N} \,|\, \forall (f,n), \sum_k c_{fkn} = x_{fn}\}, \tag{3.26}$$

we wish to marginalize the variables $\mathbf{C}$. However, a major difference with what was done in the Gamma-Poisson model must now be discussed. In the previous chapter, as we were dealing with integer variables, $\mathcal{C}$ was a finite set, yielding a semi-explicit closed form expression of the marginal likelihood with a finite sum. In the IGCN model, we are dealing with continuous variables, and as such the finite sum translates here to an intractable integral. More precisely, writing

$$p(\mathbf{X};\mathbf{W}) = \int_{\mathbf{C}\in\mathcal{C}} p(\mathbf{C};\mathbf{W})\mathrm{d}\mathbf{C} = \int_{\mathbf{C}\in\mathcal{C}} \prod_{k,n} p(\mathbf{c}_{kn};\mathbf{w}_k)\mathrm{d}\mathbf{C}, \tag{3.27}$$

and replacing $p(\mathbf{c}_{kn};\mathbf{w}_k)$ by its expression (see Eq. (3.22)), we obtain

$$p(\mathbf{X};\mathbf{W}) = \int_{\mathbf{C}\in\mathcal{C}} \prod_{k,n} \left(\frac{1}{\pi^F \prod_f w_{fk}} \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} \frac{\Gamma(\alpha_k + F)}{\left(\beta_k + \sum_f \frac{|c_{fkn}|^2}{w_{fk}}\right)^{\alpha_k+F}} \right) \mathrm{d}\mathbf{C} \tag{3.28}$$

$$= \left(\prod_{k,n} \frac{\beta_k^{\alpha_k}}{\pi^F} \frac{\Gamma(\alpha_k+F)}{\Gamma(\alpha_k)} \frac{1}{\prod_f w_{fk}} \right) \int_{\mathbf{C}\in\mathcal{C}} \prod_{k,n} \frac{1}{\left(\beta_k + \sum_f \frac{|c_{fkn}|^2}{w_{fk}}\right)^{\alpha_k+F}} \mathrm{d}\mathbf{C}. \tag{3.29}$$

3.4.2 Self-regularization

The negative log-likelihood may be expressed as

$$-\frac{1}{N}\mathcal{L}(\mathbf{W}) = \sum_{f,k} \log\left(w_{fk}\right) - \frac{1}{N}\log \int_{\mathbf{C}\in\mathcal{C}} \prod_{k,n} \frac{1}{\left(\beta_k + \sum_f \frac{|c_{fkn}|^2}{w_{fk}}\right)^{\alpha_k+F}} \mathrm{d}\mathbf{C} + \mathrm{cst}, \tag{3.30}$$

with

$$\mathrm{cst} = -\sum_k \log\left(\frac{\beta_k^{\alpha_k}\Gamma(\alpha_k+F)}{\pi^F\Gamma(\alpha_k)}\right). \tag{3.31}$$

The first term depends only on $\mathbf{W}$, and acts like a regularization parameter on $\mathbf{W}$. The second term is an interaction term between $\mathbf{W}$ and $\mathbf{V}$ (through $\mathbf{C}$), and acts as a data-fitting term. The final term is a constant. It is therefore expected that this revealed regularization term will promote local sparsity in the estimated dictionaries, as opposed to group-sparsity which we could observe in the GaP model.

3.5 Optimization algorithms

3.5.1 Expectation-Minimization

We now focus on the task of optimizing Eq. (3.18). The structure of the problem being identical to the problem studied in the Gamma-Poisson model, we consider similar EM algorithms. In particular, we consider the same three variants, based on three different choices for the set of latent variables. We summarize below the three functionals that are iteratively minimized, given the current estimate of the dictionary denoted $\tilde{\mathbf{W}}$.

EM-CH.

$$Q_{\mathrm{CH}}(\mathbf{W};\tilde{\mathbf{W}}) = -\int_{\mathbf{C},\mathbf{H}} \log p(\mathbf{C},\mathbf{H};\mathbf{W}) p(\mathbf{C},\mathbf{H}|\mathbf{X};\tilde{\mathbf{W}}) \mathrm{d}\mathbf{C}\mathrm{d}\mathbf{H}. \tag{3.32}$$

EM-H.

$$Q_{\mathrm{H}}(\mathbf{W};\tilde{\mathbf{W}}) = -\int_{\mathbf{H}} \log p(\mathbf{X},\mathbf{H};\mathbf{W}) p(\mathbf{H}|\mathbf{X};\tilde{\mathbf{W}}) \mathrm{d}\mathbf{H}. \tag{3.33}$$

EM-C.

$$Q_{\mathrm{C}}(\mathbf{W};\tilde{\mathbf{W}}) = -\int_{\mathbf{C}} \log p(\mathbf{C};\mathbf{W}) p(\mathbf{C}|\mathbf{X};\tilde{\mathbf{W}}) \mathrm{d}\mathbf{C}. \tag{3.34}$$

All these algorithms are novel, and in particular have not been considered in Dikmen and Févotte (2011). They instead considered a variational approach, which consisted in an iterative variational bound construction and optimization. However, this does not qualify as an MM approach, because the constructed bound is never tight to the objective function,

therefore failing to ensure the decrease of the objective function[3]. This is in contrast with the considered EM algorithms. Our approach conceptually asymptotically yield a critical point of the objective function, something the approach of Dikmen and Févotte (2011) cannot guarantee.

In all three cases, the posterior of the latent variables involved is intractable and neither are the integrals involved in Equations (3.32)-(3.33)-(3.34). To overcome this problem, we resort to Monte Carlo EM (MCEM) (Wei and Tanner, 1990).

3.5.2 Monte Carlo E-step

A Gibbs sampler procedure can be devised to yield samples $p(\mathbf{C}, \mathbf{H}|\mathbf{X}; \tilde{\mathbf{W}})$, which also returns samples from the marginal posterior distributions $p(\mathbf{C}|\mathbf{X}; \tilde{\mathbf{W}})$ and $p(\mathbf{H}|\mathbf{X}; \tilde{\mathbf{W}})$ at convergence.

In the IGCN model, both conditional distributions $p(\mathbf{H}|\mathbf{C}; \tilde{\mathbf{W}})$ and $p(\mathbf{C}|\mathbf{H}, \mathbf{X}; \tilde{\mathbf{W}})$ are known, however sampling from the latter requires a slightly more involved procedure. For $p(\mathbf{H}|\mathbf{C}; \tilde{\mathbf{W}})$, we have

$$p(\mathbf{H}|\mathbf{C}; \tilde{\mathbf{W}}) = \prod_{k,n} p(h_{kn}|\mathbf{c}_{kn}; \tilde{\mathbf{w}}_k), \tag{3.35}$$

where

$$p(h_{kn}|\mathbf{c}_{kn}; \tilde{\mathbf{w}}_k) = \mathcal{IG}\left(\alpha_k + F,\ \beta_k + \sum_f \frac{|c_{fkn}|^2}{\tilde{w}_{fk}}\right). \tag{3.36}$$

As for the other conditional $p(\mathbf{C}|\mathbf{H}, \mathbf{X}; \tilde{\mathbf{W}})$, we have

$$p(\mathbf{C}|\mathbf{H}, \mathbf{X}; \tilde{\mathbf{W}}) = \prod_{f,n} p(\underline{\mathbf{c}}_{fn}|\mathbf{h}_n, v_{fn}; \underline{\tilde{\mathbf{w}}}_f), \tag{3.37}$$

where

$$p(\underline{\mathbf{c}}_{fn}|\mathbf{h}_n, v_{fn}; \underline{\tilde{\mathbf{w}}}_f) = \mathcal{CN}(\boldsymbol{\mu}_{fn}, \boldsymbol{\Sigma}_{fn}) \tag{3.38}$$

where the parameters of the complex normal distribution are defined as

$$\boldsymbol{\mu}_{fn} = \frac{x_{fn}}{\sum_k \rho_{fkn}} \boldsymbol{\rho}_{fn}, \quad \boldsymbol{\Sigma}_{fn} = \mathrm{diag}(\boldsymbol{\rho}_{fn}) - \frac{\boldsymbol{\rho}_{fn}\boldsymbol{\rho}_{fn}^{\mathrm{T}}}{\sum_k \rho_{fkn}}, \tag{3.39}$$

with the notations

$$\rho_{fkn} = w_{fk} h_{kn}, \quad \boldsymbol{\rho}_{fn} = [\rho_{f1n}, \ldots, \rho_{fKn}]^{\mathrm{T}}. \tag{3.40}$$

All derivations are given in Appendix 3.B. However, note that the covariance matrix $\boldsymbol{\Sigma}_{fn}$ is not full-rank (it is more precisely of rank $K-1$). When this is the case, the (complex)

[3] Note that very recent works show that variational Bayesian methods can lead to a consistant estimation of the parameters under certain assumptions, see for instance Alquier and Ridgway (2017).

Algorithm 3: Sampling from $p(\mathbf{c}|x, \boldsymbol{\rho})$

Input: $x \in \mathbb{C}$, $\boldsymbol{\rho} \in \mathbb{C}^K$

$r = x$
$t = \sum_{k=1}^{K} \rho_k$
for $k = 1, ..., K-1$ **do**
  # Sample as Eq.(3.41)
  $u \sim \mathcal{CN}(0, 1)$
  $c_k = \sqrt{\rho_k(1 - \frac{\rho_k}{t})}\, u + \frac{r}{t}\rho_k$
  # Update sampling parameters
  $r = x - c_k$
  $t = t - \rho_k$
end
$c_K = r$

Output: $\mathbf{c} \in \mathbb{C}^K$ drawn from $p(\mathbf{c}|x, \boldsymbol{\rho})$

normal distribution is referred to as a *singular* (complex) normal distribution. Sampling from a singular normal distribution is not straightforward since we cannot use the Cholesky decomposition routine.

In this case, we resort to a sequential sampling of the first $K-1$ marginal distributions. Dropping the indices f and n for enhanced readability, this amounts to sequentially sampling the first $K-1$ components c_k as

$$c_k \sim \mathcal{CN}\left(\frac{x - \sum_{l=1}^{k-1} c_l}{\sum_{l=k}^{K} \rho_l}\rho_k, \rho_k\left(1 - \frac{\rho_k}{\sum_{l=k}^{K} \rho_l}\right)\right), \tag{3.41}$$

and set $c_K = 1 - \sum_{k=1}^{K-1} c_k$. This procedure is summed up in Algorithm 3.

In the Gibbs sampler of the Gamma-Poisson model, sampling $\mathbf{C}$ boiled down to sampling as many multinomial distribution as non-zero values of $\mathbf{V}$ (recall the data were integer-valued). There is unfortunately no similar skipping trick in the IGCN model, and we are doomed to sample $F \times N$ multivariate complex normal distributions. The whole Gibbs sampling procedure is summarized in Algorithm 4.

3.5.3 Monte Carlo M-step

We consider a set of J samples $\{\mathbf{C}^{(j)}, \mathbf{H}^{(j)}\}$ returned by the Gibbs sampler (after burn-in). All derivation details regarding the optimization step can be found in Appendix 3.C.

Algorithm 4: Gibbs sampler in the IGCN model

Input: Complex matrix $\mathbf{X}$, current value of the dictionary $\tilde{\mathbf{W}}$

Generate initial state $\mathbf{H}^{(0)}$
for $j = 1, ..., J$ **do**
  # Sample $\mathbf{C}$ given $\mathbf{H}$
  for $f = 1, ..., F$ **do**
    for $n = 1, ..., N$ **do**
      $\underline{\mathbf{c}}_{fn}^{(j)} \sim \mathcal{CN}(\boldsymbol{\mu}_{fn}^{(j-1)}, \boldsymbol{\Sigma}_{fn}^{(j-1)})$ # See Algorithm 3
    end
  end
  # Sample $\mathbf{H}$ given $\mathbf{C}$
  for $k = 1, ..., K$ **do**
    for $n = 1, ..., N$ **do**
      $h_{kn}^{(j)} \sim \mathcal{IG}(\alpha_k + F,\ \beta_k + \sum_f \frac{|c_{fkn}^{(j)}|^2}{\tilde{w}_{fk}})$
    end
  end
end

Output: J samples asymptotically from the joint posterior $p(\mathbf{C}, \mathbf{H}|\mathbf{X}; \tilde{\boldsymbol{\theta}})$

EM-CH. Minimization of the MC approximation of Q_{CH} in Eq. (3.32) w.r.t. $\mathbf{W}$ leads to the closed-form update

$$w_{fk}^{\text{MCEM-CH}} = \frac{1}{NJ} \sum_{n,j} \frac{|c_{fkn}^{(j)}|^2}{h_{kn}^{(j)}}. \tag{3.42}$$

EM-H. Using the standard majorization of the IS divergence, the following multiplicative update decreases the MC approximation of Q_{H} in Eq. (3.33) at every iteration.

$$w_{fk}^{\text{MCEM-H}} = \tilde{w}_{fk} \sqrt{\frac{\sum_{j,n} h_{kn}^{(j)} v_{fn} [\tilde{\mathbf{W}}\mathbf{H}^{(j)}]_{fn}^{-2}}{\sum_{j,n} h_{kn}^{(j)} [\tilde{\mathbf{W}}\mathbf{H}^{(j)}]_{fn}^{-1}}}. \tag{3.43}$$

EM-C. The MC approximation of Q_{C} in Eq. (3.34) writes

$$\hat{Q}^{\text{C}}(\mathbf{W}) \overset{\text{c}}{=} \frac{1}{J} \sum_{j,k,n} \left(\sum_f \log(w_{fk}) + (\alpha_k + F) \log \left(\beta_k + \sum_f \frac{|c_{fkn}^{(j)}|^2}{w_{fk}} \right) \right). \tag{3.44}$$

To optimize this function w.r.t. $\mathbf{W}$, we resort to an MM scheme. In particular, we majorize the logarithm in the second term, which is always below its tangents. The minimization of

this auxiliary function can be done in closed form, leading to the following update

$$w_{fk} = \frac{\alpha_k + F}{NJ} \sum_{j,n} \frac{|c_{fkn}^{(j)}|^2}{\beta_k + \sum_f \frac{|c_{fkn}^{(j)}|^2}{\tilde{w}_{fk}}}. \tag{3.45}$$

3.6 Estimating audio sources

In the context of audio decomposition, the factors $\mathbf{W}$ and $\mathbf{H}$ are not the final purpose of the learning process. Once having point estimates of the factors, we would like to be able to reconstruct the corresponding sources in the temporal domain.

Point estimate of H

Consider an estimate $\hat{\mathbf{W}}$ obtained via MMLE. If we are seeking a point estimate of $\mathbf{H}$, a possible choice is the maximum a posteriori (MAP) estimate, which corresponds to the minimization of

$$\begin{aligned} C_{\mathrm{MAP}}(\mathbf{H}) &\overset{\mathrm{def}}{=} -\log p(\mathbf{H}|\mathbf{V}; \hat{\mathbf{W}}) && (3.46)\\ &\overset{\mathrm{c}}{=} -\log p(\mathbf{V}|\mathbf{H}; \hat{\mathbf{W}}) - \log p(\mathbf{H}) && (3.47)\\ &\overset{\mathrm{c}}{=} D_{\mathrm{IS}}(\mathbf{V}|\hat{\mathbf{W}}\mathbf{H}) + \sum_{k,n} \left((\alpha_k + 1) \log h_{kn} + \frac{\beta_k}{h_{kn}} \right). && (3.48) \end{aligned}$$

Using the standard majorization of the IS divergence (Févotte and Idier, 2011; Dikmen and Févotte, 2011), we obtain the following bound tight at $\mathbf{H} = \tilde{\mathbf{H}}$

$$C_{\mathrm{MAP}}(\mathbf{H}) \leq \sum_{k,n} \left(\frac{p_{kn}}{h_{kn}} + q_{kn} h_{kn} \right) + \sum_{k,n} \left((\alpha_k + 1) \log h_{kn} + \frac{\beta_k}{h_{kn}} \right), \tag{3.49}$$

where

$$p_{kn} = \tilde{h}_{kn}^2 \sum_f \frac{\hat{w}_{fk} v_{fn}}{[\hat{\mathbf{W}}\tilde{\mathbf{H}}]_{fn}^2}, \quad q_{kn} = \sum_f \frac{\hat{w}_{fk}}{[\hat{\mathbf{W}}\tilde{\mathbf{H}}]_{fn}}. \tag{3.50}$$

The minimization of the auxiliary function w.r.t. h_{kn} yields a degree 2 polynomial with exactly one non-negative root given by

$$h_{kn} = \frac{-(\alpha_k + 1) + \sqrt{(\alpha_k + 1)^2 + 4 q_{kn} (\beta_k + p_{kn})}}{2 q_{kn}}. \tag{3.51}$$

Point estimate of C

We now turn to the problem of estimating the components. In the frequency domain, the k^{th} source corresponds to the $F \times N$ spectrogram $\mathbf{C}_k = \{c_{fkn}\}_{f,n}$. As it turns out, the

posterior mean of $\mathbf{C}$, given $\hat{\mathbf{W}}$, $\hat{\mathbf{H}}$ and $\mathbf{X}$, is obtained with the so-called Wiener filtering equation (Févotte et al., 2009)

$$\hat{c}_{fkn} = \frac{\hat{w}_{fk}\hat{h}_{kn}}{[\hat{\mathbf{W}}\hat{\mathbf{H}}]_{fn}} x_{fn}. \tag{3.52}$$

Once this estimate has been computed, we simply obtain the audio sources by inversing the STFT of each of the K spectrograms.

3.7 Experimental work

In this section, we conduct experimental work on a real audio decomposition task. Python implementations of the three MCEM algorithms are available on GitHub.

3.7.1 Experimental setup

We consider the piano dataset first used in Févotte et al. (2009). This dataset is a real recording of a short piano sequence comprising four different notes. The notes are all played together in the first measure, and each possible pair of notes is then played in the next six measures. The time-domain recording is displayed on Figure 3.1. Using an analysis window of 1024 samples (46ms) with 50% overlap, we end up with a spectrogram of $F = 513$ frequency bins and $N = 676$ time frames. This log-spectrogram is displayed on Figure 3.1 as well.

We then proceed to estimate $\mathbf{W}$ with MCEM-CH, setting the hyperparameters $\alpha_k = 0.1$, $\beta_k = 1$ and $K = 10$. MCEM-C and MCEM-H result in similar performances and are not reported here. The Gibbs sampling procedure being extremely computationally intensive, we resort here to a "cheap" MCEM algorithm where only 10 samples are generated at each EM iteration with no burn-in[4] (i.e., $J = 10$). The algorithm is run for 10000 EM iterations, and is initialized by taking random columns of $\mathbf{V}$. Setting the estimate of $\mathbf{W}$ to be the last iterate of the algorithm, $\mathbf{H}$ and the audio sources are estimated with the procedures described in Section 3.6.

We compare the performance of our method with the performance of the IS-NMF method (i.e., NMF with the IS divergence). The corresponding multiplicative update rules can be found in Eq. (1.14) by taking $\beta = 0$. IS-NMF is a standard baseline when it comes to audio decomposition (Févotte et al., 2009). We have ran it 5 times from different random initializations and selected the solution with lowest objective value. The audio sources are also estimated with the procedure described in Section 3.6.

[4] Generating 150 samples and discarding the first 50 for burn-in, as was was done in Chapter 2 with the `Taste Profile` dataset leads to a CPUtime of 90 seconds per iteration. The presented cheaper version roughly divides this number by 15.

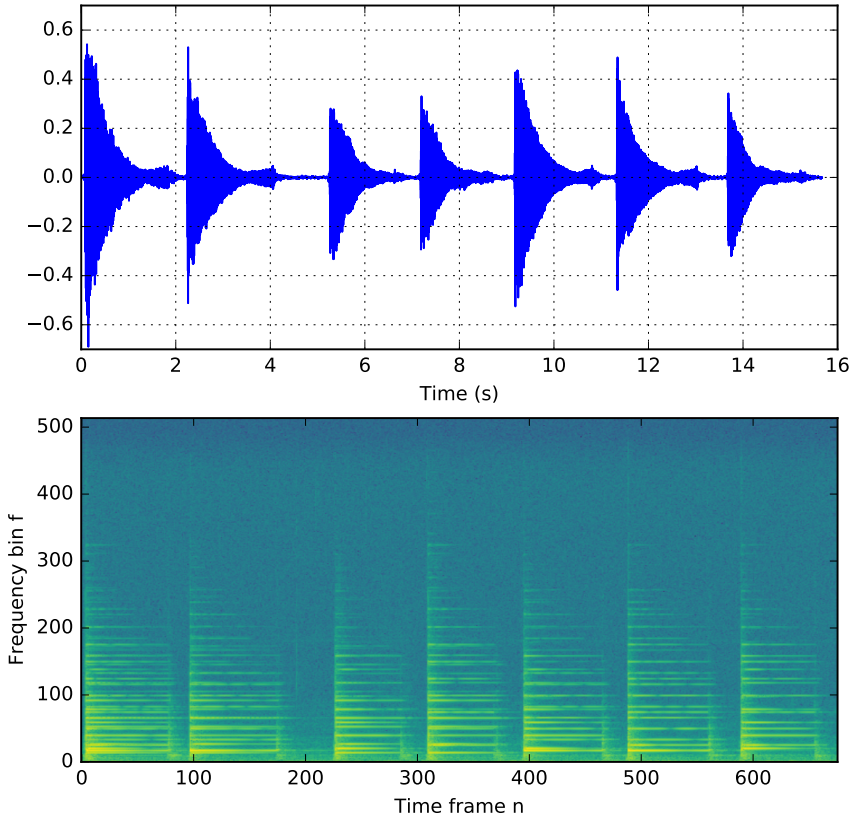


Figure 3.1: The piano dataset. From top to bottom:(a) Time-domain recorded signal. (b) Log-power spectrogram $\mathbf{V} = |\mathbf{X}|^2$

3.7.2 Results

Figure 3.2 displays the norm in $\log_{10}$ scale of the $K = 10$ columns of the iterates w.r.t. CPU time, in hours. As we can see, even after 10000 iterations[5], the algorithm does not seem to fully have converged yet. Moreover, it seems prone to label switching.

We now investigate the returned dictionaries and reconstructed audio components. For both methods, the $K = 10$ components are ordered by decreasing value of their variance, computed from the reconstructed time-domain components.

Figure 3.3 displays the columns of $\mathbf{W}$ returned by IS-NMF (in blue, left column) and those returned by MMLE (in red, right column). The columns are represented against frequency bin f, in $\log_{10}$ scale. As we can see, the dictionary returned by MMLE does not exhibit sparsity, and is very similar to the one returned by IS-NMF.

[5] As we can see, 10 000 iterations corresponds roughly to 24 hours. The CPUtime of the IS-NMF method is roughly 90 seconds.

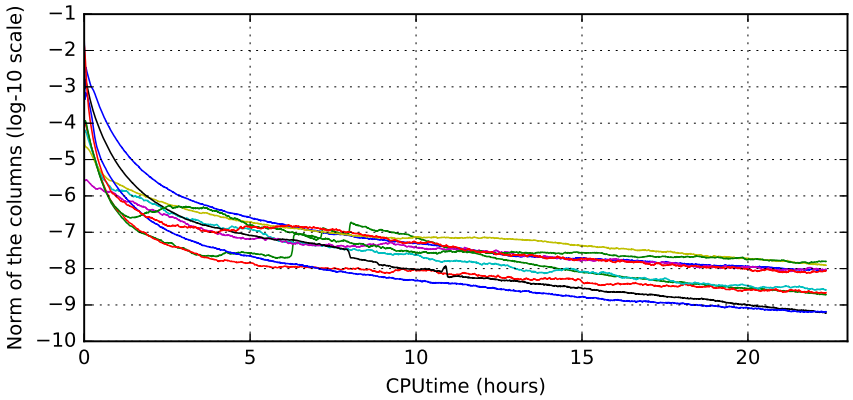


Figure 3.2: Evolution of the norm of each of the $K = 10$ columns $\mathbf{w}_k$ of the dictionary in $\log_{10}$ scale w.r.t. the number of EM iterations for MCEM-CH on the piano dataset.

This is further confirmed on Figure 3.4, which displays the reconstructed sources in the time domain returned by IS-NMF (in blue, left column), and those returned by MMLE (in red, right column). The audio accuracy performance of both methods is the same. Indeed, the first four components correspond to the four different notes of the piano sequence, the fifth component corresponds to note attacks, and the remaining components are inaudible. We would have expected the inaudible audio components returned by MMLE to have smaller variances (i.e., spectral power) than those returned by IS-NMF, thanks to the implicit regularization term revealed in Eq. (3.30). Unfortunately, we do not observe such a phenomenon here. This, in addition to the prohibitive computational cost, makes the benefits of the proposed method somewhat limited for audio signal processing.

3.8 Discussion

In this chapter, we have tackled maximum marginal likelihood estimation in the IGCN matrix factorization model. We were able to propose a new formulation of the IGCN model in which $\mathbf{H}$ has been marginalized out. Unfortunately, this new formulation did not lead to a closed-form expression of the marginal likelihood. It nonetheless revealed a penalty term indicating that MMLE in this model should induce "local" sparsity (as opposed to group-sparsity in the Gamma-Poisson model) in the estimated dictionaries[6].

Moreover, we have developed three EM algorithms for the task of optimizing the marginal likelihood in this model. This constitutes a breakthrough w.r.t. the state of the art, since the variational algorithm of Dikmen and Févotte (2011) came with no convergence guarantees.

[6]The considerations regarding the interaction between the data-fitting term and the regularization term discussed at the end of the previous chapter apply here as well.

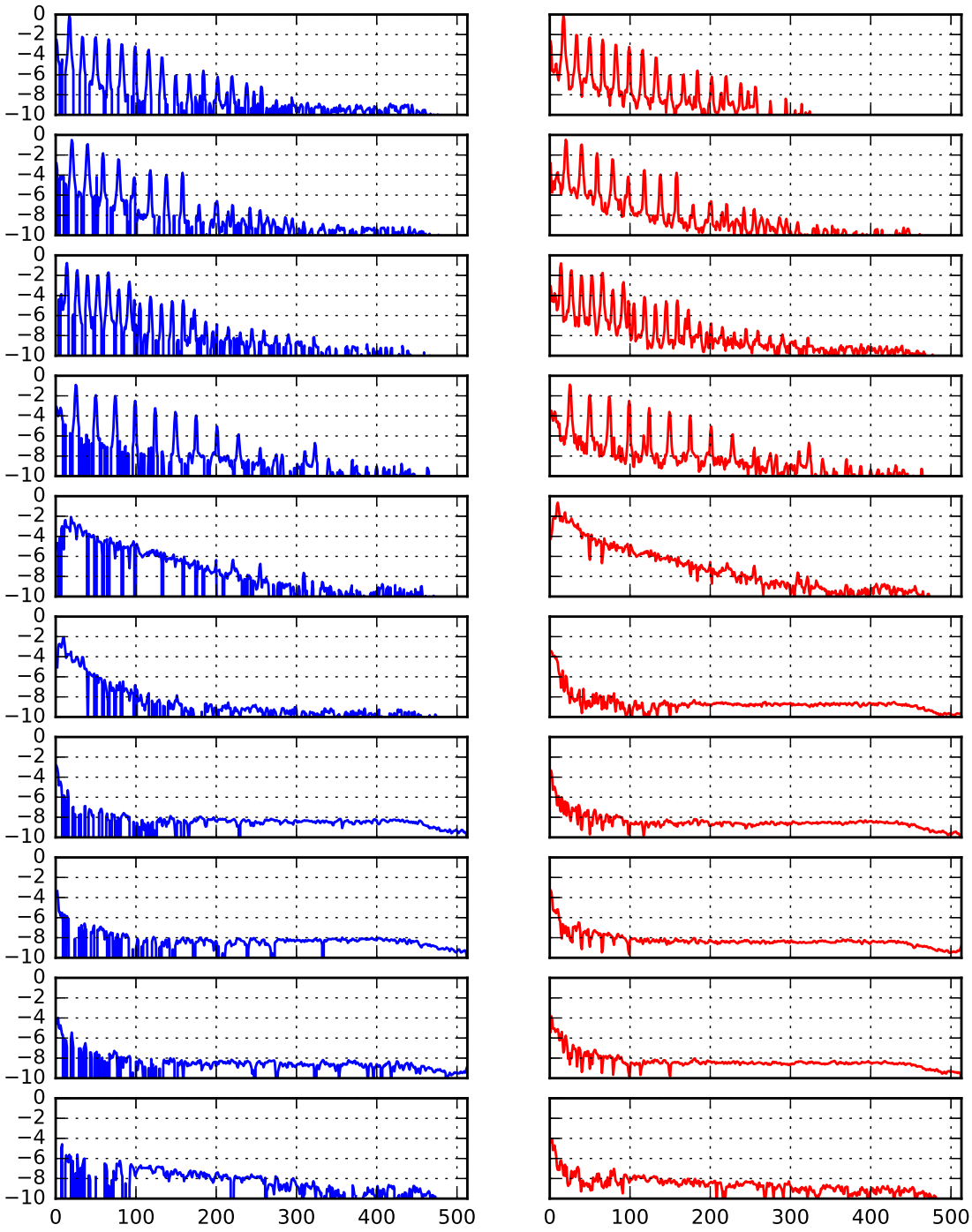


Figure 3.3: Columns of $\mathbf{W}$ in $\log_{10}$ scale w.r.t. frequency bin f. From left to right :(a) With IS-NMF (in blue). (b) With MMLE (in red). For each method, the $K = 10$ components are sorted by the decreasing variance of the associated time-domain signal.

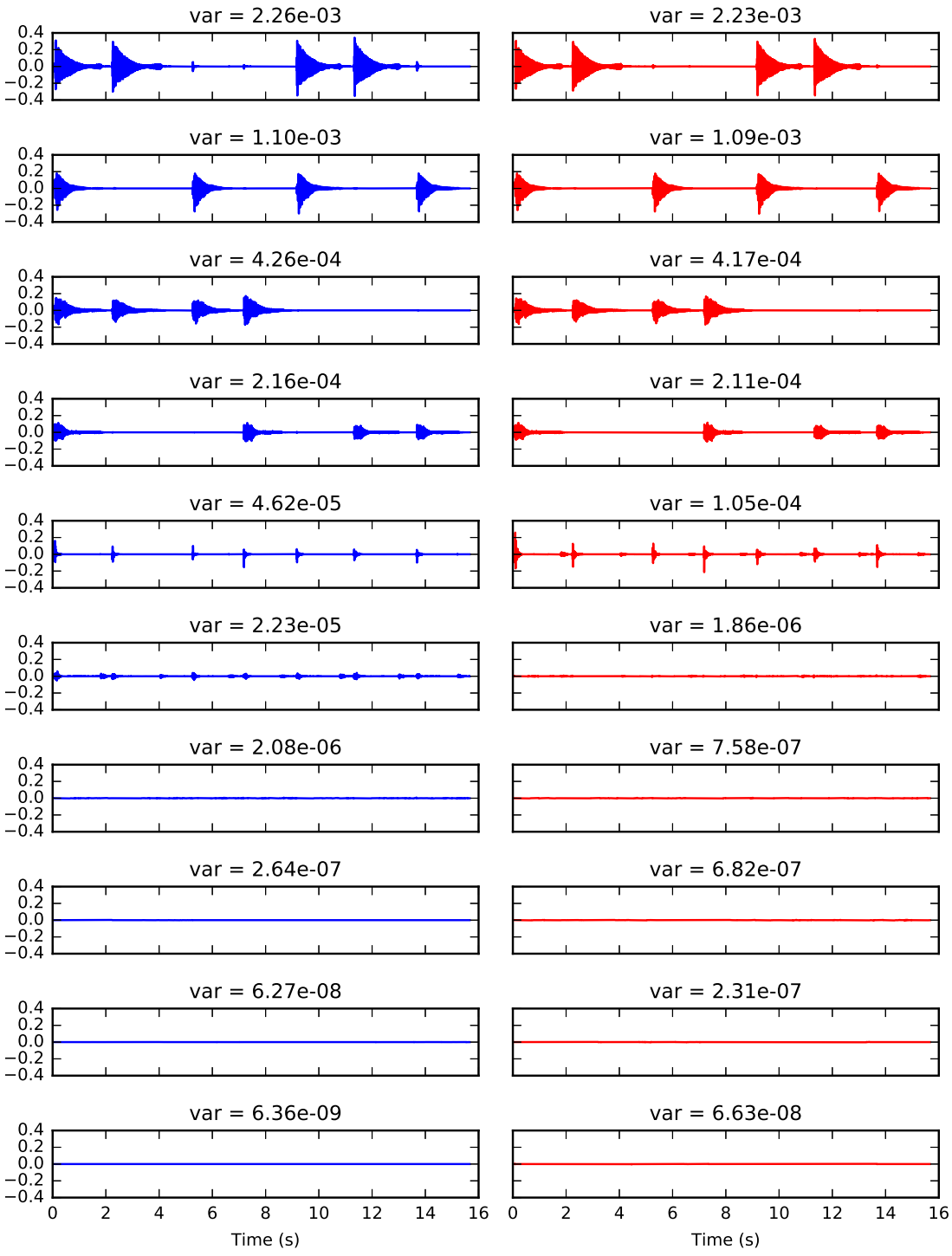


Figure 3.4: Reconstructed time-domain components. From left to right :(a) With IS-NMF (in blue). (b) With MMLE (in red). For each method, the $K = 10$ components are sorted by decreasing variance.

The experimental work on a real audio decomposition example proved challenging. Firstly, it seems that a large number of EM iterations is needed for the algorithm to converge. Secondly, the estimated audio components do not represent a significant improvement w.r.t. to the baseline IS-NMF estimation, and the self-regularization phenomenon is not striking. Thirdly, the computational cost of the proposed MCEM algorithms is prohibitive. As such, an exciting perspective would be, as in the previous chapter, to develop lighter algorithms for the optimization of the marginal likelihood.

Finally, we wonder whether the somewhat disappointing experimental results may be imputed to the choice of the prior distribution, namely the inverse Gamma distribution. Indeed, this choice was only driven by the conjugacy with the (complex) normal distribution. It would be an interesting perspective to use different prior distributions. In particular, the Gamma distribution may have been a more suitable choice of prior for $\mathbf{H}$ (it would be a particular case of the prior considered in Dikmen and Févotte (2011) as well). When the conjugacy of the model is lost, MCEM-CH and MCEM-H remain feasible, at the cost of a more involved sampling procedure. Indeed, one of the conditionals in the Gibbs sampler would be unknown, and we would have to resort to an additional Metropolis-Hastings step.

Appendices to Chapter 3

Contents

3.A Probability distributions **104**
3.A.1 Complex normal distribution 104
3.A.2 Multivariate Student's t-distribution 105
3.B Gibbs sampling of the posterior distribution **105**
3.B.1 First conditional 106
3.B.2 Second conditional 106
3.C EM algorithms **107**
3.C.1 MCEM-CH 107
3.C.2 MCEM-H 107
3.C.3 MCEM-C 108

3.A Probability distributions

3.A.1 Complex normal distribution

Most of the developments of this subsection can be found in Picinbono (1996). See also Goodman (1963) and Gallager (2012).

We begin by recalling that a complex random vector $Z \in \mathbb{C}^n$ is simply $Z = X + iY$ where X and Y are random vectors in $\mathbb{R}^n$. A complex random vector Z is said to be normal if its real and imaginary part X and Y are jointly normal.

We may therefore describe a complex random distribution by the following parameters

$$\boldsymbol{\mu} = \mathbb{E}(Z), \quad \boldsymbol{\Gamma} = \mathbb{E}((Z-\boldsymbol{\mu})(Z-\boldsymbol{\mu})^{\mathrm{H}}), \quad \mathbf{C} = \mathbb{E}((Z-\boldsymbol{\mu})(Z-\boldsymbol{\mu})^{\mathrm{T}}), \tag{3.53}$$

where $^{\mathrm{H}}$ is the Hermitian transpose[7]. $\boldsymbol{\mu} \in \mathbb{C}^n$ is the mean, $\boldsymbol{\Gamma} \in \mathbb{C}^{n\times n}$ is the covariance matrix, and $\mathbf{C} \in \mathbb{C}^{n\times n}$ is the so-called relation matrix.

[7] $(\mathbf{A}^{\mathrm{H}})_{ij} = \mathbf{A}^{\star}_{ji}$, where * denotes the conjugate.

We now focus on a very important case of this distribution, namely the circularly-symmetric complex normal distribution. Symmetric circularity means that for all $\phi \in [-\pi, \pi]$ the random vectors Z and $Ze^{j\varphi}$ have the same distribution. For the complex normal case, this amounts to having $\boldsymbol{\mu} = \mathbf{0}_n$ and $\mathbf{C} = \mathbf{0}_{n\times n}$. The random vector Z is then usually denoted

$$Z \sim \mathcal{CN}(0, \boldsymbol{\Gamma}), \tag{3.54}$$

and its p.d.f. writes, for all $\mathbf{z} \in \mathbb{C}^n$

$$p(\mathbf{z}) = \frac{1}{\pi^n \det(\boldsymbol{\Gamma})} \exp(-\mathbf{z}^{\mathrm{H}} \boldsymbol{\Gamma}^{-1} \mathbf{z}). \tag{3.55}$$

Moreover, X and Y are distributed as

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}_{2n}, \frac{1}{2}\begin{pmatrix} \mathrm{Re}(\boldsymbol{\Gamma}) & -\mathrm{Im}(\boldsymbol{\Gamma}) \\ \mathrm{Im}(\boldsymbol{\Gamma}) & \mathrm{Re}(\boldsymbol{\Gamma}) \end{pmatrix}\right). \tag{3.56}$$

Note that the circularly-symmetric complex normal distribution is almost always used in signal processing works, sometimes without a clear reference to the symmetric circularity property.

3.A.2 Multivariate Student's t-distribution

We introduce the complex multivariate Student's t-distribution. Let $\mathbf{x}$ be a p-dimensional complex random vector. The parameters of the distribution are :

- Degrees of freedom, $\nu > 0$
- Mean, $\boldsymbol{\mu} \in \mathbb{C}^p$
- Scale matrix $\boldsymbol{\Sigma} \in \mathbb{C}^{p\times p}$, positive semi-definite

And we have for its p.d.f. :

$$f(\mathbf{x}; \nu, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{2^p \Gamma(\frac{\nu}{2} + p)}{\Gamma(\frac{\nu}{2})(\nu\pi)^p \det(\boldsymbol{\Sigma})} \left(1 + \frac{2}{\nu}(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{H}} \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)^{-(\frac{\nu}{2}+p)}. \tag{3.57}$$

3.B Gibbs sampling of the posterior distribution

Given the current value of the parameter $\tilde{\mathbf{W}}$, we would like to be able to sample from the posterior of the latent variables $p(\mathbf{C}, \mathbf{H}|\mathbf{V}; \tilde{\mathbf{W}})$. In a Gibbs sampler, we are interested in the conditionals $p(\mathbf{H}|\mathbf{C}; \tilde{\mathbf{W}})$ and $p(\mathbf{C}|\mathbf{H}, \mathbf{V}; \tilde{\mathbf{W}})$.

3.B.1 First conditional

We have

$$p(\mathbf{H}|\mathbf{C};\tilde{\mathbf{W}}) \propto p(\mathbf{C}|\mathbf{H};\tilde{\mathbf{W}})p(\mathbf{H}) \tag{3.58}$$

$$\propto \prod_{f,k,n} p(c_{fkn}|h_{kn};\tilde{w}_{fk}) \prod_{k,n} p(h_{kn}) \tag{3.59}$$

$$\propto \prod_{kn} \left(\prod_{f} \left[\frac{1}{\pi w_{fk} h_{kn}} \exp\left(-\frac{|c_{fkn}|^2}{w_{fk} h_{kn}} \right) \right] \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} \left(\frac{1}{h_{kn}} \right)^{\alpha_k+1} \exp(-\frac{\beta_k}{h_{kn}}) \right) \tag{3.60}$$

$$\propto \prod_{kn} \left(\frac{1}{h_{kn}} \right)^{\alpha_k+F+1} \exp\left(-\frac{1}{h_{kn}} \left(\beta_k + \sum_f \frac{|c_{fkn}|^2}{w_{fk}} \right) \right) \tag{3.61}$$

$$= \prod_{kn} \mathcal{IG}\left(\alpha_k + F,\ \beta_k + \sum_f \frac{|c_{fkn}|^2}{w_{fk}} \right). \tag{3.62}$$

3.B.2 Second conditional

We recall the following result about the normal distribution.

Theorem 3.2. Consider the model $\mathbf{x} = \mathbf{A}\mathbf{s}$, where $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{A} \in \mathbb{R}^{d\times n}$, and $\mathbf{s} \in \mathbb{R}^n$, and further assume that $\mathbf{s} \sim \mathcal{N}(\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$. Then we have

$$p(\mathbf{s}|\mathbf{x};\mathbf{A},\boldsymbol{\mu}_s,\boldsymbol{\Sigma}_s) = \mathcal{N}(\overline{\boldsymbol{\mu}}, \overline{\boldsymbol{\Sigma}}), \tag{3.63}$$

where

$$\overline{\boldsymbol{\mu}} = \boldsymbol{\mu}_s + \boldsymbol{\Sigma}_s \mathbf{A}^{\mathrm{T}} \mathbf{G}(\mathbf{x} - \mathbf{A}\boldsymbol{\mu}_s), \tag{3.64}$$

$$\overline{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_s - \boldsymbol{\Sigma}_s \mathbf{A}^{\mathrm{T}} \mathbf{G} \mathbf{A} \boldsymbol{\Sigma}_s, \tag{3.65}$$

$$\mathbf{G} = (\mathbf{A}\boldsymbol{\Sigma}_s\mathbf{A}^{\mathrm{T}})^{-1}. \tag{3.66}$$

In particular, consider $c_1, \ldots, c_K$ K independent random variables, which are $\mathcal{N}(0, \sigma_k^2)$ distributed. Moreover, consider $v = \sum_k c_k$. Then we have $v = \mathbf{1}_K^{\mathrm{T}}\mathbf{c}$ and as such, applying Theorem 3.2 gives us that $p(\mathbf{c}|v) = \mathcal{N}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}})$ where

$$\tilde{\boldsymbol{\mu}} = \frac{v}{\sum_k \sigma_k^2}\, \boldsymbol{\sigma}, \tag{3.67}$$

$$\tilde{\boldsymbol{\Sigma}} = \mathrm{diag}(\boldsymbol{\sigma}) - \frac{\boldsymbol{\sigma}\boldsymbol{\sigma}^{\mathrm{T}}}{\sum_k \sigma_k^2}, \tag{3.68}$$

where $\boldsymbol{\sigma} = [\sigma_1^2, \ldots, \sigma_K^2]$. The result straightforwardly extends when $c_k \sim \mathcal{CN}(0, \sigma_k^2)$.

3.C EM algorithms

3.C.1 MCEM-CH

The MC approximation of Q_{CH} writes

$$\hat{Q}_{\mathrm{C}}(\mathbf{W}) = \frac{1}{J}\sum_{j} \log p(\mathbf{C}^{(j)}, \mathbf{H}^{(j)}; \mathbf{W}) \tag{3.69}$$

$$= \frac{1}{J}\sum_{j} \Big(\log p(\mathbf{C}^{(j)}|\mathbf{H}^{(j)}; \mathbf{W}) + \log p(\mathbf{H}^{(j)})\Big) \tag{3.70}$$

$$\stackrel{\mathrm{c}}{=} \frac{1}{J}\sum_{j,f,k,n} \log p(c_{fkn}^{(j)}|h_{kn}^{(j)}; w_{fk}) \tag{3.71}$$

$$\stackrel{\mathrm{c}}{=} \frac{1}{J}\sum_{j,f,k,n} \log \left(\frac{1}{\pi w_{fk} h_{kn}^{(j)}} \exp\ \left(-\frac{|c_{fkn}^{(j)}|^2}{w_{fk} h_{kn}^{(j)}}\right)\right). \tag{3.72}$$

Deriving w.r.t. w_{fk}, and setting the equation to 0 yields

$$\sum_{j,n} \left(\frac{1}{w_{fk}} - \frac{|c_{fkn}^{(j)}|^2}{w_{fk}^2 h_{kn}^{(j)}}\right) = 0, \tag{3.73}$$

which can be easily solved as

$$w_{fk} = \frac{1}{NJ}\sum_{n,j} \frac{|c_{fkn}^{(j)}|^2}{h_{kn}^{(j)}}. \tag{3.74}$$

3.C.2 MCEM-H

The MC approximation of Q_{H} writes

$$\hat{Q}_{\mathrm{H}}(\mathbf{W}) = \frac{1}{J}\sum_{j} \log p(\mathbf{X}, \mathbf{H}^{(j)}; \mathbf{W}) \tag{3.75}$$

$$= \frac{1}{J}\sum_{j} \Big(\log p(\mathbf{X}|\mathbf{H}^{(j)}; \mathbf{W}) + \log p(\mathbf{H}^{(j)})\Big) \tag{3.76}$$

$$\stackrel{\mathrm{c}}{=} \frac{1}{J}\sum_{j,f,n} \log p(x_{fn}|\mathbf{h}_n;\ \underline{\mathbf{w}}_f). \tag{3.77}$$

As it turns out, we have

$$\hat{Q}_{\mathrm{H}}(\mathbf{W}) \stackrel{\mathrm{c}}{=} \frac{1}{J}\sum_{j} D_{\mathrm{IS}}(|\mathbf{X}|^2|\mathbf{W}\mathbf{H}^{(j)}). \tag{3.78}$$

Noting the parallel with the standard IS-NMF problem, this can be tackled with an MM algorithm.

3.C.3 MCEM-C

The MC approximation of Q^{C} writes

$$\hat{Q}^{\mathrm{C}}(\mathbf{W}) = -\frac{1}{J}\sum_{j} \log p(\mathbf{C}^{(j)}; \mathbf{W}) \tag{3.79}$$

$$= -\frac{1}{J}\sum_{j,k,n} \log p(\mathbf{c}_{kn}; \mathbf{w}_k) \tag{3.80}$$

$$\overset{\mathrm{c}}{=} \frac{1}{J}\sum_{j,k,n} \left(\sum_{f} \log(w_{fk}) + (\alpha_k + F) \log\left(\beta_k + \sum_{f} \frac{|c_{fkn}^{(j)}|^2}{w_{fk}} \right) \right). \tag{3.81}$$

We resort to majorization-minimization for optimization. The difficult term is the second logarithm term, which can be easily majorized. Indeed, the logarithm is concave, therefore always below its tangents:

$$\log(a+x) \leq \frac{x}{a+\tilde{x}} - \underbrace{\frac{\tilde{x}}{a+\tilde{x}} + \log(a+\tilde{x})}_{\text{cst}}. \tag{3.82}$$

We can therefore majorize Eq. (3.81):

$$\hat{Q}^{\mathrm{C}}(\mathbf{W}) \leq \frac{1}{J}\sum_{j,k,n} \left(\sum_{f} \log(w_{fk}) + (\alpha_k + F) \frac{\sum_f \frac{|c_{fkn}^{(j)}|^2}{w_{fk}}}{\beta_k + \sum_f \frac{|c_{fkn}^{(j)}|^2}{\tilde{w}_{fk}}} \right) + \text{cst}. \tag{3.83}$$

Deriving w.r.t. w_{fk}, and setting the equation to 0 yields

$$\sum_{j,n} \left(\frac{1}{w_{fk}} - \frac{\alpha_k + F}{\beta_k + \sum_f \frac{|c_{fkn}^{(j)}|^2}{\tilde{w}_{fk}}} \frac{|c_{fkn}^{(j)}|^2}{w_{fk}^2} \right) = 0, \tag{3.84}$$

which can be solved as

$$w_{fk} = \frac{\alpha_k + F}{NJ} \sum_{j,n} \frac{|c_{fkn}^{(j)}|^2}{\beta_k + \sum_f \frac{|c_{fkn}^{(j)}|^2}{\tilde{w}_{fk}}}. \tag{3.85}$$

Chapter 4

Temporal Non-Negative Matrix Factorization

This is joint work with Olivier Gouvert, Cédric Févotte and Olivier Cappé. A journal article is in preparation.

Contents

4.1 Introduction **110**
4.2 Comparative study of Gamma Markov chains **111**
4.2.1 Direct chaining on the rate parameter 112
4.2.2 Hierarchical chaining with an auxiliary variable 113
4.2.3 Chaining on the shape parameter 115
4.2.4 BGAR(1) 116
4.3 The BGAR-NMF model **119**
4.4 Maximum marginal likelihood estimation **121**
4.4.1 Objective 121
4.4.2 Sequential Monte Carlo methods 121
4.4.3 M-step 123
4.4.4 Experimental work 123
4.5 MAP estimation **125**
4.5.1 Problem setting 125
4.5.2 Optimization 126
4.5.3 Experimental work 129
4.6 Discussion **132**

4.1 Introduction

In Chapters 2 and 3, we focused on two probabilistic NMF models whose activation coefficients $\mathbf{h}_n$ were assumed independent. This implies the independence of the samples $\mathbf{v}_n$ as well. Such an assumption is obviously limiting in settings where the samples are known to be correlated; in particular when they describe the evolution of a certain process over time. As such, we wonder how to add statistical correlation to probabilistic NMF models.

A relatively easy way to do so is to lift the independence assumption on the columns of $\mathbf{H}$, for example by considering a Markov structure on these columns

$$p(\mathbf{H}) = p(\mathbf{h}_1) \prod_{n \geq 2} p(\mathbf{h}_n | \mathbf{h}_{n-1}). \tag{4.1}$$

We will refer to such models as *dynamical* NMF models. Note that very recent works go beyond the Markovian assumption, i.e., assume dependency with multiple past time steps, and are labeled as "deep" (Gong and Huang, 2017; Guo et al., 2018).

The transition distribution $p(\mathbf{h}_n|\mathbf{h}_{n-1})$ may make use of a transition matrix $\mathbf{\Pi}$ of size $K \times K$ to capture relationships between the different components; this has notably been considered in in Févotte et al. (2013) and Schein et al. (2016). In this case, the distribution of h_{kn} depends on a linear combination of all the components at the previous time step

$$p(\mathbf{h}_n|\mathbf{h}_{n-1}) = \prod_k p(h_{kn}| \sum_l \pi_{kl} h_{l(n-1)}). \tag{4.2}$$

In this chapter, we will restrict ourselves to $\mathbf{\Pi} = \mathbf{I}_K$. Equivalently, this amounts to modeling the K rows of $\mathbf{H}$ as independent (and therefore to smoothing the rows independently). Eq. (4.2) thus reduces to

$$p(\mathbf{h}_n|\mathbf{h}_{n-1}) = \prod_k p(h_{kn}|h_{k(n-1)}). \tag{4.3}$$

We will refer to such a model as a *temporal* NMF model.

A first way of dealing with the temporal evolution of a non-negative variable is to map a real variable to $\mathbb{R}_+$. It is then commonly assumed that this variable evolves in Gaussian noise. This is for example exploited in the seminal work of Blei and Lafferty (2006) on the extension of latent Dirichlet allocation to allow for topic evolution[1]. A similar assumption is made in Charlin et al. (2015), which introduces dynamics in the context of a Poisson likelihood (factorizing the user-item-time tensor). Gaussian assumptions allow to use well-known computational techniques, such as Kalman filtering, but result in loss of interpretability.

We will here focus on naturally non-negative Markov chains. Various non-negative Markov chains have been proposed in the NMF literature (Cemgil and Dikmen, 2007; Févotte et al., 2009; Acharya et al., 2015). They are all built in relation with the Gamma (or inverse

[1] Note that this particular mapping is actually slightly more complex, as the K-dimensional real vector must be mapped to the $(K-1)$ simplex due to further constraints in the model.

Gamma) distribution. As a matter of fact, these models exhibit the same drawback: the chains all have a degenerate stationary distribution. This can lead to undesirable behaviors, such as the instability or the degeneracy of realizations of the chains. We emphasize that this is problematic from the probabilistic perspective only, since these prior distributions may still represent an appropriate regularization in a MAP setting.

The contributions of this chapter are 4-fold:

- We review the existing non-negative Markov chains of the NMF literature and discuss some of their limitations. In particular we show that these chains all have a degenerate stationary distribution;
- We present an overlooked non-negative Markov chain from the time series literature, the first-order autoregressive Beta-Gamma process, denoted as BGAR(1) (Lewis et al., 1989), whose stationary distribution is Gamma. To the best of our knowledge, this particular chain has never been considered to model temporal dependencies in matrix factorization problems;
- We propose a novel model based on this BGAR(1) process, which we coin BGAR-NMF. We derive a a Monte Carlo Expectation-Maximization (MCEM) algorithm for MMLE, as well as an MM-based algorithm for MAP estimation;
- We show that the proposed MMLE approach fails to produce satisfactory results on real datasets, unlike the MAP approach, further assessed on a prediction task.

The remainder of the chapter is organized as follows. Section 4.2 introduces and compares non-negative Markov chains from the literature. Section 4.3 presents the novel temporal NMF model. Section 4.4 describes MMLE and its associated experimental work while Section 4.5 focuses on MAP estimation. We conclude in Section 4.6.

4.2 Comparative study of Gamma Markov chains

This section reviews existing models of Gamma Markov chains, i.e., Markov chains which evolve in $\mathbb{R}_+$ in relation with the Gamma distribution. We have identified three different models in the NMF literature:

1. Chaining on the rate parameter of a Gamma distribution (Section 4.2.1);
2. Chaining with an auxiliary variable to ensure conjugacy (Section 4.2.2);
3. Chaining on the shape parameter of a Gamma distribution (Section 4.2.3).

As shall be discussed in these subsections, these three models are all built around the assumption $\mathbb{E}(h_n|h_{n-1}) \propto h_{n-1}$ (which roughly means that the chain should not drift too far away from its previous value), but lack a well-defined stationary distribution, which leads to the degeneracy of the realizations of the chains. A fourth model from the time series literature, called BGAR(1), is presented in Section 4.2.4. It is built to have a well-defined stationary distribution (it is marginally Gamma distributed), and does not share the assumption $\mathbb{E}(h_n|h_{n-1}) \propto h_{n-1}$. The realizations of the chain are not degenerate and

exhibit some interesting properties. To the best of our knowledge, this kind of process has never been used in a probabilistic NMF problem to model temporal evolution.

Throughout the section, $(h_n)_{n\geq 1}$ denotes the (scalar) Markov chain of interest, where the index k as in Eq. (4.3) has been dropped for enhanced readability. It is further assumed that h_1 is set to a fixed, deterministic value. All the formulas needed to derive the computations of mean and variance throughout the next subsections are given in Appendix 4.A.

4.2.1 Direct chaining on the rate parameter

4.2.1.1 Model

Let us consider a general Gamma Markov chain model with a chaining on the rate parameter:

$$h_n|h_{n-1} \sim \text{Gamma}\left(\alpha, \frac{\beta}{h_{n-1}}\right). \tag{4.4}$$

As it turns out, Eq. (4.4) can be rewritten as a multiplicative noise model:

$$h_n = h_{n-1} \times \phi_n, \tag{4.5}$$

where ϕ_n are i.i.d. Gamma(α, β) random variables.

We have

$$\mathbb{E}(h_n|h_{n-1}) = \frac{\alpha}{\beta} h_{n-1}, \quad \text{var}(h_n|h_{n-1}) = \frac{\alpha}{\beta^2} h_{n-1}^2. \tag{4.6}$$

This model was introduced in Févotte et al. (2009) to add smoothness to the activation coefficients in the context of audio signal processing. The parameters were set to $\alpha > 1$ and $\beta = \alpha - 1$, such that the mode would be located at $h_n = h_{n-1}$. A similar inverse Gamma Markov chain was also considered in Févotte et al. (2009) and in Févotte (2011).

4.2.1.2 Analysis

From Eq. (4.5) we can write:

$$h_n = h_1 \prod_{i=2}^{n} \phi_i. \tag{4.7}$$

The independence of the ϕ_i yields:

$$\mathbb{E}(h_n) = h_1 \left(\frac{\alpha}{\beta}\right)^{n-1}, \quad \text{var}(h_n) = h_1^2 \left[\left(\frac{\alpha^2}{\beta^2} + \frac{\alpha}{\beta^2}\right)^{n-1} - \left(\frac{\alpha^2}{\beta^2}\right)^{n-1}\right]. \tag{4.8}$$

We enumerate all the possible regimes, which all give rise to degenerate stationary distributions for different reasons:

- $\beta > \sqrt{\alpha(\alpha+1)}$: both mean and variance go to zero;

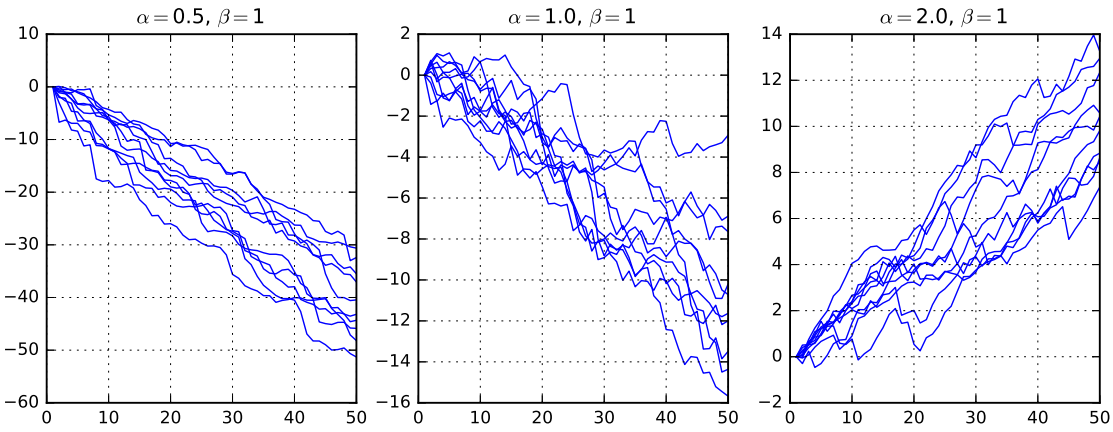


Figure 4.1: Realizations of the Markov chain defined in Eq. (4.4). The initial value h_1 is set to 1, and chains were simulated until $n = 50$. Each subplot contains ten independent realizations, with the value of the parameters (α, β) given at the top of the subplot. $\log_{10}(h_n)$ is displayed.

- $\beta = \sqrt{\alpha(\alpha+1)}$: variance converges to 1, however the mean goes to zero;
- $\beta \in \left]\alpha; \sqrt{\alpha(\alpha+1)}\right[$:variance goes to infinity, mean goes to zero;
- $\beta = \alpha$: mean is equal to 1, but the variance goes to infinity;
- $\beta < \alpha$: both mean and variance go to infinity.

Each subplot of Figure 4.1 displays in $\log_{10}$-scale ten independent realizations of the chain, for a different set of parameters (α, β). As we can see, the realizations of the chain either collapse to 0, or diverge.

4.2.2 Hierarchical chaining with an auxiliary variable

4.2.2.1 Model

Let us consider the following Gamma Markov chain model introduced in Cemgil and Dikmen (2007):

$$z_n | h_{n-1} \sim \text{Gamma}(\alpha_z, \beta_z h_{n-1}), \tag{4.9}$$
$$h_n | z_n \sim \text{Gamma}(\alpha_h, \beta_h z_n). \tag{4.10}$$

As it turns out, this model can also be rewritten as a multiplicative noise model:

$$h_n = h_{n-1} \times \tilde{\phi}_n, \tag{4.11}$$

where $\tilde{\phi}_n$ are i.i.d. random variables defined as the ratio of two independent Gamma random variables of parameters (α_h, β_h) and (α_z, β_z). The distribution of $\tilde{\phi}_n$ is actually known in

closed form, namely

$$\tilde{\phi}_n \sim \text{BetaPrime}\left(\alpha_h, \alpha_z, 1, \tilde{\beta}\right), \tag{4.12}$$

with $\tilde{\beta} = \frac{\beta_z}{\beta_h}$ (see Appendix 4.B for a definition).

We have

$$\mathbb{E}(h_n|h_{n-1}) = \tilde{\beta}\frac{\alpha_h}{\alpha_z - 1} \qquad \alpha_z > 1, \tag{4.13}$$

$$\text{var}(h_n|h_{n-1}) = \tilde{\beta}^2\frac{\alpha_h(\alpha_h + \alpha_z - 1)}{(\alpha_z - 1)^2(\alpha_z - 2)} \qquad \alpha_z > 2. \tag{4.14}$$

This model is less straightforward in its construction than the previous one, as it makes use of an auxiliary variable z_n (note that a similar inverse Gamma construction was proposed as well in Cemgil and Dikmen (2007)). There are two motivations behind the introduction of this auxiliary variable:

1. Firstly, it ensures what is referred to as "positive correlation" in Cemgil and Dikmen (2007), i.e., $\mathbb{E}(h_n|h_{n-1}) \propto h_{n-1}$ (something the model described by Eq. (4.4) does as well).
2. Secondly, it ensures the so-called conjugacy of the model, i.e., the conditional distributions $p(z_n|h_{n-1}, h_n)$ and $p(h_n|z_n, z_{n+1})$ remain Gamma distributions. Indeed, these are the distributions of interest when considering Gibbs sampling or variational inference. This property is not achieved by the model described by Eq.(4.4) (i.e., $p(h_n|h_{n-1}, h_{n+1})$ is neither Gamma, nor a known distribution).

This particular chain has been used in the context of audio signal processing in Virtanen et al. (2008) (under the assumption of a Poisson likelihood, which does not fit the nature of the data), and also to model the evolution of user and item preferences in the context of recommender systems (Jerfel et al., 2017; Do and Cao, 2018).

4.2.2.2 Analysis

From Eq. (4.11), we can write:

$$h_n = h_1 \prod_{i=2}^{n} \tilde{\phi}_i. \tag{4.15}$$

We have by independence of the $\tilde{\phi}_i$:

$$\mathbb{E}(h_n) = h_1\left(\tilde{\beta}\frac{\alpha_h}{\alpha_z - 1}\right)^{n-1} \qquad \alpha_z > 1, \tag{4.16}$$

$$\text{var}(h_n) = h_1^2\tilde{\beta}^{2(n-1)}\left[\left(\frac{\alpha_h^2}{(\alpha_z - 1)^2} + \frac{\alpha_h(\alpha_h + \alpha_z - 1)}{(\alpha_z - 1)^2(\alpha_z - 2)}\right)^{n-1} - \left(\frac{\alpha_h^2}{(\alpha_z - 1)^2}\right)^{n-1}\right] \qquad \alpha_z > 2. \tag{4.17}$$

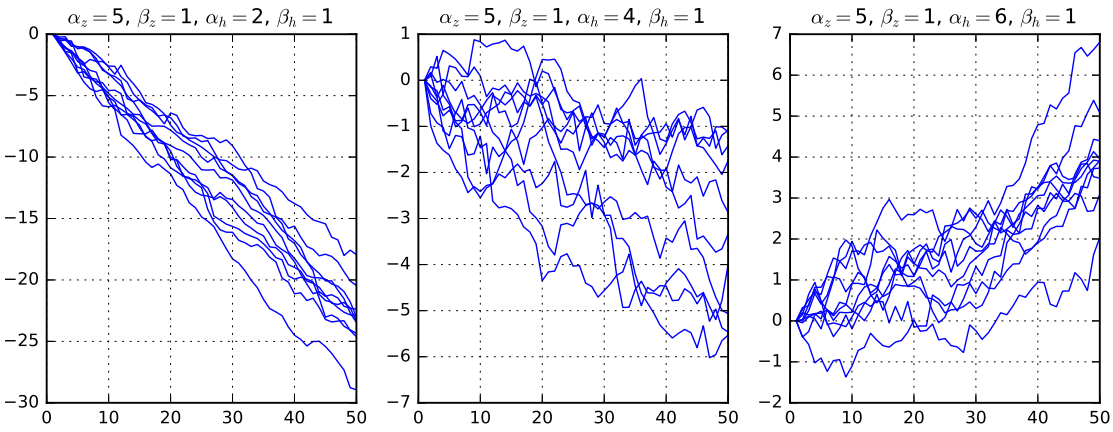


Figure 4.2: Realizations of the Markov chain defined in Eq. (4.9)-(4.10). The initial value h_1 is set to 1, and chains were simulated until $n = 50$. Each subplot contains ten independent realizations, with the value of the parameters $(\alpha_z, \beta_z, \alpha_h, \beta_h)$ given at the top of the subplot. $\log_{10}(h_n)$ is displayed.

The conclusions are the same as with the previous model. Each subplot of Figure 4.2 displays in $\log_{10}$-scale ten independent realizations of the chain, for a different set of parameters $(\alpha_z, \beta_z, \alpha_h, \beta_h)$. As we can see, the realizations of the chain either collapse to 0, or diverge.

4.2.3 Chaining on the shape parameter

4.2.3.1 Model

Let us consider a general Gamma Markov chain model with a chaining on the shape parameter:

$$h_n | h_{n-1} \sim \text{Gamma}(\alpha h_{n-1}, \beta). \tag{4.18}$$

We have

$$\mathbb{E}(h_n | h_{n-1}) = \frac{\alpha}{\beta} h_{n-1}, \quad \text{var}(h_n | h_{n-1}) = \frac{\alpha}{\beta^2} h_{n-1}. \tag{4.19}$$

In contrast with the two models presented in Section 2.1, this model cannot be rewritten as a multiplicative noise model, or any noise model. Therefore this model is more intricate to interpret. It was introduced in Acharya et al. (2015) in the context of Poisson factorization. It is mainly motivated by a computational trick that can be used when working with a Poisson likelihood, hence making a Gibbs sampling feasible in the model. The authors set the value of α to 1 (though the same trick can be applied for any value of α).

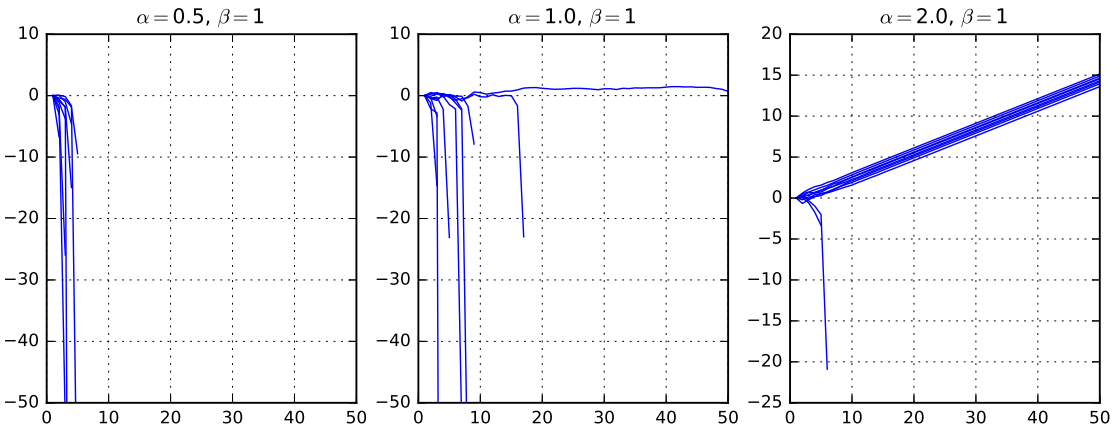


Figure 4.3: Realizations of the Markov chain defined in Eq. (4.18). The initial value h_1 is set to 1, and chains were simulated until $n = 50$. Each subplot contains ten independent realizations, with the value of the parameters (α, β) given at the top of the subplot. $\log_{10}(h_n)$ is displayed.

4.2.3.2 Analysis

Using the law of total expectation and total variance, it can be shown that

$$\mathbb{E}(h_n) = \left(\frac{\alpha}{\beta}\right)^{n-1} h_1, \quad \text{var}(h_n) = \frac{1}{\beta}\left(\frac{\alpha}{\beta}\right)^{n-1} h_1 \sum_{i=0}^{n-2} \left(\frac{\alpha}{\beta}\right)^{i}. \tag{4.20}$$

The discussion is hence driven by the value of $r = \frac{\alpha}{\beta}$.

- If $r < 1$, mean and variance go to zero (Dirac distribution located at 0)
- If $r = 1$, mean is fixed but variance goes to infinity (linearly)
- If $r > 1$, mean and variance go to infinity

This chain only exhibits degenerate stationary distributions. Each subplot of Figure 4.3 displays in $\log_{10}$-scale ten independent realizations of the chain, for a different set of parameters (α, β). As we can see, the realizations of the chain either collapse to 0, or diverge.

4.2.4 BGAR(1)

We now discuss the first order autoregressive Beta-Gamma process of Lewis et al. (1989), a stochastic process which is marginally Gamma distributed. The authors referred to the process as "BGAR(1)". However, to the best of our knowledge, no extension to higher-order autoregressive processes exists in the time series literature. As such, from now on, we will simply refer to it as "BGAR".

4.2.4.1 Model

Consider $\alpha > 0$, $\beta > 0$, $\rho \in [0, 1[$. The BGAR process is defined as:

$$h_1 \sim \text{Gamma}(\alpha, \beta), \tag{4.21}$$
$$h_n = b_n h_{n-1} + \epsilon_n \qquad n \geq 2, \tag{4.22}$$

where $b_n \in [0, 1]$ and $\epsilon_n > 0$ are i.i.d. random variables distributed as:

$$b_n \sim \text{Beta}(\alpha\rho, \alpha(1-\rho)), \tag{4.23}$$
$$\epsilon_n \sim \text{Gamma}(\alpha(1-\rho), \beta). \tag{4.24}$$

$(h_n)_{n\geq 1}$ is called the BGAR process. It is parametrized by α, β and ρ. We have

$$\mathbb{E}(h_n|h_{n-1}) = \rho h_{n-1} + \frac{\alpha(1-\rho)}{\beta}, \tag{4.25}$$
$$\text{var}(h_n|h_{n-1}) = \frac{\rho(1-\rho)}{\alpha+1} h_{n-1}^2 + \frac{\alpha(1-\rho)}{\beta^2}. \tag{4.26}$$

As we can see, BGAR(1) already differs from the three previously presented models because the conditional expectation $\mathbb{E}(h_n|h_{n-1})$ is not proportional to h_{n-1} (it is an affine transformation).

We emphasize that the distribution $p(h_n|h_{n-1})$ is not known in closed form. Only $p(h_n|b_n, h_{n-1})$ is known; it is a shifted Gamma distribution. The generative model may therefore be rewritten as

$$h_1 \sim \text{Gamma}(\alpha, \beta), \tag{4.27}$$
$$b_n \sim \text{Beta}(\alpha\rho, \alpha(1-\rho)) \qquad n \geq 2, \tag{4.28}$$
$$h_n|b_n, h_{n-1} \sim \text{Gamma}(\alpha(1-\rho), \beta, \text{loc} = b_n h_{n-1}) \qquad n \geq 2. \tag{4.29}$$

where the distribution in Eq. (4.29) is a shifted Gamma distribution with a location parameter "loc".

4.2.4.2 Analysis

To study the marginal distribution of the process, we recall the following lemma.

Lemma 4.1. If $X \sim \text{Beta}(a, b)$ and $Y \sim \text{Gamma}(a+b, c)$ are independent random variables, then $Z = XY$ is $\text{Gamma}(a, c)$ distributed.

Proposition 4.1. h_n is marginally $\text{Gamma}(\alpha, \beta)$ distributed.

Proof. Follows by induction. Consider n such that h_n is $\text{Gamma}(\alpha, \beta)$ distributed. Then, $\epsilon_{n+1} h_n$ is $\text{Gamma}(\alpha\rho, \beta)$ distributed (Lemma 4.1). Finally, $h_{n+1} = \epsilon_{n+1} h_n + b_{n+1}$ is $\text{Gamma}(\alpha, \beta)$ distributed (sum of independent Gamma random variables), which concludes the proof. □

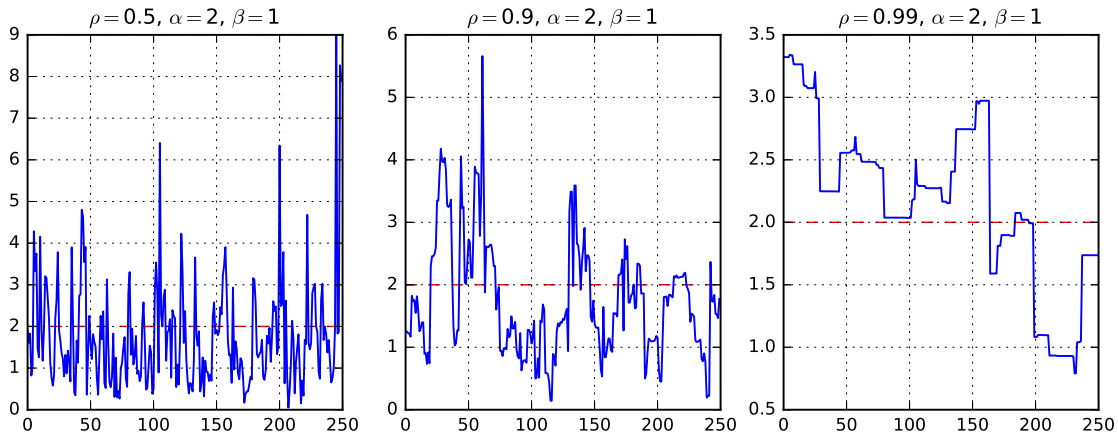


Figure 4.4: Three realizations of the BGAR(1) process, with parameters fixed to $\alpha = 2$ and $\beta = 1$, and a different parameter ρ in each subplot. The mean of the process is displayed by a dashed red line.

Therefore the parameters α and β control the marginal distribution. The parameter ρ controls the correlation between successive values, as is discussed in the following proposition.

Proposition 4.2. $\mathrm{corr}(h_n, h_{n+r}) = \rho^r$

Proof. Given in Appendix 4.C for the case between two successive values. □

Two limit cases can be exhibited:

- When $\rho = 0$, the h_n are i.i.d. random variables;
- When $\rho \to 1$, the process is not random anymore, and $h_n = h_1$ for all n (note that $\rho = 1$ is not an admissible value).

Note that BGAR is not the only Markovian process with a marginal Gamma distribution considered in the literature. We mention the GAR(1) process (first-order autoregressive Gamma process) of Gaver and Lewis (1980), which is also marginally Gamma distributed. However, this particular process is piecewise deterministic, and its parameters are "coupled": the parameters of the marginal distribution also have an influence on other properties of the model. As such, it is not well-suited to our problem, and will not be considered here.

Realizations of the process. Figure 4.4 displays three realizations of the BGAR process, with parameters fixed to $\alpha = 2$ and $\beta = 1$, and a different parameter ρ in each subplot. When $\rho = 0.5$, the correlation is weak, and no particular structure is observed. However, as ρ goes to 1, the correlation becomes stronger, and we typically observe a "floor" phenomenon.

Moreover, we have

$$\left(\mathbb{E}(h_n|h_{n-1}) > h_{n-1}\right) \Leftrightarrow \left(h_{n-1} < \frac{\alpha}{\beta}\right) \tag{4.30}$$

If h_{n-1} is below the mean of the marginal distribution ($\frac{\alpha}{\beta}$), then h_n will be in expectation above h_{n-1}, and vice-versa. However, note that, as $\rho \to 1$, this phenomenon gets weaker, as the variance of h_n goes to zero.

4.3 The BGAR-NMF model

In this section, we introduce a novel temporal NMF model which makes use of the BGAR process presented in Section 4.2.4. We will use the slightly abusive notation $\underline{\mathbf{h}} \sim \text{BGAR}(\rho, \alpha, \beta)$ to denote that the entries of the row vector $\underline{\mathbf{h}}$ are a realization of the BGAR process with parameters α, β, and ρ.

We consider the following temporal NMF model

$$\underline{\mathbf{h}}_k \sim \text{BGAR}(\rho_k, \alpha_k, \beta_k) \tag{4.31}$$
$$v_{fn}|\mathbf{h}_n \sim \text{Poisson}([\mathbf{WH}]_{fn}). \tag{4.32}$$

That is to say that each row of $\mathbf{H}$ is independent and follows a BGAR(1) process of different parameters. $\mathbf{W}$ is left to be a deterministic variable. We choose to work without loss of generality with a Poisson observation model. Indeed, as discussed in Section 1.3.2, this model can be generalized to non-integer data by considering the compound Poisson distribution. A graphical representation of the model is given in Figure 4.5.

As explained preivously, we may rewrite the model with the auxiliary variables b_{kn} (Eqs. (4.28)-(4.29)), leading to

$$h_{k1} \sim \text{Gamma}(\alpha_k, \beta_k) \tag{4.33}$$
$$b_{kn} \sim \text{Beta}(\alpha_k\rho_k, \alpha_k(1-\rho_k)) \qquad n \geq 2 \tag{4.34}$$
$$h_{kn}|b_{kn}, h_{k(n-1)} \sim \text{Gamma}(\alpha_k(1-\rho_k), \beta_k, \text{loc} = b_{kn}h_{k(n-1)}) \qquad n \geq 2 \tag{4.35}$$
$$v_{fn}|\mathbf{h}_n \sim \text{Poisson}([\mathbf{WH}]_{fn}). \tag{4.36}$$

A graphical representation of the augmented model is given in Figure 4.6.

In our setting, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]^{\mathrm{T}}$, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_K]^{\mathrm{T}}$ and $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_K]^{\mathrm{T}}$ are treated as fixed hyperparameters. $\mathbf{W}$ is also treated as a deterministic variable to be estimated.

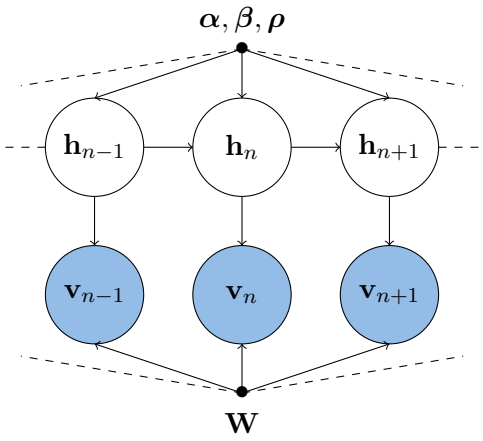


Figure 4.5: The BGAR-NMF model. Observed variables are in blue, while latent variables are in white. In our setting, $\mathbf{h}_n$ is of size K, and $\mathbf{v}_n$ is of size F.

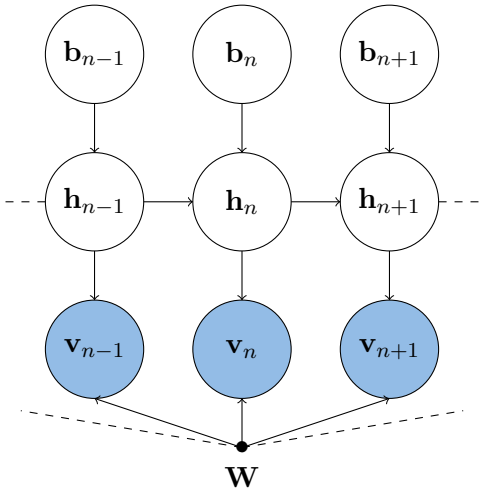


Figure 4.6: The augmented BGAR-NMF model. Observed variables are in blue, while latent variables are in white. In our setting, $\mathbf{b}_n$ is of size K, $\mathbf{h}_n$ is of size K, and $\mathbf{v}_n$ is of size F. The hyperparameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}$ have been omitted for enhanced readability.

4.4 Maximum marginal likelihood estimation

4.4.1 Objective

In this section, we aim at estimating $\mathbf{W}$ by maximizing the marginal likelihood:

$$\max_{\mathbf{W}} p(\mathbf{V};\mathbf{W}) = \int_{\mathbf{H},\mathbf{B}} p(\mathbf{V},\mathbf{H},\mathbf{B};\mathbf{W})\mathrm{d}\mathbf{H}\mathrm{d}\mathbf{B}, \tag{4.37}$$

where the joint likelihood of the model is given by

$$p(\mathbf{V},\mathbf{H},\mathbf{B};\mathbf{W}) = \prod_n p(\mathbf{v}_n|\mathbf{h}_n;\mathbf{W}) \prod_k \left(p(h_{k1}) \prod_{n\geq 2} \Big(p(h_{kn}|b_{kn}, h_{k(n-1)}) p(b_{kn}) \Big) \right). \tag{4.38}$$

We once again turn to an EM algorithm to solve this optimization task. Given the current value of the parameter $\tilde{\mathbf{W}}$, we iteratively optimize w.r.t. $\mathbf{W}$ the following functional

$$Q(\mathbf{W};\tilde{\mathbf{W}}) = \int_{\mathbf{H},\mathbf{B}} \log p(\mathbf{V},\mathbf{H},\mathbf{B};\mathbf{W}) p(\mathbf{H},\mathbf{B}|\mathbf{V};\tilde{\mathbf{W}})\mathrm{d}\mathbf{H}\mathrm{d}\mathbf{B}. \tag{4.39}$$

In our setting, the posterior of the latent variables $p(\mathbf{H},\mathbf{B}|\mathbf{V};\tilde{\mathbf{W}})$ is intractable. As such, we resort to MCEM, in which the functional of Eq. (4.39) is replaced by its Monte Carlo approximation. We therefore need a way to draw samples from $p(\mathbf{H},\mathbf{B}|\mathbf{V};\tilde{\mathbf{W}})$.

The observations $\mathbf{V}$ and the set $\{\mathbf{H},\mathbf{B}\}$ define a so-called hidden Markov model (HMM), sometimes referred to as general state-space models. In particular, $\{\mathbf{H},\mathbf{B}\}$ are called the state variables[2], and $\mathbf{W}$ is a static parameter of the HMM. Inference in HMM is a well-studied topic (Cappé et al., 2005), and the estimation of these so-called static parameters is a specific problem, see for example the survey of Kantas et al. (2015). In particular, sequential Monte Carlo (SMC) methods are central to such inference problems, and we describe in the following subsection the key elements of the methodology.

4.4.2 Sequential Monte Carlo methods

We denote by $\mathbf{x}_n = \{\mathbf{h}_n, \mathbf{b}_n\}$ the state variable at time step n. We recall that the associated observation is the variable $\mathbf{v}_n$. In this subsection, we will use the standard SMC notations $\mathbf{x}_{1:n} = \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ and $\mathbf{v}_{1:n} = \{\mathbf{v}_1,\ldots,\mathbf{v}_n\}$ to denote collections of state or observation variables from time 1 through n. With these notations, our goal is to sample from $p(\mathbf{x}_{1:N}|\mathbf{v}_{1:N})$, which is referred to as the smoothing distribution. The methodology boils down to two steps.

[2]Of course, taking $\mathbf{H}$ to be the state variables is the most straightforward choice, however, since the transition distribution $p(\mathbf{h}_n|\mathbf{h}_{n-1})$ is not known, we use the augmented version of our model with the variables $\mathbf{B}$.

4.4.2.1 Particle filtering

Particle filtering, a specific instance of sequential Monte Carlo methods, sequentially yields a so-called particle approximation of the filtering distributions, $\{p(\mathbf{x}_n|\mathbf{v}_{1:n})\}_n$. That is to say that these distributions are approximated with a finite number (denoted N_p) of weighted particles:

$$p(\mathbf{x}_n|\mathbf{v}_{1:n}) \simeq \sum_{i=1}^{N_p} \xi_n^{(i)} \delta_{\mathbf{x}_n^{(i)}}(\mathbf{x}_n). \tag{4.40}$$

$\xi_n^{(i)}$ is the weight associated to the particle $\mathbf{x}_n^{(i)}$, and are such that $\xi_n^{(i)} > 0$ and $\sum_i \xi_n^{(i)} = 1$. δ is the Dirac function. These approximations are computed sequentially using efficient importance sampling and resampling techniques. The interested reader is referred to the following tutorials for details: Cappé et al. (2007); Doucet and Johansen (2011).

We use the standard bootstrap particle filtering (i.e., when the importance distribution is chosen to be the transition distribution of the HMM) for this task. The associated algorithm is described in Appendix 4.D.

4.4.2.2 Particle smoothing

Once particle filtering has been carried out, we may obtain realizations from the smoothing distribution via the following backward recursion:

$$p(\mathbf{x}_{1:N}|\mathbf{v}_{1:N}) = p(\mathbf{x}_N|\mathbf{v}_{1:N}) \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{v}_{1:n}). \tag{4.41}$$

Indeed, a particle approximation of the filtering distribution yields a particle approximation of $p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{v}_{1:n})$, since

$$p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{v}_{1:n}) \propto p(\mathbf{x}_{n+1}|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{v}_{1:n}), \tag{4.42}$$

where $p(\mathbf{x}_{n+1}|\mathbf{x}_n)$ is simply the transition distribution of our HMM. As such, we obtain

$$p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{v}_{1:n}) \simeq \sum_{i=1}^{N_p} \xi_{n|n+1}^{(i)} \delta_{\mathbf{x}_n^{(i)}}(\mathbf{x}_n), \tag{4.43}$$

where the weights are such that

$$\xi_{n|n+1}^{(i)} = \frac{p(\mathbf{x}_{n+1}|\mathbf{x}_n^{(i)}) \xi_n^{(i)}}{\sum_j p(\mathbf{x}_{n+1}|\mathbf{x}_n^{(j)}) \xi_n^{(j)}}. \tag{4.44}$$

This methodology is called forward filtering–backward simulation (FFBSi), and was introduced in Godsill et al. (2004). The states are generated sequentially in reverse time. See Algorithm 5 for details.

Algorithm 5: Sample a single realization from the smoothing distribution

```
# Initialization (n = N)
Choose x̃_N = x_N^(i) with probability ξ_N^(i)
for n = N − 1, ..., 1 do
    # Recompute weights
    Compute ξ_{n|n+1}^(i) as in Eq. (4.44)
    # Back-Propagation
    Choose x̃_n = x_n^(i) with probability ξ_{n|n+1}^(i)
end
```

We emphasize that Algorithm 5 describes how to sample *one* realization from the smoothing distribution. Another additional for loop has to be placed on top of this algorithm to sample multiple realizations. However, this task can be trivially parallelized.

4.4.3 M-step

Once having sampled N_T trajectories from the smoothing distribution, the MC approximation of the functional in Eq. (4.39) reduces to

$$\hat{Q}(\mathbf{W}) = \sum_{j=1}^{N_T} \log p(\mathbf{V}, \mathbf{H}^{(j)}, \mathbf{B}^{(j)}; \mathbf{W}) \tag{4.45}$$

$$\stackrel{c}{=} \sum_{j=1}^{N_T} \log p(\mathbf{V}|\mathbf{H}^{(j)}; \mathbf{W}). \tag{4.46}$$

The observation distribution is Poisson, as such the optimization of this function has already been addressed in Section 2.5. We recall the update rule

$$w_{fk} = \tilde{w}_{fk} \frac{\sum_{j,n} h_{kn}^{(j)} v_{fn} [\mathbf{W}\mathbf{H}^{(j)}]_{fn}^{-1}}{\sum_{j,n} h_{kn}^{(j)}}. \tag{4.47}$$

4.4.4 Experimental work

We illustrate the behavior the proposed MCEM algorithm on two datasets: a small-dimensioned synthetic dataset, and a real dataset.

On synthetic data. We generate a dataset of size 10×100 according to the BGAR-NMF model, with a random dictionary of mean 1, $\rho_k^\star = 0.95$, $\alpha_k^\star = 1$, $\beta_k^\star = 1$. This dataset is denoted by $\mathbf{V}_s$. We then apply our MCEM algorithm with $K = 5$, $\boldsymbol{\alpha} = \boldsymbol{\alpha}^\star$, $\boldsymbol{\beta} = \boldsymbol{\beta}^\star$, $\boldsymbol{\rho} = \boldsymbol{\rho}^\star$,

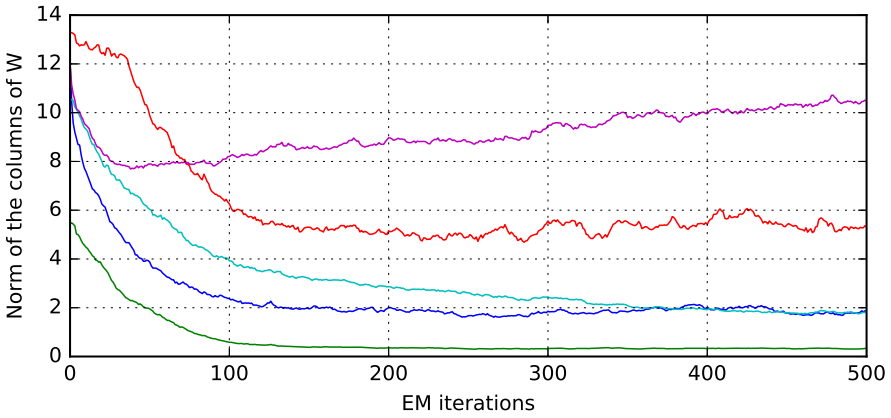


Figure 4.7: Evolution of the norm of the $K = 5$ columns $\mathbf{w}_k$ of the dictionary w.r.t. the number of EM iterations on synthetic dataset $\mathbf{V}_s$.

$N_p = 10000$, $N_T = 1000$ for a total of 500 EM iterations. Figure 4.7 displays the norm of the columns of the iterates w.r.t. the number of iterations.

On a real dataset. We consider the `NIPS` dataset[3], which contains word counts (with stop words removed) of all the articles published at the NIPS[4] conference between 1987 and 2015. We regrouped the articles per year, yielding an observation matrix of size 11463×29. We apply our MCEM algorithm with $K = 10$, $N_p = 10000$, $N_T = 1000$ for a total of 3000 EM iterations. Figure 4.8 displays the norm of the columns of the iterates w.r.t. the number of iterations.

Discussion of the results. Figures 4.7 and 4.8 are typical examples of the output of the proposed MCEM algorithm. If the algorithm seemingly converges on small-dimensioned problems, it fails to produce satisfactory results on larger-dimensioned real world datasets. On the displayed `NIPS` dataset example, there is too much variance in the iterates, making it impossible to obtain an exploitable point estimate. Increasing the values of N_p or N_T did not lead to significant improvements in our case. The most likely explanation is that the obtained samples do not correctly describe the posterior of the latent variables. Moreover, the method seems prone to label switching. Designing a finer SMC-based algorithm remains an open problem at this stage. The following section presents an alternative inference paradigm for the BGAR-NMF model.

[3] https://archive.ics.uci.edu/ml/datasets/NIPS+Conference+Papers+1987-2015.
[4] Now called NeurIPS.

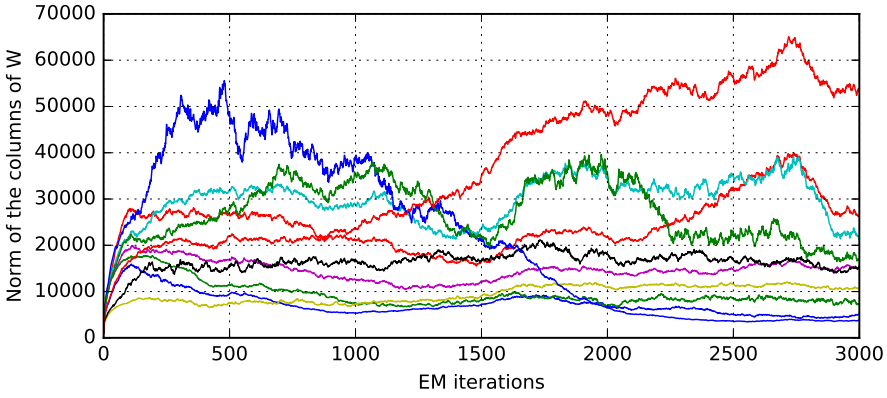


Figure 4.8: Evolution of the norm of the $K = 10$ columns $\mathbf{w}_k$ of the dictionary w.r.t. the number of EM iterations on the `NIPS` dataset.

4.5 MAP estimation

In this section, we develop an alternative procedure to MMLE, namely a MAP estimation in the BGAR-NMF model. From a statistical point of view, a MAP estimation is less well-posed than MMLE. Nevertheless, this computation-friendlier procedure will enable us to work on real datasets.

4.5.1 Problem setting

Objective function

MAP estimation amounts to the minimization of the following function

$$C(\mathbf{W}, \mathbf{H}, \mathbf{B}) = -\log p(\mathbf{H}, \mathbf{B}|\mathbf{V}; \mathbf{W}) \tag{4.48}$$

$$\stackrel{c}{=} -\log p(\mathbf{V}|\mathbf{H}; \mathbf{W}) - \log p(\mathbf{H}, \mathbf{B}) \tag{4.49}$$

$$\stackrel{c}{=} -\sum_{f,n} \log p(v_{fn}|[\mathbf{WH}]_{fn}) - \sum_{k} \left(\log p(h_{k1}) + \sum_{n \geq 2} \Big(\log p(h_{kn}|b_{kn}, h_{k(n-1)}) + \log p(b_{kn}) \Big) \right). \tag{4.50}$$

This expression consists in a data-fitting term, and a regularization term aiming at smoothing the rows of $\mathbf{H}$. Let us now detail each term. With the notations $\gamma_k = \alpha_k(1 - \rho_k)$ and

$\eta_k = \alpha_k \rho_k$, we can write

$$-\log p(v_{fn}|[\mathbf{WH}]_{fn}) \stackrel{c}{=} -v_{fn}\log([\mathbf{WH}]_{fn}) + [\mathbf{WH}]_{fn}, \tag{4.51}$$

$$-\log p(h_{k1}) \stackrel{c}{=} (1-\alpha_k)\log(h_{k1}) + \beta_k h_{k1}, \tag{4.52}$$

$$-\log p(h_{kn}|b_{kn}, h_{k(n-1)}) \stackrel{c}{=} (1-\gamma_k)\log(h_{kn} - b_{kn}h_{k(n-1)}) + \beta_k(h_{kn} - b_{kn}h_{k(n-1)}), \tag{4.53}$$

$$-\log p(b_{kn}) \stackrel{c}{=} (1-\eta_k)\log(b_{kn}) + (1-\gamma_k)\log(1-b_{kn}). \tag{4.54}$$

Constraints

By construction, the variables h_{kn} and b_{kn} must lie in a specific interval given the values of all the other variables. Indeed, as $h_{kn} = b_{kn}h_{k(n-1)} + \epsilon_{kn}$ (Eq. (4.22)), where ϵ_{kn} is a non-negative random variable, we have $h_{kn} \geq b_{kn}h_{k(n-1)}$, and $b_{kn} \leq \frac{h_{kn}}{h_{k(n-1)}}$.

Similarly, as $h_{k(n+1)} = b_{k(n+1)}h_{kn} + \epsilon_{k(n+1)}$, we have $h_{kn} \leq \frac{h_{k(n+1)}}{b_{k(n+1)}}$. As such, we obtain the following constraints:

$$0 \leq h_{k1} \leq \frac{h_{k2}}{b_{k2}}, \tag{4.55}$$

$$b_{kn}h_{k(n-1)} \leq h_{kn} \leq \frac{h_{k(n+1)}}{b_{k(n+1)}} \qquad 2 \leq n < N, \tag{4.56}$$

$$b_{kN}h_{k(N-1)} \leq h_{kN}, \tag{4.57}$$

and

$$0 \leq b_{kn} \leq \min\left(1, \frac{h_{kn}}{h_{k(n-1)}}\right). \tag{4.58}$$

We therefore introduce the notations

$$c_{kn} = b_{kn}h_{k(n-1)}, \quad d_{kn} = \frac{h_{k(n+1)}}{b_{k(n+1)}}, \quad z_{kn} = \frac{h_{kn}}{h_{k(n-1)}}, \tag{4.59}$$

as these quantities arise naturally in our derivations.

4.5.2 Optimization

To optimize the function C, we resort to an MM-based scheme. The only term we are going to majorize is $-\log p(\mathbf{V}|\mathbf{H};\mathbf{W})$. This is already well-known from the literature (Lee and Seung, 2000; Févotte and Idier, 2011). The function

$$G_1(\mathbf{H};\tilde{\mathbf{H}}) = -\sum_{k,n} p_{kn}\log(h_{kn}) + \sum_{k,n} q_k h_{kn}, \tag{4.60}$$

with the notations

$$p_{kn} = \tilde{h}_{kn}\sum_f w_{fk}\frac{v_{fn}}{[\mathbf{W}\tilde{\mathbf{H}}]_{fn}}, \quad q_k = \sum_f w_{fk}, \tag{4.61}$$

is a tight auxiliary function of $-\log p(\mathbf{V}|\mathbf{H};\mathbf{W})$ at $\mathbf{H}=\tilde{\mathbf{H}}$. Similarly the function

$$G_2(\mathbf{W};\tilde{\mathbf{W}}) = -\sum_{f,k} p'_{fk}\log(w_{fk}) + \sum_{f,k} q'_k w_{fk}, \tag{4.62}$$

with the notations

$$p'_{kn} = \tilde{w}_{fk}\sum_n h_{kn}\frac{v_{fn}}{[\tilde{\mathbf{W}}\mathbf{H}]_{fn}}, \quad q'_k = \sum_n h_{kn}, \tag{4.63}$$

is a tight auxiliary function of $-\log p(\mathbf{V}|\mathbf{H};\mathbf{W})$ at $\mathbf{W}=\tilde{\mathbf{W}}$.

As such, regardless of the variable to be optimized, we end up with an auxiliary function with logarithmic terms or linear terms. Therefore, the optimization of this variable will boil down to solving a polynomial equation.

Minmization w.r.t. W

Minimizing C w.r.t. to $\mathbf{W}$ amounts to minimizing G_2. The minimization w.r.t. w_{fk} can be done in closed-form thanks to the following update rule:

$$w_{fk} = \tilde{w}_{fk}\frac{\sum_n h_{kn}v_{fn}[\tilde{\mathbf{W}}\mathbf{H}]^{-1}_{fn}}{\sum_n h_{kn}}. \tag{4.64}$$

Minimization w.r.t. H

Minimizing C w.r.t. $\mathbf{H}$ amounts to minimizing $G_1(\mathbf{H};\tilde{\mathbf{H}}) - \log p(\mathbf{H},\mathbf{B})$. Consider its minimization w.r.t. a certain h_{kn}. The logarithmic terms may give rise to degenerate or intractable problems. As such, we have to control the limit values of the auxiliary function. By choosing $(1-\gamma_k) > 0$, we ensure that this limit is $+\infty$ (this choice will be discussed at the end of the subsection). The function to be minimized being continuous, this ensures the existence of a minimizer.

We must break down three different cases. In all sub-cases, it amounts to solving a polynomial equation. We know that at least one polynomial root must lie in the definition interval. If this is the case for several roots, we simply choose the root which gives the lowest objective value.

- For h_{k1}, the minimization of the auxiliary function amounts to solving the following order 2 polynomial equation over the interval $[0, d_{k1}]$

  $$a_{2,k1}h_{k1}^2 + a_{1,k1}h_{k1} + a_{0,k1} = 0, \tag{4.65}$$

  where

  $$a_{2,k1} = -(q_k + \beta_k(1-b_{k2})), \tag{4.66}$$
  $$a_{1,k1} = -(1-\alpha_k - p_{k1}) + (q_k + \beta_k(1-b_{k2}))d_{k1} - (1-\gamma_k), \tag{4.67}$$
  $$a_{0,k1} = (1-\alpha_k - p_{k1})d_{k1}. \tag{4.68}$$

  The roots can be found in closed form.

- For h_{kn} with $2 \leq n < N$, the minimization of the auxiliary function amounts to solving the following order 3 polynomial over the interval $[c_{kn}, d_{kn}]$

$$a_{3,kn}h_{kn}^3 + a_{2,kn}h_{kn}^2 + a_{1,kn}h_{kn} + a_{0,kn} = 0, \tag{4.69}$$

where

$$a_{3,kn} = -(q_k + \beta_k(1 - b_{k(n+1)})), \tag{4.70}$$
$$a_{2,kn} = p_{kn} - 2(1 - \gamma_k) + (q_k + \beta_k(1 - b_{k(n+1)})\,(c_{kn} + d_{kn})\,, \tag{4.71}$$
$$a_{1,kn} = -p_{kn}\,(c_{kn} + d_{kn}) + (1 - \gamma_k)\,(c_{kn} + d_{kn}) - (q_k + \beta_k(1 - b_{k(n+1)}))c_{kn}d_{kn}, \tag{4.72}$$
$$a_{0,kn} = p_{kn}c_{kn}d_{kn}. \tag{4.73}$$

The roots can be found numerically.

- For h_{kN}, the minimization of the auxiliary function amounts to solving the following order 2 polynomial over the interval $[c_{kn}, +\infty[$

$$a_{2,kN}h_{kN}^2 + a_{1,kN}h_{kN} + a_{0,kN} = 0, \tag{4.74}$$

where

$$a_{2,kN} = -(q_k + \beta_k), \tag{4.75}$$
$$a_{1,kN} = -p_{kN} - c_{kN}(q_k + \beta_k) + (1 - \gamma_k), \tag{4.76}$$
$$a_{0,kN} = c_{kN}p_{kN}. \tag{4.77}$$

The roots can be found in closed form.

Minimization w.r.t. B

Minimizing C w.r.t. $\mathbf{H}$ amounts to minimizing $-\log p(\mathbf{H}, \mathbf{B})$ only. Consider its minimization w.r.t. a specific b_{kn}. Similarly, the logarithmic terms may give rise to degenerate solutions. The choices of parameters $(1 - \gamma_k) > 0$ and $(1 - \eta_k) > 0$ ensures that the limits of the auxiliary function are $+\infty$. Then, the same considerations apply.

The minimization of the auxiliary function w.r.t. b_{kn} amounts to solving the following order 3 polynomial over the interval $[0, \min(1, z_{kn})]$

$$a_{3,kn}b_{kn}^3 + a_{2,kn}b_{kn}^2 + a_{1,kn}b_{kn} + a_{0,kn}d_{kn}, \tag{4.78}$$

where

$$a_{3,kn} = -b_k h_{k(n-1)} \tag{4.79}$$
$$a_{2,kn} = 2(1 - \gamma_k) + (1 - \eta_k) + \beta_k h_{k(n-1)}(z_{kn} + 1) \tag{4.80}$$
$$a_{1,kn} = -(1 - \gamma_k)(z_{kn} + 1) - (1 - \eta_k)(z_{kn} + 1) - \beta_k h_{k(n-1)}z_{kn} \tag{4.81}$$
$$a_{0,kn} = (1 - \eta_k)z_{kn}. \tag{4.82}$$

Again, the roots can be found numerically.

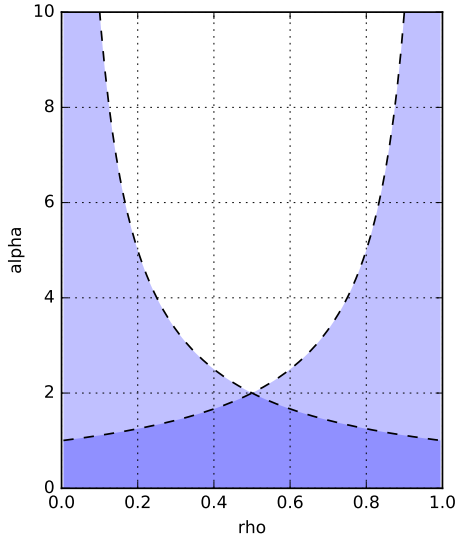


Figure 4.9: Hyperparameter values (in white) of the parameters α_k and ρ_k ensuring a well-posed MAP estimation in the BGAR-NMF model.

Admissible values of hyperparameters

To recap the discussion on admissible values of hyperparameters, to ensure the existence of minimizers to the auxiliary function, we have restricted ourselves to

$$\begin{cases} \alpha_k(1-\rho_k) > 1, \\ \alpha_k\rho_k > 1. \end{cases} \tag{4.83}$$

This set is graphically displayed on Figure 4.9. As we can see, choosing the value of ρ_k close to be close to one (to ensure correlation) leads to high values of α_k.

4.5.3 Experimental work

4.5.3.1 On synthetic data

For illustrative purposes, we apply the previously described MM algorithm to the synthetic dataset used in Section 4.4.4. We set $K = 1$ to observe the variations in the learned $\mathbf{H}$ w.r.t. ρ. The value of α is set to $(1-\rho)^{-1} + 1$ to enforce $1 - \gamma > 0$. The value of β is set to 1. The algorithms is run for a total of 1000 iterations.

Figure 4.10 displays the evolution of the cost function C w.r.t. the number of iterations with $\alpha = 0.9$ and $\rho = 11$. As ensured by the MM framework, we have a convergent algorithm. Figure 4.11 displays the learned $\mathbf{H}$ for different values of ρ. As the values of α vary with the values of ρ, so does the scale of $\mathbf{H}$. They are therefore renormalized. As expected, as $\rho \to 1$, $\mathbf{H}$ gets smoothed.

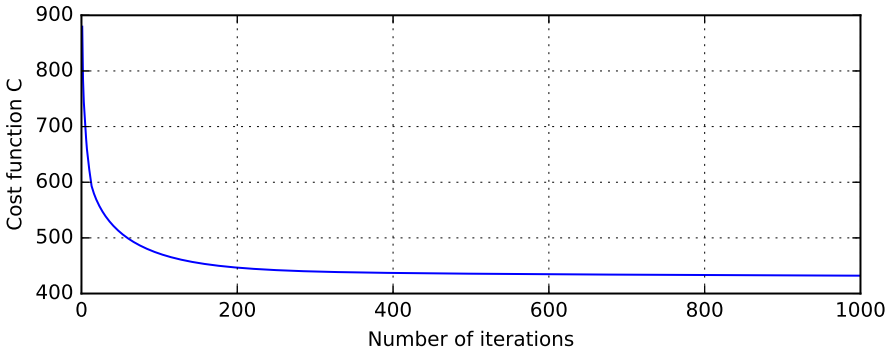


Figure 4.10: Example of the evolution of the cost function C w.r.t. the number of iterations on a synthetic dataset. The hyperparameters are here set to $\rho = 0.9$, $\alpha = 11$ and $\beta = 1$.

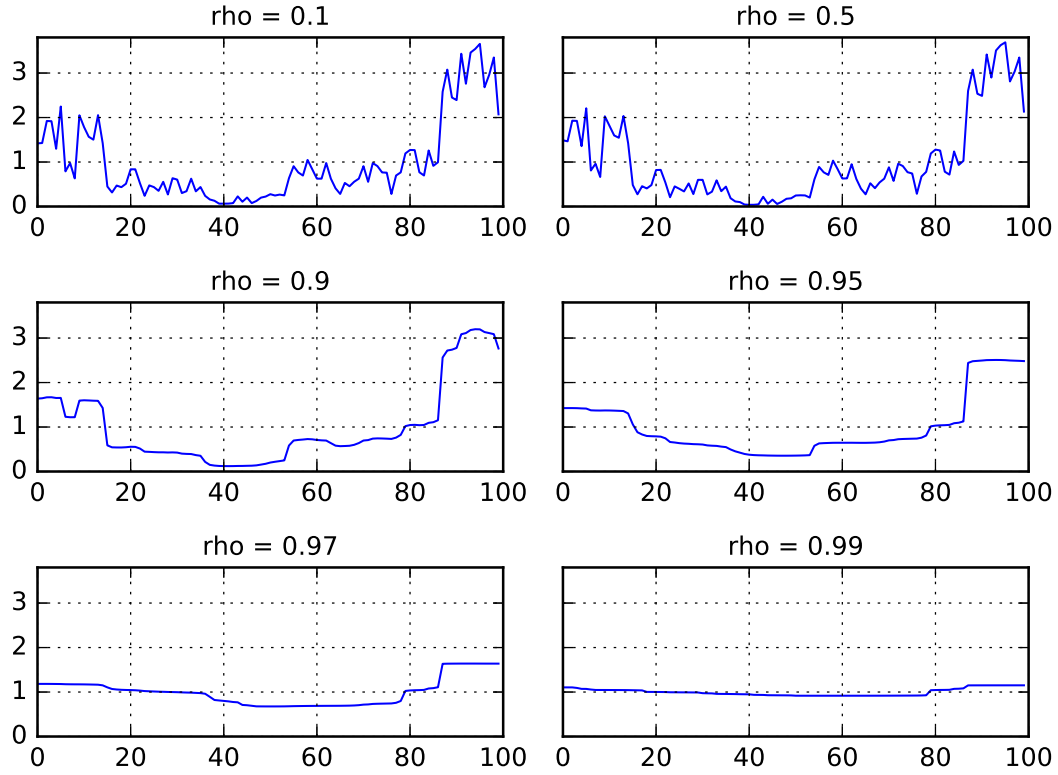


Figure 4.11: Evolution of the learned $\mathbf{H}$ w.r.t. the value of ρ on a synthetic dataset. For all subplots, the value of α is set to $(1-\rho)^{-1}+1$, and β to 1.

Model	MAE	MSE	KL error
GaP (Chap. 2)	17.3 ± 0.38	$3.94 \times 10^3 \pm 3.42 \times 10^2$	$2.72 \times 10^5 \pm 8.69 \times 10^3$
Rate (4.2.1)	13.1 ± 0.58	$4.45 \times 10^3 \pm 1.19 \times 10^3$	$2.02 \times 10^5 \pm 1.18 \times 10^4$
Hierarchical (4.2.2)	$\mathbf{9.34 \pm 0.28}$	$8.96 \times 10^2 \pm 1.93 \times 10^2$	$\mathbf{1.36 \times 10^5 \pm 6.05 \times 10^3}$
Shape (4.2.3)	12.5 ± 0.28	$1.88 \times 10^3 \pm 2.25 \times 10^2$	$1.93 \times 10^5 \pm 5.40 \times 10^3$
BGAR (4.2.4)	$\mathbf{9.26 \pm 0.15}$	$\mathbf{8.18 \times 10^2 \pm 6.85 \times 10^1}$	$\mathbf{1.36 \times 10^5 \pm 4.95 \times 10^3}$

Table 4.1: Prediction results on the `NIPS` dataset. Lower values are better. The mean and standard deviation of each error is reported over 100 runs (10 different splits with 10 different initializations).

4.5.3.2 On a real dataset

We are now going to compare the performance of the proposed BGAR-NMF model on prediction tasks w.r.t. the performance of all the other temporal models presented in Section 4.2 on two real datasets. The performance of the Gamma-Poisson model of Chapter 2, i.e., a model with a non-temporal prior on $\mathbf{H}$, will be considered as well.

For a fair comparison, all these models are going to be assessed within a MAP framework. The algorithms describing MAP estimation in the three other temporal models are reported in Appendix 4.E. MAP estimation in the Gamma-Poisson model is described in Appendix 4.F.

The experimental protocol is as follows. The data matrix is randomly split in 3 subsets: 80% for the training set, 10% for the validation set, and the remaining 10% for the test set. That is to say that the algorithms are applied to the training set, and the predictive performance is assessed on the validation set at each iteration using the mean absolute error (MAE) between the original value v_{fn} and its associated estimate $\hat{v}_{fn} = [\mathbf{WH}]_{fn}$. We then resort to early stopping, i.e., the algorithms are stopped as soon as the MAE on the validation set increases.

We consider the `NIPS` dataset (11463×29) presented in Section 4.4.4. We apply the previously described experimental protocol to 10 random splits, with 10 different initialization for each split. Setting $K = 10$, the averaged MAE, MSE, and KL error on the test sets are then reported on Table 4.1 Note that each model has been assessed with different values of hyperparameters, and Table 4.1 reports the best obtained performance.

As we can see, for all three evaluation metrics, BGAR-NMF outperforms the other models in the MAP framework. It must be noted that its performance is very close to the performance of the temporal model based on the hierarchical chaining of Cemgil and Dikmen (2007). As expected, GaP, the only non-temporal model, produces the worst performance of the five models.

4.6 Discussion

In this chapter, we addressed the problem of designing Markovian temporal NMF models. In these models, the activation coefficients $\mathbf{h}_n$ are no longer independent, but modeled as a Markov chain, which allows to add statistical correlation to the model. We focused on naturally non-negative Markov chains. To this end, we have conducted a review of the NMF literature, which revealed a limitation shared by all models considered until then: a degenerate stationary distribution of the chain. We then presented the BGAR(1) process from the time series literature, which overcomes this limitation, and which, to the best of our knowledge, has never been exploited in learning problems.

We proposed the use of BGAR(1) as a prior distribution for $\mathbf{H}$, combined to a Poisson observation distribution, leading to a novel probabilistic model which we coined BGAR-NMF. We then addressed MMLE in this model with a MCEM algorithm, whose sampling mechanism was based on SMC methods. If our estimation algorithm seemingly works on small-dimensioned synthetic datasets, it fails to produce satisfactory results on real datasets. This is likely linked to samples of poor quality, indicating that it may be difficult to correctly sample from the posterior of the latent variables in this case. An interesting perspective would be to resort to more sophisticated combinations of EM and SMC tools, such as the methodology described in Olsson et al. (2008).

We then turned to an alternative inference paradigm in the BGAR-NMF model, namely MAP estimation. We were able to derive a convergent algorithm for a certain range of values of the hyperparameters of the BGAR(1) chains. It has given satisfactory results both on synthetic and real datasets. In particular, we have shown that BGAR-NMF achieves state-of-the-art performance on a prediction task. In our algorithm, fixing a value of ρ (i.e., the linear correlation between two successive values of the chain) close to 1 restricts us to high values of α. Small values of α are interesting because they induce sparsity, and correspond to the regime where Gaussian approximations do not work. As such, an exciting perspective would be to be able to carry out inference in this regime.

We would also like to emphasize that our MAP algorithm can readily extend to an exponential observation model, as in Chapter 3. Using a similar MM-based scheme would yield higher-order polynomial equations for the updates of the variables.

As a final perspective, we mention the application of our BGAR-NMF model to music recommendation. In this field, data may be stored as a user-song matrix, whose entries represent the songs listening counts of users, such as the `Taste Profile` dataset used in Section 2.6. There also exists similar datasets including temporal information. The listening history of users may then be split in time periods in equal length, yielding a user-item-time *tensor*. The time-dependent user preferences or item attributes may then be modeled with BGAR.

Appendices to Chapter 4

Contents

4.A Moments **133**
4.A.1 Product of independent random variables 133
4.A.2 Laws of total expectation and variance 134
4.B Beta prime distribution **134**
4.C BGAR(1) linear correlation **134**
4.D Bootstrap particle filter **134**
4.E MAP estimation in temporal NMF models **135**
4.E.1 Direct chaining on the rate parameter 136
4.E.2 Hierarchical chaining with an auxiliary variable 136
4.E.3 Chaining on the shape parameter 137
4.F MAP estimation in the GaP model **138**

4.A Moments

4.A.1 Product of independent random variables

Let $X_1, \ldots, X_n$ be n independent random variables. Then we have

$$\mathbb{E}\left(\prod_i X_i\right) = \prod_i \mathbb{E}(X_i). \tag{4.84}$$

Moreover,

$$\begin{aligned}
\operatorname{var}\left(\prod_i X_i\right) &= \mathbb{E}\left(\left(\prod_i X_i\right)^2\right) - \mathbb{E}\left(\prod_i X_i\right)^2 && (4.85)\\
&= \prod_i \mathbb{E}(X_i^2) - \prod_i \mathbb{E}(X_i)^2 && (4.86)\\
&= \prod_i \Big(\mathbb{E}(X_i)^2 + \operatorname{var}(X_i)\Big) - \prod_i \mathbb{E}(X_i)^2. && (4.87)
\end{aligned}$$

4.A.2 Laws of total expectation and variance

We recall the law ot total expectation and the law of total variance. We have

$$\mathbb{E}(X) = \mathbb{E}(\mathbb{E}(X|Y)), \tag{4.88}$$

and

$$\mathrm{var}(X) = \mathbb{E}(\mathrm{var}(X|Y)) + \mathrm{var}(\mathbb{E}(X|Y)). \tag{4.89}$$

4.B Beta prime distribution

Distribution for a continuous random variable in $[0, +\infty[$, with parameters $\alpha > 0$, $\beta > 0$, $p > 0$ and $q > 0$. Its p.d.f. writes, for $x \geq 0$:

$$f(x; \alpha, \beta, p, q) = \frac{p\left(\frac{x}{q}\right)^{\alpha p - 1}\left(1 + \left(\frac{x}{q}\right)^{p}\right)^{-\alpha-\beta}}{q\mathrm{B}(\alpha, \beta)}. \tag{4.90}$$

4.C BGAR(1) linear correlation

We have between two successive values h_n and h_{n+1}:

$$\begin{aligned}
\mathrm{corr}(h_n, h_{n+1}) &= \frac{\mathbb{E}(h_n h_{n+1}) - \mathbb{E}(h_n)\mathbb{E}(h_{n+1})}{\sigma(h_n)\sigma(h_{n+1})} && (4.91)\\
&= \frac{\mathbb{E}(h_n(b_{n+1}h_n + \epsilon_{n+1})) - \mathbb{E}(h_n)\mathbb{E}(h_{n+1})}{\sigma(h_n)\sigma(h_{n+1})} && (4.92)\\
&= \frac{\mathbb{E}(b_{n+1})\mathbb{E}(h_n^2) + \mathbb{E}(h_n)\mathbb{E}(\epsilon_{n+1}) - \mathbb{E}(h_n)\mathbb{E}(h_{n+1})}{\sigma(h_n)\sigma(h_{n+1})} && (4.93)\\
&= \frac{\frac{\alpha\rho}{\alpha\rho+\alpha(1-\rho)}\frac{\alpha(\alpha+1)}{\beta^2} + \frac{\alpha}{\beta}\frac{\alpha(1-\rho)}{\beta} - \frac{\alpha}{\beta}\frac{\alpha}{\beta}}{\frac{\alpha}{\beta^2}} && (4.94)\\
&= \rho. && (4.95)
\end{aligned}$$

4.D Bootstrap particle filter

Particle filtering, in its simplest form, amounts to sequential importance sampling (SIS). This means that a particle approximation of $p(\mathbf{x}_{1:n}|\mathbf{v}_{1:n})$ will be obtained with n sequential importance sampling steps (we begin by sampling $\mathbf{x}_1$, then $\mathbf{x}_2$, and so on and so forth), which, if chosen smartly, allow for a recursive computation of the weights of the particles.

In the so-called bootstrap particle filter, the importance distribution $q(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{v}_n)$ is chosen to be the transition distribution of the HMM, $p(\mathbf{x}_n|\mathbf{x}_{n-1})$. This is convenient from a

Algorithm 6: Bootstrap sequential importance sampling with resampling (SIS/R)

```
# Initialization (n = 1)
for i = 1, ..., N_p do
    x_1^(i) ~ p(x_1)
    ξ̃_1^(i) = p(v_1 | x_1^(i))
end
for n = 2, ..., N do
    # Resampling
    (j_1, ..., j_{N_p}) ~ Mult(N_p, (ξ_{n-1}^(1), ..., ξ_{n-1}^(N_p)))
    for i = 1, ..., N_p do
        x̄_{n-1}^(i) = x_{n-1}^(j_i)
    end
    # Propagation
    for i = 1, ..., N_p do
        x_n^(i) ~ p(x_n | x̄_{n-1}^(i))
        ξ̃_n^(i) = p(v_n | x_n^(i))
    end
end
```

computational point of view, because in this case the unnormalized weights $\tilde{\xi}_n^{(i)}$ associated to particle i at time step n is simply the likelihood $p(\mathbf{v}_n|\mathbf{x}_n^{(i)})$.

The additional resampling step helps to prevent the degeneracy of the importance weights, i.e., when the target distribution is approximated by only one or a few relevant particles. This is a well-known problem from particle filtering methods. We once again refer the interested reader to Cappé et al. (2007) or Doucet and Johansen (2011) for more details.

4.E MAP estimation in temporal NMF models

Throughout this section, a Poisson likelihood is assumed, i.e., we have

$$-\log p(v_{fn}|[\mathbf{WH}]_{fn}) \overset{c}{=} -v_{fn}\log([\mathbf{WH}]_{fn}) + [\mathbf{WH}]_{fn}). \tag{4.96}$$

The majorizations of this term are described in Eq. (4.60) (w.r.t. $\mathbf{H}$) and in Eq. (4.62) (w.r.t. $\mathbf{W}$).

4.E.1 Direct chaining on the rate parameter

We aim at optimizing the following function

$$C(\mathbf{W},\mathbf{H}) = -\sum_{f,n} \log p(v_{fn}|[\mathbf{WH}]_{fn}) - \sum_k \left(\log p(h_{k1}) + \sum_{n\geq 2} \log p(h_{kn}|h_{k(n-1)}) \right), \quad (4.97)$$

where $p(h_{kn}|h_{k(n-1)})$ is given by Eq. (4.4).

We resort to an MM scheme, which amounts to minimizing the following functions

$$f(h_{k1}) \overset{\text{c}}{=} (\alpha_k - p_{k1}) \log(h_{k1}) + q_k h_{k1} + \beta_k \frac{h_{k2}}{h_{k1}}, \quad (4.98)$$

$$f(h_{kn}) \overset{\text{c}}{=} (1 - p_{kn}) \log(h_{kn}) + \left(q_k + \frac{\beta_k}{h_{k(n-1)}} \right) h_{kn} + \beta_k \frac{h_{k(n+1)}}{h_{kn}}, \quad (4.99)$$

$$f(h_{kN}) \overset{\text{c}}{=} (1 - \alpha_k - p_{kN}) \log(h_{kN}) + \left(q_k + \frac{\beta_k}{h_{k(N-1)}} \right) h_{kN}. \quad (4.100)$$

The optimization of these functions boils down to solving an order-2 polynomial equation

$$a_{2,kn} h_{kn}^2 + a_{1,kn} h_{kn} + a_{0,kn} = 0, \quad (4.101)$$

whose coefficients are recapped in the following table.

Variable	$a_{2,kn}$	$a_{1,kn}$	$a_{0,kn}$
h_{k1}	q_k	$a_k - p_{k1}$	$-\beta h_{k2}$
h_{kn}	$q_k + \frac{\beta_k}{h_{k(n-1)}}$	$1 - p_{kn}$	$-\beta_k h_{k(n+1)}$
h_{kN}	0	$q_k + \frac{\beta_k}{h_{k(N-1)}}$	$1 - \alpha_k - p_{kN}$

Table 4.2: Coefficients of the order-2 polynomial equation (Eq. (4.101)) w.r.t. k and n.

4.E.2 Hierarchical chaining with an auxiliary variable

We aim at optimizing the following function

$$C(\mathbf{W},\mathbf{H},\mathbf{Z}) = -\sum_{f,n} \log p(v_{fn}|[\mathbf{WH}]_{fn}) - \sum_k \left(\log p(h_{k1}) + \sum_{n\geq 2} \left(\log p(z_{kn}|h_{k(n-1)}) + \log p(h_{kn}|z_{kn}) \right) \right), \quad (4.102)$$

where $p(z_{kn}|h_{k(n-1)})$ and $p(h_{kn}|z_{kn})$ are given by Eqs (4.9)-(4.10).

We resort to an MM scheme, which amounts to the minimization of the following functions. For z_{kn}, we have

$$f(z_{kn}) \overset{\text{c}}{=} (1-\alpha_z-\alpha_h)\log(z_{kn}) + (\beta_z h_{k(n-1)} + \beta_h h_{kn}) z_{kn}. \tag{4.103}$$

It can be easily solved as

$$z_n = \frac{\alpha_z + \alpha_h - 1}{\beta_z h_{k(n-1)} + \beta_h h_{kn}}. \tag{4.104}$$

For h_{kn}, we detail the three sub-cases

$$f(h_{k1}) \overset{\text{c}}{=} -(p_{k1}+\alpha_z)\log(h_{k1}) + (q_k + \beta_z z_{k2}) h_{k1}, \tag{4.105}$$

$$f(h_{kn}) \overset{\text{c}}{=} (1-\alpha_h-\alpha_z-p_{kn})\log(h_{kn}) + (q_k + \beta_h z_{kn} + \beta_z z_{k(n+1)}) h_{kn}, \tag{4.106}$$

$$f(h_{kN}) \overset{\text{c}}{=} (1-\alpha_h-p_{kN})\log(h_{kn}) + (q_k + \beta_h z_{kN}) h_{kN}. \tag{4.107}$$

These can be easily solved as

$$h_{k1} = \frac{p_{k1}+\alpha_z}{q_k + \beta_z z_{k2}}, \quad h_{kn} = \frac{p_{kn}+\alpha_h+\alpha_z-1}{q_k + \beta_h z_{kn} + \beta_z z_{k(n+1)}}, \quad h_{kN} = \frac{p_{kN}+\alpha_h-1}{q_k + \beta_h z_{kN}}. \tag{4.108}$$

Imposing that $\alpha_h > 1$ is a sufficient condition to guarantee non-negativity under the four proposed update rules.

4.E.3 Chaining on the shape parameter

We aim at optimizing the following function

$$C(\mathbf{W},\mathbf{H}) = -\sum_{f,n} \log p(v_{fn}|[\mathbf{WH}]_{fn}) - \sum_k \left(\log p(h_{k1}) + \sum_{n\geq 2} \log p(h_{kn}|h_{k(n-1)}) \right), \tag{4.109}$$

where $p(h_{kn}|h_{k(n-1)})$ is given by Eq. (4.18).

We resort to an MM scheme, which amounts to minimizing the following functions

$$f(h_{k1}) \overset{\text{c}}{=} -p_{k1}\log(h_{k1}) + (q_k - \alpha_k \log(\beta_k h_{k2})) + \log\Gamma(\alpha_k h_{k2}), \tag{4.110}$$

$$f(h_{kn}) \overset{\text{c}}{=} (1-\alpha_k h_{k(n-1)} - p_{kn})\log(h_{kn}) + (q_k + \beta_k - \alpha_k \log(\beta_k h_{k(n+1)}))h_{kn} + \log\Gamma(\alpha_k h_{kn}), \tag{4.111}$$

$$f(h_{kN}) \overset{\text{c}}{=} (1-\alpha_k h_{k(N-1)} - p_{kn})\log(h_{kN}) + (q_k + \beta_k) h_{kN}. \tag{4.112}$$

The optimization of the first two functions is carried out with Newton's method. The optimizing of the third and last function can easily be done and yields

$$h_{kN} = \frac{p_{kN} + \alpha_k h_{N-1} - 1}{q_k + \beta_k} \tag{4.113}$$

4.F MAP estimation in the GaP model

This has first been described in Dikmen and Févotte (2012). We aim at optimizing the following function

$$C(\mathbf{W}, \mathbf{H}) = -\sum_{f,n} \log p(v_{fn}|[\mathbf{WH}]_{fn}) - \sum_{k,n} \log p(h_{kn}), \tag{4.114}$$

where $p(v_{fn}|[\mathbf{WH}]_{fn})$ is a Poisson likelihood (cf. Eq (4.96)), and h_{kn} are Gamma(α_k, β_k) distributed.

We resort to an MM scheme, which then amounts to the minization of the following function

$$f(h_{kn}) = (1 - \alpha_k - p_{kn}) \log(h_{kn}) + (q_k + \beta_k) h_{kn}. \tag{4.115}$$

This function can be optimized in closed form, yielding the update

$$h_{kn} = \frac{p_{kn} + \alpha_k - 1}{q_k + \beta_k}. \tag{4.116}$$

This update ensures the non-negativity of h_{kn} when $\alpha_k > 1$.

Conclusion

Conclusions

In this thesis, we tackled the general problem of maximum marginal likelihood estimation in semi-Bayesian NMF models. In this framework, the dictionary $\mathbf{W}$, a deterministic variable, is estimated by maximizing the marginal likelihood of the model, that is the likelihood where the activation coefficients $\mathbf{H}$ have been integrated out. This problem was first tackled in Dikmen and Févotte (2011, 2012), where an intriguing "self-regularization" phenomenon on the columns of the dictionary was empirically observed, but could not be explained.

In Chapter 2 and Chapter 3, we have studied two particular instances in such models: the Gamma-Poisson (GaP) model and the inverse Gamma complex normal (IGCN) model, respectively. In both cases, we have conducted a similar analysis:

- We were able to rewrite the generative models free of $\mathbf{H}$, which led in turn to a novel expression of the marginal likelihood. This expression revealed a column-wise regularization term on $\mathbf{W}$ in the GaP model, and an element-wise one in the IGCN model.
- We have also proposed EM algorithms for the task of optimizing the marginal likelihood. In the GaP model, the proposed novel variant (EM-C) has been shown to have favorable properties. In the IGCN model, all the proposed variants are novel. If the self-regularization phenomenon clearly occurs in the GaP model, it is much less clear in the IGCN model, where our experimental work showed the somewhat limited practical interest of our method.

In Chapter 4, we addressed the problem of designing temporal Markovian NMF models. To this end, we have conducted a thorough review of the literature. It revealed that all the models considered until now shared the same limitation, namely, a degenerate stationary distribution of the chain. We have proposed the use of an overlooked autoregressive Markov chain from the time series literature, called BGAR(1), which is marginally Gamma distributed. To the best of our knowledge, this particular process has never been used for learning purposes. Combined with a Poisson observation model, it led to a novel NMF model, which we coined BGAR-NMF.

We then tackled maximum marginal likelihood estimation with tools from the sequential Monte Carlo framework. If the method seemingly works on small-dimensioned problems, it does not scale to larger-dimensioned datasets. We then pursued a MAP estimation in this

model, tackled with a MM-based algorithm for a certain range of hyperparameters. This method allowed us to demonstrate the interest of the proposed BGAR-NMF model.

Perspective and future works

We discuss in this section several perspectives of the work conducted in this thesis, gathered in two themes.

Model aspects

In Chapter 2 and Chapter 3, our analysis of the marginal likelihood in the considered models rested upon two elements:

- First of all, we worked with so-called *composite* models, that is we were able to use auxiliary variables $\mathbf{C}$ such that $\sum_k c_{fkn} = v_{fn}$ (or x_{fn}). This is because we considered observation distributions (Poisson, normal) closed under summation. Developing an analysis framework for distributions which do not share this property remains an open challenge at this stage.
- Secondly, the prior distribution of $\mathbf{H}$ was always chosen to be conjugate to the observation distribution (Gamma is conjugate to Poisson, and inverse Gamma is conjugate to normal with known mean). This conjugacy is convenient for computation, but may not always be relevant from the model perspective (see the discussion at the end of Chapter 3). As such, it would be an interesting perspective to see how the estimator behaves with other priors.

Finally, a broader, harder challenge would be to tackle the analysis of the marginal likelihood not in more general settings, by considering families of distributions, such as the exponential family, for instance.

In Chapter 4, we aimed at proposing a well-posed prior distribution to model the temporal evolution of $\mathbf{H}$. We used the BGAR(1) process from the time series literature, which is marginally Gamma distributed. Other processes tailored to specific applications may be conceived. For example, in audio signal processing, the activation coefficients are known to present exponential decay. An exciting perspective would be to develop alternative Markov chains imitating this structure.

Optimization aspects

In all the models discussed in this thesis, when dealing with the optimization of the marginal likelihood, we always relied on the EM algorithm. This constitutes a natural choice, as we are dealing with latent variable models. However, since the posterior of the latent variables was never tractable, we resorted to MCEM variants, leading to algorithms with prohibitive computational costs. Other variants of the EM algorithms have been

considered (see the discussion at the end of Chapter 2), but did not lead to significant improvements.

As such, it would be beneficial to have alternative methods for optimization, i.e., methods breaking out of EM-based schemes. In particular, an attractive perspective would be to optimize the marginal likelihood directly, for example by using stochastic optimization-based schemes. This line of research might also trigger novel ways of evaluating the marginal likelihood.

Appendix A

Résumé Substantiel en Français

Contexte et état de l'art

Les problèmes de factorisation de matrice

Dans de nombreuses situations, les données sont disponibles sous forme matricielle. Considérons une collection de N échantillons $\mathbf{v}_n$ ($n \in \{1, \ldots, n\}$) appartenant à $\mathbb{R}^F$ (c'est-à-dire décrits par F attributs). Ces échantillons peuvent être concaténés colonne par colonne, afin de former une matrice de taille $F \times N$, que l'on note $\mathbf{V}$. Cette matrice $\mathbf{V}$ est appelée matrice d'observation, ou matrice des données.

Certaines techniques d'analyse de ces données peuvent se formuler comme un problème dit de factorisation de matrice. De manière générale, il s'agit de trouver une approximation de $\mathbf{V}$ sous la forme d'un produit de deux matrices :

$$\mathbf{V} \simeq \mathbf{WH}, \tag{A.1}$$

où $\mathbf{W}$ est de taille $F \times K$, et $\mathbf{H}$ est de taille $K \times N$. Ces deux matrices sont conjointement appelées facteurs. Le rang de la factorisation, K, est traditionnellement choisi tel que $K \leq \min(F, N)$, produisant ainsi une approximation de rang faible de la matrice des données. La factorisation de matrice est alors une technique de réduction linéaire de la dimensionnalité, puisque chaque échantillon est approximé par une combinaison linéaire de K éléments de base :

$$\mathbf{v}_n \simeq \sum_{k=1}^{K} h_{kn}\mathbf{w}_k. \tag{A.2}$$

Plus précisément, les colonnes de $\mathbf{W}$ (parfois appelées « atomes ») représentent des éléments caractéristiques ou récurrents des données. La matrice $\mathbf{W}$ est ainsi habituellement appelée « dictionnaire ». Quant aux colonnes de $\mathbf{H}$, elles encodent la proportion de chaque atome nécessaire pour représenter les échantillons. On appelle $\mathbf{H}$ les coefficients d'activation. Ainsi, la factorisation de matrice a pour but de découvrir automatiquement une certaine structure latente des données. Ces méthodes font donc partie des méthodes dites d'apprentissage non-supervisé.

Le problème de factorisation de matrice est traditionnellement formulé comme un problème d'optimisation. Il s'agit de choisir une certaine fonction D quantifiant la dissimilarité entre $\mathbf{V}$ et son approximation $\mathbf{WH}$, que l'on va chercher à minimiser :

$$\min_{\mathbf{W},\mathbf{H}} D(\mathbf{V}|\mathbf{WH}). \tag{A.3}$$

D'autre part, des contraintes additionnelles sur $\mathbf{W}$ ou $\mathbf{H}$ peuvent s'ajouter à la formulation du problème pour des questions d'interprétabilité. C'est le cas de la factorisation en matrices non-négatives (NMF, de l'anglais *non-negative matrix factorization*), que nous détaillons dans la section suivante.

La factorisation en matrices non-négatives

Lorsque la matrice d'observation $\mathbf{V}$ est non-négative (c'est-à-dire à coefficients positifs ou nuls), et que l'on contraint les facteurs $\mathbf{W}$ et $\mathbf{H}$ à être eux aussi non-négatifs, le problème est appelé factorisation en matrices non-négatives :

$$\min_{\mathbf{W}\geq 0,\ \mathbf{H}\geq 0} D(\mathbf{V}|\mathbf{WH}). \tag{A.4}$$

L'ajout de ces contraintes permet deux choses. Premièrement, le dictionnaire $\mathbf{W}$ se situe dans le même espace que les données, et ses colonnes peuvent donc être directement interprétées comme des éléments caractéristiques des données. Deuxièmement, la non-négativité des éléments de $\mathbf{H}$ impose des représentations constructives. En effet, l'échantillon $\mathbf{v}_n$ ne pourra être représenté que par des sommes pondérées des colonnes (non-négatives) de $\mathbf{W}$. Ainsi, la NMF produit des représentations dites « par parties ».

La NMF a trouvé de nombreux champs d'application tels que le traitement du signal audio (pour de la séparation de sources aveugle ou de la transcription automatique), en fouille de données textuelles (pour de la modélisation thématique), ou encore en imagerie hyperspectrale (pour du démélange).

De nombreux choix de mesures de dissimilarité D ont été considérés dans la littérature. On notera en particulier l'étude de la famille paramétrique de la β-divergence, qui permet de généraliser les choix les plus populaires, tels que la divergence euclidienne, la divergence généralisée de Kullback-Leibler (KL), ou la divergence d'Itakura-Saito (IS).

Factorisations probabilistes

Il s'avère que pour de nombreux choix de mesures de dissimilarité D, le problème de minimisation décrit par l'équation (A.4) est équivalent à l'estimation jointe des facteurs $\mathbf{W}$ et $\mathbf{H}$ au sens du maximum de vraisemblance pour un certain modèle statistique décrivant les données. Par exemple, l'utilisation de la divergence KL sous-tend un modèle de Poisson :

$$v_{fn} \sim \text{Poisson}([\mathbf{WH}]_{fn}), \tag{A.5}$$

tandis que l'utilisation de la divergence IS sous-tend un modèle exponentiel

$$v_{fn} \sim \text{Exp}\left(\frac{1}{[\mathbf{WH}]_{fn}}\right). \tag{A.6}$$

Cela ouvre la voie à un paradigme alternatif pour la NMF, à savoir la NMF *probabiliste*, que l'on définit comme des tâches d'apprentissage dans des modèles statistiques dont la loi de $\mathbf{v}_n$ est paramétrisée par le produit $\mathbf{Wh}_n$. Ces modèles de NMF probabilistes regroupent de nombreux modèles de la littérature, tels que des modèles pour les données de compte.

Plusieurs variantes de modèles peuvent alors être considérées.

1. Modèles fréquentistes. Les facteurs $\mathbf{W}$ et $\mathbf{H}$ sont supposés être des paramètres déterministes. La tâche d'apprentissage dans de tels modèles correspond alors à l'estimation jointe au sens du maximum de vraisemblance de $\mathbf{W}$ et $\mathbf{H}$;
2. Modèles bayésiens. Dans ce deuxième cas, les facteurs $\mathbf{W}$ et $\mathbf{H}$ sont traités comme des variables aléatoires possédant une distribution a priori. Il s'agit de la majorité des travaux de la littérature sur la NMF probabiliste ;
3. Modèles semi-bayésiens. Dans ce troisième et dernier cas, seul $\mathbf{H}$ est traité comme une variable aléatoire, et $\mathbf{W}$ est supposé être un paramètre déterministe.

Nous nous consacrons dans cette thèse aux modèles de NMF semi-bayésiens.

Estimation dans les modèles de NMF semi-bayésiens

L'estimation dans les modèles de NMF semi-bayésiens a été traitée dans Dikmen and Févotte (2011) et Dikmen and Févotte (2012). Dans ces deux articles, deux approches ont été systématiquement comparées :

1. Maximiser la vraisemblance jointe $p(\mathbf{V}, \mathbf{H}; \mathbf{W})$, c'est-à-dire :

$$\max_{\mathbf{W},\mathbf{H}} \log p(\mathbf{V}, \mathbf{H}; \mathbf{W}) = \log p(\mathbf{V}|\mathbf{H}; \mathbf{W}) + \log p(\mathbf{H}). \tag{A.7}$$

   Nous appelons cette approche MJLE (de l'anglais *maximum joint likelihood estimation*).

2. Maximiser la vraisemblance marginale $p(\mathbf{V}; \mathbf{W})$, c'est-à-dire lorsque $\mathbf{H}$ a été marginalisé de la vraisemblance jointe :

$$\max_{\mathbf{W}} \log p(\mathbf{V}; \mathbf{W}) = \log \int_{\mathbf{H}} p(\mathbf{V}|\mathbf{H}; \mathbf{W}) p(\mathbf{H}) \mathrm{d}\mathbf{H}. \tag{A.8}$$

   Nous appelons cette approche MMLE (de l'anglais *maximum marginal likelihood estimation*). Notons que l'inférence de la distribution a posteriori $p(\mathbf{H}|\mathbf{V}; \hat{\mathbf{W}})$ peut-être entreprise dans un second temps si nécessaire, où $\hat{\mathbf{W}}$ est l'estimé au sens du maximum de vraisemblance marginale.

Il est clair que l'approche MMLE est statistiquement mieux posée que l'approche MJLE. En effet, dans l'approche MJLE, le nombre de paramètres à estimer, $FK + KN$, croît avec le nombre d'échantillons. Les propriétés d'optimalité statistique de l'estimateur de maximum de vraisemblance ne peuvent pas s'appliquer puisqu'elles requièrent un nombre fixe de paramètre par rapport au nombre d'échantillons. L'approche MMLE ne présente pas ce problème, puisque le nombre de paramètres à estimer est FK.

Il a été empiriquement constaté, sur des jeux de données synthétiques et réels, que l'approche MMLE avait une tendance à régulariser de manière automatique le rang de la factorisation. En particulier, les dictionnaires estimés par l'approche MMLE avaient une tendance à présenter des colonnes de norme négligeable, au contraire de ceux estimés par l'approche MJLE, qui utilisaient toujours les K colonnes.

Ces propriétés avantageuses n'avaient alors pas pu être expliquées dans un cadre théorique. L'étude de l'approche MMLE dans les modèles de NMF semi-Bayésiens est l'objet de cette thèse. Plus précisément, nous traiterons deux types de distribution a priori pour $\mathbf{H}$:

1. Dans un premier temps, nous traiterons l'hypothèse standard d'indépendance des $\mathbf{h}_n$
$$p(\mathbf{H}) = \prod_{n=1}^{N} p(\mathbf{h}_n). \tag{A.9}$$
Les modèles traités aux Chapitres 2 et 3 font partie de cette catégorie.
2. Dans un deuxième temps, nous levons cette hypothèse afin d'ajouter de la corrélation statistique au modèle. En effet, dans certains cas les colonnes de $\mathbf{V}$ ne peuvent pas être traitées comme interchangeables (lorsqu'elles décrivent par exemple un processus temporel). Nous nous intéresserons à une structure de Markov sur les colonnes de $\mathbf{H}$
$$p(\mathbf{H}) = p(\mathbf{h}_1) \prod_{n\geq 2} p(\mathbf{h}_n|\mathbf{h}_{n-1}). \tag{A.10}$$
Cette deuxième catégorie de distribution a priori est l'objet du Chapitre 4.

Résumé du Chapitre 2

Dans ce chapitre, nous entreprenons de maximiser la vraisemblance marginale dans le modèle semi-bayésien suivant

$$h_{kn} \sim \text{Gamma}(\alpha_k, \beta_k), \tag{A.11}$$
$$v_{fn}|\mathbf{h}_n \sim \text{Poisson}([\mathbf{WH}]_{fn}). \tag{A.12}$$

Ce modèle est connu de la littérature de fouille de données textuelles sous le nom de modèle Gamma-Poisson (abrégé GaP). Dans ce domaine, la matrice $\mathbf{V}$ correspond à la représentation « sac de mots » d'un corpus de documents. Cela signifie que l'élément v_{fn} de la matrice correspond au nombre d'occurrences du mot f dans le document n. La matrice $\mathbf{V}$ est ainsi à valeurs entières.

Nos contributions peuvent être résumées en deux points principaux.

Expression analytique de la vraisemblance marginale

Tout d'abord, nous proposons deux réécritures du modèle GaP dans lesquelles la variable $\mathbf{H}$ a été marginalisée. Cela nous permet d'obtenir une expression semi-analytique de la vraisemblance marginale, ce qui était jusqu'alors jugé hors de portée. L'analyse approfondie de cette expression a permis de révéler un terme de régularisation sur les colonnes de $\mathbf{W}$, expliquant ainsi le phénomène de régularisation automatique précédemment observé.

Algorithmes d'optimisation et résultats expérimentaux

Nous comparons trois variantes de l'algorithme EM pour l'optimisation de la vraisemblance marginale. Deux de ces variantes étaient déjà connues de la littérature. Nous proposons une troisième variante fondée sur la marginalisation de la variable $\mathbf{H}$. Nos expériences conduites à la fois sur données synthétiques et données réelles montrent la supériorité de cette troisième variante, en particulier lorsque les données sont sur-dispersées.

Résumé du Chapitre 3

Dans ce chapitre, nous entreprenons de maximiser la vraisemblance marginale dans le modèle semi-bayésien suivant

$$h_{kn} \sim \mathcal{IG}(\alpha_k, \beta_k), \tag{A.13}$$
$$x_{fn}|\mathbf{h}_n \sim \mathcal{CN}(0, [\mathbf{WH}]_{fn}). \tag{A.14}$$

Nous avons baptisé ce modèle génératif IGCN. La matrice $\mathbf{X}$ est ici à valeurs complexes. En posant $v_{fn} = |x_{fn}|^2$, nous retrouvons le modèle d'observation exponentiel tel que décrit par l'équation (A.6). Il s'avère qu'étudier la vraisemblance marginale de $\mathbf{V}$ revient à étudier la vraisemblance marginale de $\mathbf{X}$.

Nos contributions peuvent être résumés en deux points principaux.

Étude de la vraisemblance marginale

Nous proposons une réécriture du modèle IGCN dans laquelle la variable $\mathbf{H}$ a été marginalisée. Contrairement à l'étude entreprise dans le chapitre précédent, nous ne sommes par en mesure de proposer une expression analytique de la vraisemblance marginale. En effet, il subsiste une intégrale insoluble dans l'expression obtenue. Néanmoins, l'analyse de cette expression révèle un terme de régularisation locale sur les éléments de $\mathbf{W}$, au contraire du terme de régularisation par groupe obtenu au chapitre précédent.

Algorithmes d'optimisation et résultats expérimentaux

Nous proposons trois variantes de l'algorithme EM pour l'optimisation de la vraisemblance marginale. Cela constitue une avancée par rapport à l'état de l'art, puisque jusqu'à présent seul un algorithme sans garantie de convergence existait. Nous conduisons une expérience de décomposition audio sur un exemple réel (un enregistrement audio contenant des notes de piano).

Nous utilisons comme point de comparaison la méthode IS-NMF (c'est-à-dire le problème de NMF avec la divergence Itakura-Saito). Nous constatons que le dictionaire retourné par notre méthode ne présente pas particulièrement de structure parcimonieuse. De plus, la performance en décomposition audio est similaire à celle de la méthode de référence, mais pour un coût bien plus élevé. Cela nous amène à conclure que l'intérêt pratique de notre méthode semble limité.

Résumé du Chapitre 4

Dans ce chapitre, nous nous intéressons à la structure suivante pour la loi a priori de $\mathbf{H}$

$$p(\mathbf{H}) = \prod_k p(h_{k1}) \prod_{n \geq 2} p(h_{kn} | h_{k(n-1)}). \tag{A.15}$$

Cela signifie que les lignes de $\mathbf{H}$ sont modélisées par des chaînes de Markov indépendantes. En particulier, nous nous consacrons à l'étude de chaînes de Markov naturellement non-négatives construites autour de la distribution Gamma.

Nos contributions peuvent être résumées en 2 points principaux.

Étude comparative des chaînes de Markov de la littérature

Plusieurs modèles utilisant de telles chaînes de Markov ont été proposées dans la littérature NMF. Nous entreprenons une comparaison exhaustive de ces modèles. Ils peuvent être regroupés en trois catégories :

1. Chaînage sur le paramètre de forme de la distribution Gamma ;
2. Chaînage sur le paramètre d'intensité de la distribution Gamma ;
3. Chaînage sur le paramètre d'intensité de la distribution Gamma avec variable auxiliaire.

Il s'avère que tous ces modèles partagent le même défaut : l'absence d'une distribution stationnaire bien définie. En effet, pour ces trois catégories, la distribution stationnaire est dégénérée. Nous étudions un quatrième type de chaîne issu de la littérature des séries temporelles, appelé « BGAR(1) » (chaîne Bêta-Gamma auto-régressive d'ordre 1). La distribution stationnaire de cette chaîne est bien définie et est une distribution Gamma. Plus particulièrement, la distribution marginale de cette chaîne à tout instant est Gamma.

Travail expérimental

Nous considérons un nouveau modèle génératif, dans lequel toutes les lignes de $\mathbf{H}$ sont supposées être tirées de processus BGAR(1) indépendants. Le modèle d'observation est supposé être Poisson, comme décrit par l'équation (A.5). Nous baptisons ce modèle BGAR-NMF. Ce modèle est une instance des modèles de Markov cachés.

Afin de maximiser la vraisemblance marginale du modèle BGAR-NMF, nous développons un algorithme MCEM. Nous utilisons des méthodes de Monte Carlo séquentielles afin de générer des échantillons de la loi a posteriori des variables latentes. Malheureusement, si cette méthode fonctionne sur des exemples synthétiques de petite dimension, elle ne passe pas à l'échelle sur des jeux de données réels.

Nous entreprenons alors de maximiser la loi a posteriori dans le modèle BGAR-NMF. Pour certaines plages de valeur des hyperparamètres du processus BGAR(1), nous sommes en mesure de développer un algorithme MM pour cette tâche. L'algorithme donne des résultats satisfaisants à la fois sur des jeux de données synthétiques et réels.

Conclusion et perspectives

Dans cette thèse, nous avons attaqué le problème général de la maximisation de la vraisemblance marginale dans les modèles de NMF semi-bayésiens. Dans les Chapitres 2 et 3, nous avons étudié deux instances spécifiques de ces modèles. Dans le Chapitre 4 nous avons proposé un nouveau modèle temporel markovien de NMF fondé sur un processus de la littérature des séries temporelles qui n'avait jusque là jamais été exploité.

Nous envisageons deux perspectives à notre travail.

À propos des hypothèses de modélisation. Premièrement, l'analyse que nous avons conduite au sein des Chapitres 2 et 3 repose sur deux éléments clés : une distribution d'observation composite, et une distribution a priori de $\mathbf{H}$ conjuguée à la distribution d'observation. Un objectif intéressant serait de prolonger cette analyse à des modèles ne respectant pas ces critères, et dans un cadre plus large, à des modèles fondés sur des familles de distributions. Deuxièmement, il serait intéressant de construire d'autres chaînes de Markov, similaires à celle du Chapitre 4, pour des modèles adaptés au traitement du signal audio.

À propos des méthodes d'optimisation. Tous les algorithmes d'optimisation développés dans cette thèse sont fondés sur l'algorithme MCEM, dont le coût computationnel est très élevé. Un perspective majeure consisterait à explorer d'autres paradigmes d'optimisation, afin d'optimiser directement la vraisemblance marginale. En particulier, nous pourrions utiliser des algorithmes d'optimisation stochastique.

Appendix B

Other works

Concurrently to the work presented in this thesis, I have been involved in two collaborations. These two works are briefly discussed in this appendix, in chronological order. The interested reader is referred to the papers themselves for more details.

B.1 Bayesian Mean-Parameterized Non-Negative Binary Matrix Factorization

This work has been carried out in collaboration with Alberto Lumbreras and Cédric Févotte, and has been submitted for publication in December 2018.

Lumbreras et al. (2018)
Lumbreras, A., Filstroff, L., and Févotte, C. (2018). Bayesian mean-parameterized nonnegative binary matrix factorization. *Submitted to Data Mining and Knowledge Discovery*.
Preprint available on arXiv: https://arxiv.org/pdf/1812.06866.pdf

Abstract

This work tackles the analysis of binary data matrices. Binary matrices may represent social networks, voting data, gene expression data, or binary images. Many works of the literature assume a generative model of the following form

$$v_{fn} \sim \text{Bernoulli}(\phi([\mathbf{WH}]_{fn})), \tag{B.1}$$

where $\mathbf{W}$ and $\mathbf{H}$ are unconstrained factors, and ϕ is a link function that maps the factorization to the $[0, 1]$ range, thus ensuring a valid Bernoulli parameter. Although link functions are convenient, they sacrifice the mean-parametrization of the Bernoulli likelihood (i.e., $\mathbb{E}(\mathbf{V}) = \phi(\mathbf{WH}) \neq \mathbf{WH}$), resulting in less interpretable results. We focus in this paper on mean-parametrized models, that is which do not rely on a link function (or, equivalently, consider $\phi = \text{Id}$)

$$v_{fn} \sim \text{Bernoulli}([\mathbf{WH}]_{fn}). \tag{B.2}$$

To guarantee a valid Bernoulli parameter in Eq. (B.2), i.e. $\sum_k w_{fk}h_{kn} \in [0,1]$, we study in this paper three different sets of constraints on $\mathbf{W}$ and $\mathbf{H}$, denoted (c1), (c2) and (c3):

$$h_{kn} \in [0,1], \quad \sum_f w_{fk} = 1, \tag{c1}$$

$$\sum_k h_{kn} = 1, \quad w_{fk} \in [0,1], \tag{c2}$$

$$\sum_k h_{kn} = 1, \quad \sum_f w_{fk} = 1. \tag{c3}$$

In a Bayesian framework, we assume Beta or Dirichlet distributions to enforce these constraints, leading to the following models:

- The `Beta-Dir` model, based on the set of constraints Eq. (c1)

$$h_{kn} \sim \text{Beta}(\alpha_k, \beta_k), \tag{B.3}$$
$$\underline{\mathbf{w}}_f \sim \text{Dir}(\boldsymbol{\gamma}), \tag{B.4}$$
$$v_{fn}|\underline{\mathbf{w}}_f, \mathbf{h}_n \sim \text{Bernoulli}([\mathbf{WH}]_{fn}). \tag{B.5}$$

- The `Dir-Beta` model, based on the set of constraints Eq. (c2)

$$\mathbf{h}_n \sim \text{Dir}(\boldsymbol{\eta}), \tag{B.6}$$
$$w_{fk} \sim \text{Beta}(\alpha_k, \beta_k), \tag{B.7}$$
$$v_{fn}|\underline{\mathbf{w}}_f, \mathbf{h}_n \sim \text{Bernoulli}([\mathbf{WH}]_{fn}). \tag{B.8}$$

- The `Dir-Dir` model, based on the set of constraints Eq. (c3)

$$\mathbf{h}_n \sim \text{Dir}(\boldsymbol{\eta}), \tag{B.9}$$
$$\underline{\mathbf{w}}_f \sim \text{Dir}(\boldsymbol{\gamma}), \tag{B.10}$$
$$v_{fn}|\underline{\mathbf{w}}_f, \mathbf{h}_n \sim \text{Bernoulli}([\mathbf{WH}]_{fn}). \tag{B.11}$$

The first two models are actually symmetric, since the roles of $\mathbf{W}$ and $\mathbf{H}$ are symmetric by transposition of the data matrix $\mathbf{V}$. We end up with two Bayesian mean-parametrized NMF models for binary data, namely the `Beta-Dir` and the `Dir-Dir` models.

The inference revolves around the posterior distributions $p(\mathbf{W}, \mathbf{H}|\mathbf{V})$ in the two aforementioned models. In both cases, we develop a collapsed Gibbs sampler in augmented versions of the models. Collapsed variational inference is also considered. Experimental work is conducted on dictionary learning and prediction tasks. The proposed methods achieve similar or superior performance w.r.t. the state of the art.

B.2 A Ranking Model Motivated by Nonnegative Matrix Factorization with Applications to Tennis Tournaments

This work has been carried out in collaboration with Rui Xia and Vincent Y.F. Tan from National University of Singapore, as well as Cédric Févotte. It has been accepted for publication at the European Conference on Machine Learning (ECML-PKDD) 2019.

Xia et al. (2019)
Xia, R., Tan, V.Y.F., Filstroff, L., and Févotte, C. (2019). A Ranking Model Motivated by Nonnegative Matrix Factorization with Applications to Tennis Tournaments. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*.
Available on arXiv: https://arxiv.org/pdf/1903.06500.pdf

Abstract

In this work, we propose a novel ranking model that combines the classical Bradley-Terry-Luce (BTL) ranking model with NMF.

Consider a sport where a pool of N players may compete against one another, and the only possible outcomes are a win or a loss. In such a framework, the BTL model posits the existence of a "skill level" $\lambda_i \geq 0$ for each player i. Moreover, it assumes that the probability of the event "player i defeats player j" is

$$\mathbb{P}(i \text{ defeats } j) = \frac{\lambda_i}{\lambda_i + \lambda_j}. \tag{B.12}$$

The BTL model is thus parametrized by the vector $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_N]^{\mathrm{T}}$.

We now address a more complex setting by assuming that there exists M different tournaments in which the players can compete. As such, the BTL model naturally extends by considering a skill *matrix* $\boldsymbol{\Lambda}$ of size $M \times N$. The matrix entry λ_{mi} thus represents the skill of player i in tournament m, and we have

$$\mathbb{P}(i \text{ defeats } j \text{ in tournament } m) = \frac{\lambda_{mi}}{\lambda_{mi} + \lambda_{mj}}. \tag{B.13}$$

If no further assumption was made on $\boldsymbol{\Lambda}$, it then would amount to solving M independent BTL models. In this work, we assume that $\boldsymbol{\Lambda}$ is low-rank, that is $\boldsymbol{\Lambda} = \mathbf{WH}$, with $\mathbf{W}$ of size $M \times K$ and $\mathbf{H}$ of size $K \times N$. Indeed, we expect $\boldsymbol{\Lambda}$ to be low-rank as the number of latent variables governing the skills of players is small. For example in tennis, the surface type of the court (i.e., hard, grass or clay) influences the skills of the players.

Denote by $\mathcal{D}$ the dataset representing the outcomes of the games played between N players over M different tournaments. We have

$$\mathcal{D} \overset{\text{def}}{=} \{b_{ij}^{(m)} \in \mathbb{N} \ ; \ (i,j) \in \mathcal{P}_m\}, \tag{B.14}$$

where $\mathcal{P}_m$ denotes the set of games between pairs of players that have played at least once in tournament m, and $b_{ij}^{(m)}$ is the number of times that player i has defeated player j in tournament m.

The likelihood is therefore

$$p(\mathcal{D}; \mathbf{W}, \mathbf{H}) = \prod_{m=1}^{M} \prod_{(i,j)\in\mathcal{P}_m} \left(\frac{[\mathbf{WH}]_{mi}}{[\mathbf{WH}]_{mi} + [\mathbf{WH}]_{mj}} \right)^{b_{ij}^{(m)}}, \tag{B.15}$$

and we will seek $\mathbf{W}$ and $\mathbf{H}$ such that this likelihood is maximized. To this end, we develop a provably convergent, numerically stable MM algorithm.

We conduct our experimental work on two tennis datasets. The official rankings for both professional male and female tennis players are based on a rolling 52-week, cumulative system, where ranking points are earned only from the stage of tournament reached by the players. In particular, one will not be awarded with bonus points by defeating a higher-ranked player (unlike the Elo rating system for chess). Moreover, the current tennis ranking system does not allow to compare dominant players over a long period (e.g., 10 years). These limitations are overcome by our proposed ranking model.

We collected the outcomes of games between $N = 20$ top male and female players in the most important tournaments of respective tours, over a period of 10 years (2008-2017). Our model automatically infers that the surface of the court is a key determinant of the performances of male players, but less so for females.

Bibliography

Acharya, A., Ghosh, J., and Zhou, M. (2015). Nonparametric Bayesian Factor Analysis for Dynamic Count Matrices. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1–9. Cited on pages 110 and 115.

Alquier, P. and Guedj, B. (2017). An Oracle Inequality for Quasi-Bayesian Nonnegative Matrix Factorization. *Mathematical Methods of Statistics*, 26(1):55–67. Cited on page 41.

Alquier, P. and Ridgway, J. (2017). Concentration of tempered posteriors and of their variational approximations. *arXiv e-print arXiv:1706.09293*. Cited on page 94.

Basbug, M. E. and Engelhardt, B. E. (2016). Hierarchical Compound Poisson Factorization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1795–1803. Cited on page 42.

Berger, J. O., Liseo, B., and Wolpert, R. L. (1999). Integrated Likelihood Methods for Eliminating Nuisance Parameters. *Statistical Science*, 14(1):1–28. Cited on page 46.

Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P., and Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(1):155–173. Cited on page 34.

Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B., and Lamere, P. (2011). The million song dataset. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 591–596. Cited on page 72.

Bingham, E., Kabán, A., and Fortelius, M. (2009). The aspect Bernoulli model: multiple causes of presences and absences. *Pattern Analysis and Applications*, 12(1):55–78. Cited on page 42.

Bioucas-Dias, J. M., Plaza, A., Dobigeon, N., Parente, M., Du, Q., Gader, P., and Chanussot, J. (2012). Hyperspectral Unmixing Overview: Geometrical, Statistical, and Sparse Regression-Based Approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):354–379. Cited on page 34.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877. Cited on page 46.

Blei, D. M. and Lafferty, J. D. (2006). Dynamic Topic Models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 113–120. Cited on page 110.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022. Cited on pages 42 and 46.

Buntine, W. and Jakulin, A. (2006). Discrete component analysis. In *Subspace, Latent Structure and Feature Selection*, pages 1–33. Springer. Cited on pages 56, 57, and 60.

Candès, E. J., Wakin, M. B., and Boyd, S. P. (2008). Enhancing sparsity by reweighted $\ell 1$ minimization. *Journal of Fourier Analysis and Applications*, 14(5-6):877–905. Cited on page 62.

Canny, J. (2004). GaP: A Factor Model for Discrete Data. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 122–129. Cited on pages 41, 56, and 57.

Capdevila, J., Cerquides, J., Torres, J., Petitjean, F., and Buntine, W. (2018). A Left-to-Right Algorithm for Likelihood Estimation in Gamma-Poisson Factor Analysis. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, pages 638–654. Cited on page 77.

Cappé, O., Godsill, S. J., and Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924. Cited on pages 122 and 135.

Cappé, O. and Moulines, E. (2009). On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 71(3):593–613. Cited on page 77.

Cappé, O., Moulines, E., and Rydén, T. (2005). *Inference in Hidden Markov Models*. Springer. Cited on page 121.

Cemgil, A. T. (2009). Bayesian Inference for Nonnegative Matrix Factorisation Models. *Computational Intelligence and Neuroscience*, (Article ID 785152). Cited on pages 41, 56, and 67.

Cemgil, A. T. and Dikmen, O. (2007). Conjugate Gamma Markov random fields for modelling nonstationary sources. In *Proceedings of the International Conference on Independent Component Analysis and Signal Separation (ICA)*, pages 697–705. Cited on pages 110, 113, 114, and 131.

Charlin, L., Ranganath, R., McInerney, J., and Blei, D. M. (2015). Dynamic Poisson Factorization. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, pages 155–162. Cited on page 110.

Cichocki, A. and Amari, S.-i. (2010). Families of Alpha- Beta- and Gamma- Divergences: Flexible and Robust Measures of Similarities. *Entropy*, 12(6):1532–1568. Cited on pages 36 and 49.

Cichocki, A., Cruces, S., and Amari, S.-i. (2011). Generalized Alpha-Beta Divergences and Their Application to Robust Nonnegative Matrix Factorization. *Entropy*, 13(1):134–170. Cited on page 50.

Cichocki, A., Lee, H., Kim, Y.-D., and Choi, S. (2008). Non-negative matrix factorization with α-divergence. *Pattern Recognition Letters*, 29(9):1433–1440. Cited on pages 36 and 49.

Cichocki, A., Zdunek, R., and Amari, S.-i. (2006). Csiszar's divergences for non-negative matrix factorization: Family of new algorithms. In *Proceedings of the International Conference on Independent Component Analysis and Signal Separation (ICA)*, pages 32–39. Cited on page 36.

Comon, P. (1994). Independent component analysis, A new concept? *Signal Processing*, 36(3):287–314. Cited on page 46.

Delyon, B., Lavielle, M., and Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, 27(1):94–128. Cited on page 77.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38. Cited on page 63.

Devarajan, K. (2008). Nonnegative Matrix Factorization: An Analytical and Interpretive Tool in Computational Biology. *PL0S Computational Biology*, 4(7). Cited on page 34.

Dikmen, O. and Févotte, C. (2011). Nonnegative dictionary learning in the exponential noise model for adaptive music signal representation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2267–2275. Cited on pages 44, 45, 47, 88, 93, 94, 97, 100, 103, 139, and 145.

Dikmen, O. and Févotte, C. (2012). Maximum marginal likelihood estimation for nonnegative dictionary learning in the Gamma-Poisson model. *IEEE Transactions on Signal Processing*, 60(10):5163–5175. Cited on pages 44, 45, 47, 56, 61, 62, 64, 66, 74, 76, 82, 138, 139, and 145.

Do, T. D. T. and Cao, L. (2018). Gamma-Poisson Dynamic Matrix Factorization Embedded with Metadata Influence. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5829–5840. Cited on page 114.

Donoho, D. and Stodden, V. (2003). When Does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts? In *Advances in Neural Information Processing Systems (NIPS)*, pages 1141–1148. Cited on page 35.

Doucet, A. and Johansen, A. M. (2011). A tutorial on particle filtering and smoothing: Fifteen years later. In *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press. Cited on pages 122 and 135.

Dunn, P. K. and Smyth, G. K. (2005). Series evaluation of tweedie exponential dispersion model densities. *Statistics and Computing*, 15(4):267–280. Cited on page 43.

Fan, J. and Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association*, 96(456):1348–1360. Cited on page 62.

Févotte, C. (2011). Majorization-Minimization Algorithm for Smooth Itakura-Saito Nonnegative Matrix Factorization. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1980–1983. Cited on page 112.

Févotte, C., Bertin, N., and Durrieu, J.-L. (2009). Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural Computation*, 21(3):793–830. Cited on pages 37, 41, 58, 88, 98, 110, and 112.

Févotte, C. and Idier, J. (2011). Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Computation*, 23(9):2421–2456. Cited on pages 36, 38, 49, 97, and 126.

Févotte, C., Le Roux, J., and Hershey, J. R. (2013). Non-negative Dynamical System with Application to Speech and Audio. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3158–3162. Cited on page 110.

Filstroff, L., Lumbreras, A., and Févotte, C. (2018). Closed-form Marginal Likelihood in Gamma-Poisson Matrix Factorization. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 1506–1514. Cited on pages 53 and 55.

Gallager, R. G. (2012). Circularly-symmetric Gaussian random vectors. `http://www.rle.mit.edu/rgallager/documents/CircSymGauss.pdf`. Cited on page 104.

Gaussier, E. and Goutte, C. (2005). Relation between PLSA and NMF and implications. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 601–602. Cited on page 40.

Gaver, D. and Lewis, P. (1980). First-order autoregressive gamma sequences and point processes. *Advances in Applied Probability*, 12(3):727–745. Cited on page 118.

Gillis, N. (2012). Sparse and Unique Nonnegative Matrix Factorization Through Data Preprocessing. *Journal of Machine Learning Research*, 13(Nov):3349–3386. Cited on page 35.

Gillis, N. and Glineur, F. (2012). Accelerated multiplicative updates and hierarchical als algorithms for nonnegative matrix factorization. *Neural Computation*, 24(4):1085–1105. Cited on page 38.

Godsill, S. J., Doucet, A., and West, M. (2004). Monte Carlo Smoothing for Nonlinear Time Series. *Journal of the American Statistical Association*, 99(465):156–168. Cited on page 122.

Gong, C. and Huang, W.-B. (2017). Deep Dynamic Poisson Factorization Model. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1666–1674. Cited on page 110.

Goodman, N. R. (1963). Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction). *The Annals of Mathematical Statistics*, 34(1):152–177. Cited on page 104.

Gopalan, P., Hofman, J. M., and Blei, D. M. (2015). Scalable Recommendation with Hierarchical Poisson Factorization. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 326–335. Cited on pages 41 and 56.

Gopalan, P., Ruiz, F. J., Ranganath, R., and Blei, D. M. (2014). Bayesian Nonparametric Poisson Factorization for Recommendation Systems. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 275–283. Cited on page 56.

Gouvert, O., Oberlin, T., and Févotte, C. (2018). Negative Binomial Matrix Factorization for Recommender Systems. *arXiv e-prints arXiv:1801.01708*. Cited on pages 42 and 72.

Gouvert, O., Oberlin, T., and Févotte, C. (2019). Recommendation from Raw Data with Adaptive Compound Poisson Factorization. In *Proceedings of Uncertainty in Artifical Intelligence (UAI)*. Cited on page 42.

Guillamet, D., Vitria, J., and Schiele, B. (2003). Introducing a weighted non-negative matrix factorization for image classification. *Pattern Recognition Letters*, 24(14):2447–2454. Cited on page 34.

Guo, D., Chen, B., Zhang, H., and Zhou, M. (2018). Deep Poisson Gamma Dynamical Systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8451–8461. Cited on page 110.

Hoffman, M. D., Blei, D. M., and Cook, P. R. (2010). Bayesian Nonparametric Matrix Factorization for Recorded Music. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 439–446. Cited on pages 41 and 88.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441. Cited on page 32.

Huang, K., Sidiropoulos, N. D., and Swami, A. (2014). Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing*, 62(1):211–224. Cited on page 35.

Hunter, D. R. and Lange, K. (2004). A Tutorial on MM Algorithms. *The American Statistician*, 58(1):30–37. Cited on page 38.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons. Cited on page 46.

Jerfel, G., Basbug, M. E., and Engelhardt, B. E. (2017). Dynamic Collaborative Filtering With Compound Poisson Factorization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 738–747. Cited on page 114.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An Introduction to Variational Methodsfor Graphical Model. *Machine Learning*, 37(2):183–233. Cited on page 46.

Jørgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society, Series B (Methodological)*, 49(2):127–145. Cited on pages 42 and 52.

Jørgensen, B. (1997). *The Theory of Dispersion Models*. CRC Press. Cited on page 42.

Kabán, A. and Bingham, E. (2008). Factorisation and denoising of 0–1 data: A variational approach. *Neurocomputing*, 71(10-12):2291–2308. Cited on page 42.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., and Chopin, N. (2015). On Particle Methods for Parameter Estimation in State-space Models. *Statistical Science*, 30(3):328–351. Cited on page 121.

Kim, H. and Park, H. (2008). Nonnegative Matrix Factorization Based on Alternating Non-negativity-constrained Least Squares and the Active Set Method. *SIAM Journal on Matrix analysis and Applications*, 30(2):713–730. Cited on page 38.

Kim, J., He, Y., and Park, H. (2014). Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. *Journal of Global Optimization*, 58(2):285–319. Cited on page 38.

Kim, J. and Park, H. (2011). Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM Journal on Scientific Computing*, 33(6):3261–3281. Cited on page 38.

Kompass, R. (2007). A generalized divergence measure for nonnegative matrix factorization. *Neural Computation*, 19(3):780–791. Cited on pages 36 and 49.

Kuhn, E. and Lavielle, M. (2004). Coupling a stochastic approximation version of EM with an MCMC procedure. *ESAIM: Probability and Statistics*, 8:115–131. Cited on page 77.

Laurberg, H., Christensen, M. G., Plumbley, M. D., Hansen, L. K., and Jensen, S. H. (2008). Theorems on Positive Data: On the Uniqueness of NMF. *Computational Intelligence and Neuroscience*, (Article ID 724206). Cited on page 35.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791. Cited on pages 33 and 39.

Lee, D. D. and Seung, H. S. (2000). Algorithms for Non-negative Matrix Factorization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 556–562. Cited on pages 33 and 126.

Lewicki, M. S. and Sejnowski, T. J. (2000). Learning overcomplete representations. *Neural Computation*, 12(2):337–365. Cited on page 33.

Lewis, P., McKenzie, E., and Hugus, D. K. (1989). Gamma processes. *Communications in Statistics. Stochastic Models*, 5(1):1–30. Cited on pages 111 and 116.

Li, S. Z., Hou, X., Zhang, H., and Cheng, Q. (2001). Learning Spatially Localized, Parts-Based Representation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 207–212. Cited on page 34.

Lin, C.-J. (2007). Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10):2756–2779. Cited on page 38.

Lumbreras, A., Filstroff, L., and Févotte, C. (2018). Bayesian Mean-parameterized Nonnegative Binary Matrix Factorization. *arXiv e-print arXiv:1812.06866*. Cited on pages 42, 53, and 151.

Ma, H., Liu, C., King, I., and Lyu, M. R. (2011). Probabilistic Factor Models for Web Site Recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 265–274. Cited on page 56.

Magron, P., Badeau, R., and Liutkus, A. (2017). Lévy NMF for robust nonnegative source separation. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 259–263. Cited on page 42.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). Online Learning for Matrix Factorization and Sparse Coding. *Journal of Machine Learning Research*, 11(Jan):19–60. Cited on page 31.

Minc, H. (1988). *Nonnegative Matrices*. Wiley. Cited on page 34.

Morris, C. N. (1983). Parametric Empirical Bayes Inference: Theory and Applications. *Journal of the American Statistical Association*, 78(381):47–55. Cited on page 45.

Moulines, E., Cardoso, J.-F., and Gassiat, E. (1997). Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3620. Cited on page 47.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609. Cited on page 31.

Olsson, J., Cappé, O., Douc, R., and Moulines, E. (2008). Sequential Monte Carlo Smoothing With Application to Parameter Estimation In Nonlinear State Space Models. *Bernoulli*, 14(1):155–179. Cited on page 132.

Paatero, P. (1997). Least squares formulation of robust non-negative factor analysis. *Chemometrics and Intelligent Laboratory Systems*, 37(1):23–35. Cited on page 33.

Paatero, P. and Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126. Cited on page 33.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572. Cited on page 32.

Picinbono, B. (1996). Second-order complex random vectors and normal distributions. *IEEE Transactions on Signal Processing*, 44(10):2637–2640. Cited on page 104.

Schein, A., Wallach, H. M., and Zhou, M. (2016). Poisson-Gamma Dynamical Systems. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5005–5013. Cited on page 110.

Schmidt, M. N., Winther, O., and Hansen, L. K. (2009). Bayesian non-negative matrix factorization. In *Proceedings of the International Conference on Independent Component Analysis and Signal Separation (ICA)*, pages 540–547. Springer. Cited on page 41.

Shahnaz, F., Berry, M. W., Pauca, V. P., and Plemmons, R. J. (2006). Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386. Cited on page 34.

Sherman, J. and Morrison, W. J. (1950). Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix. *The Annals of Mathematical Statistics*, 21(1):124–127. Cited on pages 66 and 84.

Sibuya, M., Yoshimura, I., and Shimizu, R. (1964). Negative multinomial distribution. *Annals of the Institute of Statistical Mathematics*, 16(1):409–426. Cited on page 79.

Smaragdis, P. and Brown, J. C. (2003). Non-negative matrix factorization for polyphonic music transcription. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 177–180. Cited on page 33.

Sra, S. and Dhillon, I. S. (2005). Generalized Nonnegative Matrix Approximations with Bregman Divergences. In *Advances in Neural Information Processing Systems (NIPS)*, pages 283–290. Cited on page 36.

Tan, V. Y. F. and Févotte, C. (2013). Automatic relevance determination in nonnegative matrix factorization with the β-divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1592–1605. Cited on pages 43 and 52.

Tweedie, M. C. K. (1984). An index which distinguishes between some important exponential families. In *Proceedings of the Indian Statistical Institute Golden Jubilee International Conference on Statistics: Applications and New Directions*, pages 579–604. Cited on page 43.

Vavasis, S. A. (2009). On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377. Cited on page 35.

Virtanen, T. (2007). Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuityand Sparseness Criteria. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1066–1074. Cited on page 33.

Virtanen, T., Cemgil, A. T., and Godsill, S. (2008). Bayesian Extensions to Non-negative Matrix factorisation for Audio Signal Modelling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1825–1828. Cited on page 114.

Wallach, H. M., Mimno, D. M., and McCallum, A. (2009a). Rethinking LDA: Why priors matter. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1973–1981. Cited on page 46.

Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. (2009b). Evaluation Methods for Topic Models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1105–1112. Cited on page 77.

Wang, F., Li, T., Wang, X., Zhu, S., and Ding, C. (2011). Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery*, 22(3):493–521. Cited on page 34.

Wei, G. C. G. and Tanner, M. A. (1990). A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms. *Journal of the American Statistical Association*, 85(411):699–704. Cited on pages 64 and 94.

Xia, R., Tan, V. Y. F., Filstroff, L., and Févotte, C. (2019). A Ranking Model Motivated by Nonnegative Matrix Factorization with Applications to Tennis Tournaments. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*. Cited on pages 53 and 153.

Xu, W., Liu, X., and Gong, Y. (2003). Document Clustering Based On Non-negative MatrixFactorization. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 267–273. Cited on page 34.

Yoshii, K., Itoyama, K., and Goto, M. (2016). Student's t Nonnegative Matrix Factorization and Positive Semidefinite Tensor Factorization for Single-channel Audio Source Separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 51–55. Cited on pages 42 and 91.

Yılmaz, Y. K. and Cemgil, A. T. (2012). Alpha/beta divergences and Tweedie models. *arXiv e-print arXiv:1209.4280*. Cited on page 43.

Zhang, S., Wang, W., Ford, J., and Makedon, F. (2006). Learning from Incomplete Ratings Using Non-negative Matrix Factorization. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 549–553. Cited on page 34.

Zhao, R. and Tan, V. Y. F. (2018). A Unified Convergence Analysis of the Multiplicative Update Algorithm for Regularized Nonnegative Matrix Factorization. *IEEE Transactions on Signal Processing*, 66(1):129–138. Cited on page 39.

Zhou, M. (2018). Nonparametric Bayesian Negative Binomial Factor Analysis. *Bayesian Analysis*, 13(4):1065–1093. Cited on page 42.

Zhou, M. and Carin, L. (2015). Negative Binomial Process Count and Mixture Modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):307–320. Cited on page 56.

Zhou, M., Hannah, L., Dunson, D., and Carin, L. (2012). Beta-Negative Binomial Process and Poisson Factor Analysis. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1462–1471. Cited on pages 41 and 56.

Şimşekli, U., Cemgil, A. T., and Yılmaz, Y. K. (2013). Learning the beta-Divergence in Tweedie Compound Poisson Matrix Factorization Models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1409–1417. Cited on page 42.