Multivariate analysis of high-throughput sequencing data

Ghislain Durif

To cite this version:

Ghislain Durif. Multivariate analysis of high-throughput sequencing data. Statistics [math.ST]. Université de Lyon, 2016. English. NNT : 2016LYSE1334 . tel-01581175

HAL Id: tel-01581175
https://theses.hal.science/tel-01581175

Submitted on 4 Sep 2017

N^{o} d'ordre NNT : 2016LYSE1334

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
l'Université Claude Bernard Lyon 1

École Doctorale ED341
Evolution Ecosystèmes Microbiologie Modélisation

Spécialité de doctorat : Statistiques

Ghislain DURIF

Multivariate analysis of high-throughput sequencing data

Devant le jury composé de :

Anne-Laure Fougères, Professeur, Université Claude Bernard Lyon 1	Présidente
Jean-Michel Marin, Professeur, Université de Montpellier	Rapporteur
Sylvain Sardy, Professeur, Université de Genève	Rapporteur
Mark van de Wiel, Full Professor, VU University medical center (Asmterdam)	Rapporteur
Marie-Agnès Dillies, Ingénieur de Recherche, Institut Pasteur	Examinatrice
Cédric Févotte, Directeur de Recherche, CNRS/IRIT	Examinateur
Franck Picard, Directeur de Recherche, CNRS/LBBE	Directeur de thèse
Sophie Lambert-Lacroix, Professeur, Université Grenoble-Alpes	Co-directrice de thèse

UNIVERSITE CLAUDE BERNARD - LYON 1

Président de l'Université	**M. le Professeur Frédéric FLEURY**
Prédisent du Conseil Académique	M. le Professeur Hamda BEN HADID
Vice-président du Conseil d'Administration	M. le Professeur Didier REVEL
Vice-président du Conseil Formation et Vie Universitaire	M. le Professeur Philippe CHEVALIER
Vice-président de la Commission Recherche	M. Fabrice VALLÉE
Directeur Général des Services	M. Alain HELLEU

COMPOSANTES SANTE

Faculté de Médecine Lyon Est – Claude Bernard	Directeur : M. le Professeur J. ETIENNE
Faculté de Médecine et de Maïeutique Lyon Sud – Charles Mérieux	Directeur : Mme la Professeure C. BURILLON
Faculté d'Odontologie	Directeur : M. le Professeur D. BOURGEOIS
Institut des Sciences Pharmaceutiques et Biologiques	Directeur : Mme la Professeure C. VINCIGUERRA
Institut des Sciences et Techniques de la Réadaptation	Directeur : M. X. PERROT
Département de formation et Centre de Recherche en Biologie Humaine	Directeur : Mme. la Professeure A-M. SCHOTT

COMPOSANTES ET DEPARTEMENTS DE SCIENCES ET TECHNOLOGIE

Faculté des Sciences et Technologies	Directeur : M. le Professeur F. DE MARCHI
Département Biologie	Directeur : M. le Professeur F. THEVENARD
Département Chimie Biochimie	Directeur : Mme C. FELIX
Département GEP	Directeur : M. H. HAMMOURI
Département Informatique	Directeur : M. le Professeur S. AKKOUCHE
Département Mathématiques	Directeur : M. le Professeur G. TOMANOV
Département Mécanique	Directeur : M. le Professeur H. BEN HADID
Département Physique	Directeur : M. le Professeur J-C PLENET
UFR Sciences et Techniques des Activités Physiques et Sportives	Directeur : M. Y.VANPOULLE
Observatoire des Sciences de l'Univers de Lyon	Directeur : M. B. GUIDERDONI
Polytech Lyon	Directeur : M. le Professeur E. PERRIN
École Supérieure de Chimie Physique Électronique	Directeur : M. G. PIGNAULT
Institut Universitaire de Technologie de Lyon 1	Directeur : M. le Professeur C. VITON
Institut Universitaire de Formation des Maîtres	Directeur : M. le Professeur A. MOUGNIOTTE
Institut de Science Financière et d'Assurances	Directeur : M. N. LEBOISNE

May the 4th be with you.

Abstract

The statistical analysis of Next-Generation Sequencing (NGS) data has raised many computational challenges regarding modeling and inference. High-throughput technologies now allow to monitor the expression of thousands of genes while considering a growing number of individuals, such as hundreds of single cells. Despite the increasing number of observations, genomic data remain characterized by their high-dimensionality. The research directions that will be explored in this manuscript concern hybrid dimension reduction methods that rely on both compression (representation of the data into a lower dimensional space) and variable selection. Developments are made concerning: *i*) the sparse Partial Least Squares (PLS) regression framework for supervised classification, and *ii*) the sparse matrix factorization framework for unsupervised exploration. In both situations, our main purpose will be to focus on the reconstruction and visualization of the complex organization of the data.

In this regard, we tackle particular challenges regarding the development of methods to analyze high-dimensional data, since the dimensionality directly interferes with the optimization procedures. In a first part, we will develop a sparse PLS approach, based on an adaptive sparsity-inducing penalty, that is suitable for logistic regression, e.g. to predict the label of a discrete outcome, such as the fate of patients or the specific type of unidentified single cells based on gene expression profiles. The main issue in such framework is to account for the response when discarding irrelevant variables. We will highlight the direct link between the derivation of the algorithms and the reliability of the results.

In a second part, motivated by questions regarding single-cell data analysis, we will consider the framework of matrix factorization for count data. We propose a model-based approach that is very flexible, and that accounts for over-dispersion as well as zero-inflation (both characteristic of single-cell data). Our matrix factorization method relies on a hierarchical model for which we derive an estimation procedure based on variational inference. In this scheme, we consider variable selection based on a spike-and-slab model suitable for count data. The interest of our procedure for data reconstruction, visualization and clustering will be illustrated in simulation experiments and by presenting preliminary results of an on-going analysis of single-cell data. All proposed statistical methods were implemented into two `R` packages `plsgenomics` and `CMF` based on high performance computing.

Keywords: Computational Statistics, High-dimensional data, Dimension reduction, Compression, Variable selection, Logistic regression, Sparse Partial Least Squares, Probabilistic matrix factorization, Zero-inflated data, Gamma-Poisson model, Variational inference, Next-Generation Sequencing data, Single-cell data.

Résumé

L'analyse statistique de données de séquençage à haut débit (*NGS*) pose des questions computationnelles importantes. Il est aujourd'hui possible d'enregistrer l'expression de milliers de gènes pour un nombre croissant d'individus, comme des centaines de cellules individuelles. Malgré cette augmentation de la taille des échantillons, les données de génomique sont toujours caractérisées par leur grande dimension. Les travaux de recherche présentés dans ce manuscrit portent sur des méthodes de réductions de dimension hybrides, basées sur des approches de compression (représentation dans un espace de faible dimension) et de sélection de variables. Des développements sont menés concernant : *i*) la régression *Partial Least Squares* parcimonieuse pour la classification supervisée, et *ii*) les méthodes de factorisation parcimonieuse de matrices pour l'exploration de données non supervisée. Dans les deux cas, nous nous concentrerons sur la reconstruction et la visualisation des données.

Nous aborderons le développement de méthodes pour l'analyse de données en grande dimension. Les questions de dimensionnalité interfèrent directement avec les procédures d'optimisation. Dans une première partie, nous développerons une approche de type *PLS* parcimonieuse, basée sur une pénalité adaptative, pour la régression logistique (réponse discrète). Cette approche sera par exemple utilisée pour des problèmes de prédiction (devenir de patients ou type cellulaire) à partir de profils d'expression de gènes. Ici, la principale problématique sera de prendre en compte la réponse pour écarter les variables non intéressantes. Nous mettrons en avant le lien direct entre la construction des algorithmes et la fiabilité des résultats.

Dans une seconde partie, motivés par des questions relatives à l'analyse de données *single-cell*, nous considérerons des méthodes de factorisation parcimonieuse de matrices de comptages. Nous proposerons une approche basée sur un modèle hiérarchique flexible qui prend en compte la sur-dispersion et l'amplification des zéros ou *zero-inflation* (caractéristiques des données *single-cell*) et pour lequel nous dérivons une procédure d'estimation basée sur l'inférence variationnelle. Nous introduirons également une procédure de sélection de variables basée sur un modèle *spike-and-slab*. L'intérêt de notre méthode pour la reconstruction, la visualisation et le *clustering* de données sera illustré par des simulations et par des résultats préliminaires concernant une analyse de données *single-cell*. Par ailleurs, toutes les méthodes proposées sont implémentées dans deux packages R : `plsgenomics` et `CMF`.

Mots-clés : Statistiques computationnelles, Données en grande dimension, Réduction de dimension, Compression, Sélection de Variables, Régression logistique, *Partial Least Squares* parcimonieuse, Factorisation probabiliste de matrices, Modèle Gamma-Poisson, Inférence variationnelle, Données de séquençage à haut débit, Données *single-cell*.

Table des matières

Acknowledgements

I will start in English and then jump to French depending on the many people I would like to thank. I hope the reading will not be too long, it is quite difficult to be exhaustive and concise without forgetting anyone.

First, I would like to thank Jean-Michel Marin, Sylvain Sardy and Mark van de Wiel for agreeing to review my PhD work and for the numerous interesting comments that you made regarding my manuscript. I would also like to thank Marie-Agnès Dillies and Cédric Févotte and Anne-Laure Fougères for agreeing to be examiner at my PhD defense. Thank you all (reviewers and examiners) for all the interesting questions after the presentation of my work, many of your comments will help me improve my work in the near future.

I will jump to French now.

–

Je tiens ensuite à remercier très chaleureusement Franck Picard et Sophie Lambert-Lacroix qui m'ont encadré et accompagné pendant mon doctorat. Ces trois années se sont extraordinairement bien passées grâce à vous, merci pour votre disponibilité et votre aide, c'était un plaisir de travailler à vos côtés.

Je souhaite également remercier les membres de mon comité de suivi de thèse : Anne-Laure Boulesteix, Tristan Mary-Huard et Patricia Reynaud-Bouret. Ces deux réunions en fin de première et seconde année ont été très instructives (malgré tous les déboires techniques que nous avons pu rencontrer lors des deux comités), merci pour votre temps et vos conseils.

Je remercie aussi les membres de l'équipe "Stat en Grande Dimension pour la Génomique" (que j'ai vu naître) du LBBE. Je pourrais dire que j'étais là au commencement où nous n'étions que trois. Merci à Magali Dancette, Laurent Jacob, Florian Massip et Laurent Modolo pour leur encouragement et leur aide dans les moments critiques comme sympathiques.

Un merci spécial à tous les membres du LBBE que j'ai côtoyé de près ou de loin et qui contribuent à faire du LBBE un environnement de travail aussi sympathique que stimulant. Dans le désordre : merci à Nathalie Arbasetti, Laetitia Catouaria, Odile Mulet-Marquis et Florence Peyronie qui ont toujours répondu à mes demandes administratives (parfois totalement absurdes) avec le sourire. Merci aux membres du Pôle informatique et notamment à Stéphane Delmotte, Lionel Humblot, Vincent Miéle, Simon Penel et Bruno

Spataro qui m'ont beaucoup aidé et qui ont toujours été très compréhensif même quand je faisais planter le cluster du laboratoire. Merci à Guillaume Gence (mention spéciale), à Philippe Veber pour tout et pour rien (surtout pour rien, il comprendra), aux basketteurs Guillaume Gence[1], Vincent Navratil, Jeff Taly, Adil El Filali (qui a abandonné par peur de la défaite...) et à l'équipe de basket du CC IN2P3. Ce fut salutaire de faire un peu de sport le midi. Merci aux membres du PRABI et notamment Dominique Guyot pour l'aide et pour paraload. Merci aux membres du bureau 205 qui m'ont souvent accueilli (un peu malgré eux) pour une petite pause de milieu d'après-midi (ils se reconnaîtront). Merci aussi à (un peu en vrac) Wandrille Duchemin, Thomas Bigot (alias M. Wikipedia), Michel Lecocq, Amandine Fournier, Héloise Philippon et Fanny Pouyet (pour tous les conseils pré-soutenance), et à beaucoup d'autres. Et enfin, un grand merci à Jean-Pierre Flandrois, Frédéric Jauffrit et Rose von Raesfeldt, avec qui j'ai partagé un bureau, beaucoup de temps et beaucoup de bonne humeur.

Je voudrais aussi remercier Vivian Viallon qui m'a notamment permis de rencontrer Franck et Sophie, c'est un peu grâce à toi que ce projet de thèse a commencé. Un grand merci à Mahendra Mariadissou, Julien Chiquet, Stéphane Robin, Julie Josse, toutes nos discussions m'ont bien aidé à avancer, surtout sur la fin. Merci également aux membres du projet ABS4NGS et à l'État français qui a financé ces trois années par le biais de l'ANR.

Je finirai par remercier infininement celles et ceux qui me supportent (au sens propre comme au figuré) au quotidien depuis plus ou moins longtemps, surtout mes parents qui sont là depuis toujours, ma famille et mes ami(e)s avec une mention très spéciale à Fabienne (merci pour tout) et un grand merci à mes ami(e)s de Lyon et du Mans, notamment celle et ceux qui ont fait le déplacement de loin pour ma soutenance, Alex, Débo et Guillaume.

–

Eventually (and it is really over), thanks to the reader, I hope that you will enjoy reading my work.

[1] double apparition

Summary (French)

Depuis 10 ans, le séquençage de nouvelle génération ou *Next Generation Sequencing (NGS)* a connu un essor sans précédent. Grâce à la réduction des coups des technologies "haut débit", il est maintenant possible d'enregistrer l'expression de milliers de gènes tout en considérant un nombre croissant d'individus. Les technologies les plus récentes permettent même de capturer le matériel génétique de cellules uniques (données *single cell*). Ceci représente une opportunité sans précédent d'explorer la diversité inter-cellulaire dans un organisme ou un tissu. Néanmoins, malgré l'explosion de la taille des échantillons disponibles, les données génomiques restent caractérisées par leur grande dimension, c'est-à-dire que le nombre de variables enregistrées est plus grand que le nombre d'observations dans l'échantillon. Dans ce contexte, il est nécessaire de considérer les données dans leur globalité à l'aide d'analyses multivariées afin de gérer au mieux les dépendances complexes qui sont présentes dans ces données.

Une première étape dans l'analyse de donnée concerne généralement la visualisation, en particulier pour fournir une représentation de ces données dans un espace de petite dimension, laquelle correspondra à un résumé de l'organisation complexe des données. Idéalement, l'étape de visualisation permet de comprendre la structure sous-jacente et les potentielles dépendances dans les données, c'est-à-dire quels individus présentent des caractéristiques analogues ou quelles variables se comportent de manière similaire. En conséquence, la visualisation pose la question du choix d'une géométrie appropriée. Par exemple, Aggarwal et al. (2001) ont étudié le comportement contre-intuitif de différentes métriques dans des espaces de grande dimension. Pour guider ce choix, l'approche du statisticien est souvent de reformuler le problème et de chercher un modèle statistique approprié qui induira une géométrie adaptée à la représentation des données. Dans certains cas, considérer une approche géométrique est équivalent à considérer une approche basée sur un modèle. Par exemple, dans le cas gaussien, la géométrie euclidienne standard est directement liée à la formulation de la log-vraisemblance du modèle. D'autres types de modèles, comme ceux appropriés aux données de comptage, sont liés à d'autres types de géométrie. En particulier, la formulation "moindre carrés" classique ne respectent pas les contraintes spécifiques qui correspondent à des données de comptage ou binaires.

Une fois le modèle statistique spécifié, les approches probabilistes ou *model-based* reposent sur des procédures d'inférence, dans les cas supervisés comme non supervisés. Ces deux *framework* sont généralement basés sur des méthodes d'optimisation qui sont spécifiquement conçues en fonction du type de modèle considéré ou de la géométrie associée. Dans le contexte de données en grande dimension, la dimensionnalité interfère directement

avec les processus d'optimisation, à cause de singularités numériques ou de problèmes d'identifiabilité. Ainsi, développer des méthodes appropriées pour l'analyse de données en grande dimension reste un défi d'un point de vue statistique (Donoho, 2000). Des approches de réduction de dimension basées sur différents paradigmes ont été proposées pour surmonter ces problématiques. Nous allons nous concentrer sur deux types de méthodes pour réduire la dimension : *i*) les méthodes de compression qui cherchent à représenter les données dans un espace de dimension inférieure, et *ii*) les méthodes de sélection de variables qui sont basées sur une hypothèse de parcimonie, à savoir que parmi tous les variables enregistrées, nombreuses ne sont pas informatives, elles peuvent être considérées comme du bruit et ne doivent pas être prises en compte dans le modèle. L"objectif dans les deux cas est d'apprendre la structure sous-jacente ou de sélectionner automatiquement les variables pertinentes. Le domaine des Statistiques en grande dimension est très actif depuis une dizaine d'année, notamment avec l'explosion des volumes de données dans de nombreux secteurs. Aujourd'hui, un large spectre de méthodes existent pour traiter les problématiques liées à la grande dimension. Utiliser de telles méthodologies est maintenant un prérequis à toute analyse.

Dans cette thèse, nous nous concentrons sur des méthodes hybrides, lesquelles combinent compression et sélection de variables dans un processus efficace de réduction de dimension. L'intérêt d'un tel schéma est spécifiquement d'améliorer la réduction de dimension en exploitant les avantages de chaque approche (compression et sélection). Par exemple, dans un contexte d'analyse supervisée, la *PLS* parcimonieuse ou *sparse PLS* (Chun & Keleş, 2010) est une extension de la régression *Partial Least Squares (PLS)*, introduisant une étape de sélection dans une procédure de compression. La régression *PLS* est en effet conçue pour trouver des directions latentes (dans les données) qui expliquent une réponse. En particulier, nous avons développé une approche *sparse PLS* appropriée pour la régression logistique, c'est-à-dire pour prédire le label d'une réponse discrète. Nous avons utilisé cette méthode pour prédire le sort de patients, à partir de l'expression de gènes de tissus tumoraux, ou pour prédire le type cellulaire de cellules uniques non identifiées à partir de leurs profils d'expression. La principale problématique ici est de comprendre comment prendre en compte la réponse (discrète) pour écarter des variables. Une procédure de seuillage éliminant les variables non pertinentes doit spécifiquement dépendre du modèle. De plus, l'intégration d'une telle procédure dans un algorithme d'estimation est un point crucial afin de garantir la stabilité et la fiabilité de notre méthode. Aussi, nous mettons en avant le lien direct existant entre la construction des algorithmes d'estimation et la qualité de l'analyse, spécifiquement à propos de l'interprétation des résultats.

En outre, les algorithmes d'optimisation doivent être conçus avec précaution afin de garantir que leur résultat correspond effectivement à la solution du problème statistique considéré. Cette question est au cœur du champs des statistiques computationnelles, spécifiquement pour le développement et l'implémentation de méthodologies qui seront adaptées à l'analyse de données à grande échelle. Dans le traitement d'un problème complexe, il est souvent possible d'introduire des approximations qui simplifient la construction des procédures d'optimisation. Cependant, cela peut également introduire des imprécisions qui auront un impact sur les résultats de l'analyse. En particulier, les algorithmes d'optimisation sont souvent basés sur des procédures itératives. Dans ce cas, il faut s'assurer que les itérations successives conduisent bien à la solution du problème d'optimisation posé. Par exemple, dans un contexte de problème supervisé, combiner les modèles linéaires généralisés ou *GLM* (McCullagh & Nelder, 1989) avec une méthode de réduction de dimension n'est pas direct. Dans le pire des cas, il devient même impossible de garantir la validité des résultats et la stabilité de la méthode.

De telles questions au sujet de l'optimisation sont aussi posées dans le contexte d'analyses non supervisées. Motivés par des problématiques de représentation et de *clustering* de données non-gaussiennes comme des profils d'expression de gènes, nous avons également travaillé sur le *framework* de la factorisation de matrice parcimonieuse. Généralement définie d'un point de vue algébrique, la factorisation de matrice peut aussi être définie d'un point de vue statistique, dans une optique de réduction de dimension. Afin d'analyser des données de comptage, comme des profils d'expression de gènes, nous avons développé des approches *model-based* qui sont très flexibles. Nous avons conçu une procédure de factorisation de matrice basée sur des modèles adaptés aux comptages. En particulier, notre modèle prend en compte la sur-dispersion et l'amplification des zéros ou *zero-inflation* (proportion élevée de valeurs nulles dans les données), lesquelles sont des caractéristiques des données d'expression de cellules uniques. Cette formulation est notamment liée à une géométrie sous-jacente qui est adaptée à des données de comptage sur-dispersées et *zero-inflated*.

Plus spécifiquement, notre méthode de factorisation de matrice s'appuie sur un modèle hiérarchique, lequel requiert de développer un schéma d'inférence approprié. Dans ce contexte, les méthodes d'estimation standards comme le maximum de vraisemblance ne sont pas utilisables à cause de la complexité du modèle. Cependant, il est possible d'utiliser des méthodes alternatives, notamment pour inférer la distribution a posteriori des variables latentes. Aussi, il est crucial d'introduire les approximations appropriées dans le schéma d'inférence afin de construire des algorithmes computationnellement efficaces. En particulier, si les approximations utilisées sont contrôlées, il est possible de garantir la validité de la procédure d'optimisation. Dans cette optique, nous avons utilisé le *framework* de l'inférence variationnelle. Cette approche est une alternative aux méthodes dites *Markov Chain Monte Carlo (MCMC)* qui ont généralement un coût prohibitif en terme de calcul. À l'inverse, l'inférence variationnelle est utilisée pour inférer une approximation de la distribution a posteriori dans le modèle. Ce *framework* est performant en terme de calcul et est particulièrement adapté pour construire des algorithmes d'inférence qui seront utilisés pour analyser des données en grande dimension comme des données de génomique.

–

De manière plus détaillée, ce travail de thèse est décomposé en deux grandes parties. La première concerne les problèmes de classification supervisée en grande dimension, dans le contexte de la régression logistique. Nous avons introduit une approche innovante basée sur la *sparse PLS* et les modèles linéaires généralisés. Nous avons également développé une nouvelle procédure de sélection à l'aide d'une pénalité adaptative dans le problème définissant la *sparse PLS*. Nous avons combiné cette étape de réduction de dimension avec un algorithme d'estimation régularisé pour la régression logistique. L'intérêt de nos développements méthodologiques est illustré sur des simulations et sur des données expérimentales (expression de gènes, standard et *single-cell*), en particulier concernant la stabilité et l'efficacité de notre approche.

La seconde partie se concentre sur des problématiques liées à l'exploration de données de génomique les plus récentes comme des données d'expression de cellules uniques (*single-cell*). De telles questions semblent standards mais peuvent s'avérer être, en réalité, complexes dans le cas de données *zero-inflated*. L'hypothèse de normalité n'est pas appropriée pour des données de comptage, ce qui a motivé le développement d'alternatives à l'Analyse en Composante Principale (ACP) et à l'algorithme de décomposition en valeurs singulières (*SVD*) pour factoriser des matrices de comptage. Nous avons développé une méthode de factorisation probabiliste de matrice, adaptée aux comptages, en utilisant un modèle à facteurs cachés Gamma-Poisson. Nous avons étendu ce modèle pour traiter le cas de données *zero-inflated*. En complément, nous avons étendu notre approche au cadre de la factorisation parcimonieuse de matrice à l'aide de modèle de sélection de variables probabiliste de type *spike and slab*. L'inférence d'un tel modèle est réalisé à l'aide d'un nouvel algorithme de type *variational EM* que nous avons développé. L'intérêt de notre approche pour la reconstruction et la visualisation de données ainsi que pour le *clustering* est illustré sur des simulations et par des résultats préliminaires d'une analyse en cours de données de cellules uniques.

Les travaux de recherche réalisés pendant ce doctorat ont également mené à la réalisation de deux librairies pour le logiciel et environnement de développement `R`. Les approches basées sur la *sparse PLS* ont été intégrées dans la librairie `plsgenomics` qui est disponible en ligne (`https://cran.r-project.org/`). La méthode de factorisation de matrice de comptage est implémentée dans une nouvelle librairie nommé `CMF` (pour "*Count Matrix Factorization*") et sera bientôt disponible en ligne.

List of acronyms

BCA	Between-Class Analysis.
BIC	Bayesian Information Criterion.
DE	differentially expressed.
DEA	Differential Expression Analysis.
DNA	DeoxyriboNucleic Acid.
ELBO	Evidence Lower Bound.
EM	Expectation-Maximization.
EN	Elastic Net.
GaP	Gamma-Poisson.
GLM	Generalized Linear Model.
GLMNET	Generalized Linear Model penalized by Elastic Net.
GPCA	Generalized PCA.
GPLS	Generalized PLS.
ICA	Independent Component Analysis.
ICL	Integrated Completed Likelihood.
i.i.d.	independent and identically distributed.
IRLS	Iteratively Reweighted Least Squares.

KL	Kullback-Leibler.
k-NN	k-Nearest Neighbours.
LDA	Linear Discriminant Analysis.
logit-PLS	procedure combining Ridge IRLS and PLS for logistic regression.
logit-SPLS	procedure combining Ridge IRLS and sparse PLS for logistic regression.
ls-NMF	Least Squares NMF.
MAP	Maximum a Posteriori.
MCEM	Monte Carlo EM.
MCMC	Markov Chain Monte Carlo.
MDA	Multiple Discriminant Analysis.
mIRLS	Multinomial IRLS.
MLE	Maximum Likelihood Estimation.
mRNA	messenger RiboNucleic Acid.
multinomial-SPLS	procedure combining Ridge mIRLS and sparse PLS for multinomial logistic regression.
NB	Negative Binomial.
NGS	Next-Generation Sequencing.
NMF	Non-negative Matrix Factorization.
NNSC	Non-Negative sparse Coding.
OLS	Ordinary Least Squares.
PCA	Principal Component Analysis.
PCR	Principal Component Regression.
PLS	Partial Least Squares.
PLS-DA	PLS followed by discriminant analysis.

PLS-log	PLS followed by logistic regression.
Poisson-NMF	NMF based on a Poisson model.
RIRLS	Ridge IRLS.
RNA	RiboNucleic Acid.
RNA-seq	RNA sequencing.
SAEM	Stochastic Approximation of EM.
SGPLS	sparse Generalized PLS.
SNMF	sparse NMF.
SNMF/R	sparse NMF with sparsity constraint on the right factor.
SNR	signal-to-noise ratio.
sparse-GaP	sparse Gamma-Poisson.
SPCA	sparse PCA.
SPLS	sparse PLS.
SPLS-DA	sparse PLS followed by discriminant analysis.
SPLS-log	sparse PLS followed by logistic regression.
SSVD	sparse SVD.
SVD	Singular Value Decomposition.
t-SNE	t-Distributed Stochastic Neighbor Embedding.
var-EM	variational-EM.
ZI	zero-inflated.
ZI-GaP	zero-inflated Gamma-Poisson.

Introduction

For 10 years, Next-Generation Sequencing has been on the rise. Thanks to the reduction of the costs of high-throughput technologies, it is now possible to monitor the expression of thousands of genes while considering a growing number of individuals. Some recent technologies are even able to amplify the genetic material of individual cells, leading to the emerging field of single-cell data analyses. This represents an unprecedented possibility to explore the inter-cellular diversity within an organism or a tissue. Although the number of observations that is considered has been increasing quickly, genomic data remain characterized by their high-dimensionality, meaning that the number of recorded variables is larger than the size of the sample. In this context, considering the data globally through multivariate analysis is necessary to handle the complex dependencies that are present in such data.

A first step when analyzing data generally concerns visualization, in particular to provide a representation in a lower dimensional space that will summarize the complex organization of the data. Ideally, the visualization step will allow to understand the latent structure and the potential dependencies in the data, i.e. which individuals present similar features or which variables behave similarly. Consequently, visualization raises the question of the choice of an appropriate geometry. On this matter, Aggarwal et al. (2001) studied the counter-intuitive behavior of different metrics in high dimensional space. To guide this choice, the statistician's perspective is often to restate the problem, and to search for an appropriate statistical model that will induce a suitable geometry to represent the data. In some cases, considering a geometric approach or a model-based approach are equivalent. For instance, in the Gaussian case, the standard Euclidean geometry is directly linked to the likelihood formulation. Other types of models, for example appropriate for count data, are related to other types of geometries. In particular, the least-squares formulation, if not constrained, does dot comply with the specific constraints that correspond to binary or count data.

Once the statistical model has been specified, model based-approaches rely on inference procedures, in supervised or unsupervised cases. Both frameworks are generally based on optimization procedures, that are specifically designed depending on the type of model or on the associated geometry. In the context of high-dimensional data, the dimensionality directly interferes with the optimization methods, due to numerical singularities or identifiability issues. Consequently, developing methods suitable to analyze high-dimensional data remains a statistical challenge (Donoho, 2000). Dimension reduction approaches based on different paradigms have been proposed to overcome such an issue. We will mainly consider two types of dimension reduction methods: *i*) Compression methods that search for a lower dimensional space on which the data can be represented, and *ii*) Variable selection that is rather based on a parsimony hypothesis, meaning that among all recorded variables, a lot are supposed to be uninformative and can be considered as noise to be removed from the model. The objective of both frameworks it to learn the latent structure or select the relevant variables automatically. This field of high dimensional Statistics has been extremely active in the last ten years, especially with the explosion of the volume of data in many domains. Nowadays, a wide variety of methods exist to handle issues related to the high-dimensionality and using such methodologies is a prerequisite to any analysis.

In this manuscript, we will focus on hybrid methods that combine compression and variable selection for an efficient dimension reduction process. The interest of such framework is especially to enhance the dimension reduction by exploiting the advantages of both compression and selection. For instance, in a context of supervised analysis, the sparse PLS (Chun & Keleş, 2010) extends the Partial Least Squares (PLS) regression by introducing a variable selection step into a compression procedure. The PLS was indeed designed to find latent directions (in the data) that explain the response. In particular, we will develop a sparse PLS approach that is suitable for logistic regression, i.e. to predict the label of a discrete outcome. Such a method will be used to predict the fate of a patient, based on the expression of genes coming from tumorous tissues, or to predict the specific type of unidentified single cells, based on their expression profiles. The main issue in such problems is to understand how to account for the response when discarding some variables. A thresholding procedure to cut out irrelevant variables should specifically depend on the model. Integrating such procedure to an estimation algorithm is a crucial point to ensure the stability and reliability of the method. Indeed, we will highlight the direct link between the derivation of the estimation algorithms and the quality of the analysis, especially regarding the interpretation of the results.

Furthermore, optimization algorithms should be derived cautiously to ensure that their output will correspond to the solution of the considered statistical problem. This question is at the core of computational statistics, especially to develop and implement methodologies that will be appropriate for the analysis of large-scale data. When considering a complex problem, it is always tempting to introduce approximations to cut corners and simplify the derivation of the estimation procedures. However, it sometimes remains hazardous to use heuristics or approximations. For example, optimization are often based on iterative procedures. Thus, iterative algorithms have to be derived with caution to ensure that they correspond to the resolution of the considered optimization question. In our supervised problem based on sparse PLS, we will see that combining the framework of Generalized Linear Models (McCullagh & Nelder, 1989) with dimension reduction approaches is tricky. In a bad-case scenario, it even becomes impossible to guarantee the validity of the output, especially regarding the stability of the results.

Such questions about optimization are also raised in the context of unsupervised analyses. Motivated by questions regarding the representation and the clustering of non-Gaussian data such as sequencing-based gene expression profiles, we will also consider the framework of sparse matrix factorization methods that are particularly suitable for compression and selection. The framework of matrix factorization is generally defined in an algebraic context. Nonetheless, it can also be formulated within a statistical framework. To analyze count data such as expression profiles, we will use model-based approaches that are very flexible. We will derive a matrix factorization procedure based on count-specific distributions. In particular, our model will account for over-dispersion (greater variability than expected) and zero-inflation (abnormal proportion of zeros in the data) that particularly characterize single-cell data for example. Such formulation is particularly related to an underlying geometry suitable for zero-inflated and over-dispersed count data.

As we will see, our matrix factorization method will rely on complex hierarchical models which also requires to develop appropriate inference frameworks. In such context, using standard estimation methods such as Maximum Likelihood Estimation is often compromised because of the complexity of the model. However, solutions exist, especially to infer the posterior distribution of the latent variables. In this context, the question about introducing appropriate approximations in the inference procedure is central in order to derive computationally efficient algorithms. We will show that if the scope of the considered approximation is controlled, it is possible to ensure the validity of the optimization procedure. In this regard, we will use variational inference. This approach avoids using Markov Chain Monte Carlo procedures that may have an important computational cost. Instead, the variational framework is used to infer an approximation of the posterior distribution in the model. It is computationally efficient and particularly suitable to derive inference algorithms that can be used to analyze high dimensional data such as genomic data.

This manuscript is organized into two main parts. Part I will concern problems of supervised classification in high-dimension. Chapter 1 will be dedicated to the introduction of dimension reduction methodologies in the context of Generalized Linear Models. We will especially define the principle of the sparse PLS regression. The potential issues in the existing methods that use sparse PLS for classification will be explained to motivate the development of our approach. We will also expound the interest of such an approach in the context of genomic data analysis. Chapter 2 will focus on the framework that we propose. It is based on a new adaptive selection step in sparse PLS that is combined to an existing regularization of the estimation algorithm for logistic regression. The interest of our methodological developments will be illustrated by simulations to assess its performance and by their application to two different experimental data analyses. Finally, the third chapter will conclude the first part of this manuscript and presents some perspectives of work concerning the sparse PLS.

Part II will focus on the problems regarding data exploration in most recent genomic data such as single-cell expression data. Such problems seem quite standard but remain quite problematic in the case of zero-inflated count data. In Chapter 4, we will introduce the framework of matrix factorization and explain the functioning of the historical methods such as Principal Component Analysis to decompose the signal in the data. We will also present the specificity of the single-cell data from the statistical point of view that will motivate the development of a specific dimension reduction method. Chapter 5 will explain why the Gaussian framework is not appropriate for count data. Based on this assessment, we will present the alternatives for model-based matrix factorization suitable for count data, especially by using a Gamma-Poisson factor model. The inference of such model will also be discussed to motivate our choice for variational inference. Finally, sparse matrix factorization and model-based variable selection will be introduced. In Chapter 6, we will detail our Gamma-Poisson factor model and derive a new variational-EM algorithm suitable for the inference of the model. We will then extend the model to account for zero-inflation or to enforce sparsity in the factors and derive the associated inference frameworks. The interest of our procedure for data reconstruction, visualization and clustering will be illustrated in simulation experiments and by presenting preliminary results of an on-going analysis of single-cell data. Eventually, Chapter 7 will conclude the second part and consider some potential directions that we will investigate regarding the improvement of our matrix factorization approach.

The research work during this PhD project leads to the development of two packages for the statistical software `R`. The sparse PLS related approach has been incorporated into the existing `plsgenomics` package that is currently available on the CRAN (`https://cran.r-project.org/`). The method for factorization of count matrices is implemented in a new package named `CMF` (for Count Matrix Factorization) that is currently in testing phase and that will be released on the CRAN very soon.

Part I

Supervised

Chapter 1

Prediction in the context of genomic data

In supervised statistical problems, an observed outcome, denoted as the response, is related to numerous variables, usually called regressors, predictors or covariates. Depending on the type of the outcome, there exists different types of methods to solve various statistical problems such as regression (quantitative outcome) or classification (qualitative outcome). In the first part of this manuscript, we will focus on classification problems, generally dealing with high-dimensional data, i.e. when the number of covariates is larger than the number of observations. We will develop a dimension reduction approach in the context of the Generalized Linear Model that combines a projection of the data into a lower dimensional space with a selection of the relevant covariates. The development of such methodology is motivated by the analysis of genomic data, characterized by their high-dimensionality.

In the following, we will address issues related to the classification of high-dimensional data, mainly concerning:

- algorithm combination: how to combine dimension reduction methods and classification frameworks in order to ensure the convergence and the stability of the procedure;
- calibration: concerns about the tuning of the hyper-parameters in complex models;
- efficiency: how to control the computational cost in the case of large-scale and high-dimensional data sets.

1.1 Supervised analysis in transcriptomics

The functioning of living cells is based on the synthesis of proteins (Alberts et al., 2007). The genome of each organism is composed of genes that encode for specific proteins. In particular, the protein synthesis is driven by the expression of genes thanks to the following process: during the transcription, the DeoxyriboNucleic Acid (DNA) that carries genes is copied into messenger RiboNucleic Acid (mRNA), that is then translated into proteins. In this context, monitoring the expression of genes is a proxy to characterize the cell activity.

The field of transcriptomics studies how genes are expressed in the cell(s) of an organism in order to understand the cellular activity depending on the time and environmental conditions. Since nearly twenty years, it has been possible to measure simultaneously the expression level of numerous genes (up to thousands). Replacing microarray technologies (Brown & Botstein, 1999), high-throughput technologies also known as Next-Generation Sequencing (NGS) (Hawkins et al., 2010; Reuter et al., 2015) have given access to gene expression profiles regarding hundreds of samples (McGettigan, 2013; Wolf, 2013). When analyzing such data, one of the objectives is to understand the expression patterns that are related to a specific biological or a medical condition. For instance, molecular characterization of diseases like cancer has been a hot topic for 15 years (Alon et al., 1999; Dudoit et al., 2002; Guedj et al., 2012) and there has been a huge effort regarding the prediction of the fate of cancer patient (Wang et al., 2015). In standard NGS data, the signal corresponds to an averaged measure over a population of cells (as a sample from a tissue). Thus, the expression levels account for inter-tissue or inter-group variability, contrary to single-cell data data, which will be the subject of Part II, that allow the quantification of the within-population variability.

Different supervised statistical problems are involved when integrating transcriptomic data, such as regression or classification. The objective is especially to relate a quantitative or qualitative response with the expression of genes. This response can be a physiological measure, a patient or tissue status (binary, tumorous vs non-tumorous), a type of disease, etc. For instance, when dealing with a qualitative outcome (classification problem), the objective is to find differences of expression between conditions. A first approach is to consider differential expression analysis (Anders & Huber, 2010) that is based on univariate statistics. Genes are scored and ranked (Bullard et al., 2010) depending on the differences in their expression regarding a condition. However, genomic data are usually composed of thousands of genes but much less individuals, i.e. the number of recorded variables p (as gene expression) is far larger than the sample size n. This situation is known as a high dimensional case. Such data are characterized by complex dependencies between variables. In this context, we will rather focus on multivariate statistical approaches especially suitable for high-dimensional data. Their purpose is to work simultaneously on numerous genes, in order to determine and highlight the latent structures within the data. Some dimension reduction procedures have been particularly developed to analyze genomic data (Antoniadis et al., 2003; Fort & Lambert-Lacroix, 2005; Lê Cao & Le Gall, 2011).

Following the recent advances of sequencing technologies, it is now possible to isolate and sequence the genetic material from a single cell (Stegle et al., 2015). Single-cell data give the opportunity to characterize the genomic diversity between the individual cells of a specific population. Hence, these data are concerned not only by high-dimensionality but also by increasing sample sizes and the relative questions regarding inter-individual diversity (and not just variable dependencies). Moreover, the growing number of observations also constitutes an opportunity to consider more complex models with multi-class outcomes, e.g. the classification of cells according to their types.

1.2 Introduction to some supervised dimension reduction methods

Handling high-dimensional data such as expression profiles constitutes a challenge for classical regression or classification methods (Marimont & Shapiro, 1979; Donoho, 2000). Indeed, high dimensionality is often associated with spurious dependencies between variables, leading to singularities in the optimization processes, with neither unique nor stable solution (Aggarwal et al., 2001; Hastie et al., 2009). This phenomenon is known as the "curse" of high dimensionality. Statistical analyses in this context are challenging and calls for the development of specific tools, especially dimension reduction approaches. In this section, we introduce some statistical approaches that were developed to overcome high dimensionality issues in the context of the Generalized Linear Models (GLMs).

We briefly introduce some notations that will be necessary for the formal definition of the statistical concepts in the following. We observe a sample of size n, denoted by $(\mathbf{x}_i, y_i)_{i=1:n}$, with y_i the response variables and $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$ a set of observations of p covariates. In the following, $\mathbf{y} = (y_1, \dots, y_n)^T$ is the response vector of observations and the matrix $\mathbf{X} = [x_{ij}] \in \mathbb{R}^{n \times p}$ gathers the n observations of the p covariates (i.e. $\mathbf{X} = [\mathbf{x}_1^T, \dots, \mathbf{x}_p^T]^T$). The random variable associated to the response y_i is denoted by Y_i, and the response random vector is defined as $Y = (Y_1, \dots, Y_n)^T$. We eventually define the "completed" covariate matrix of dimension $n \times (p+1)$ that is defined as $\widetilde{\mathbf{X}} = [(1, \dots, 1)^T, \mathbf{X}]$. The additional column of 1 will be useful to consider the intercept in regression problem (c.f. below). The rows of $\widetilde{\mathbf{X}}$ are naturally denoted by $\widetilde{\mathbf{x}}_i$.

1.2.1 GLMs and supervised statistical problems

When the response $\mathbf{y}$ is quantitative, a standard method to model the dependency between $\mathbf{y}$ and the predictors $\mathbf{X}$ is the linear regression, see Seber & Lee (2003) for a complete introduction. The response is assumed to linearly depend on the covariates, the model can be written as $Y_i = \beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}_{\setminus 0} + \varepsilon_i$ (for any i) where $\boldsymbol{\beta} = \{\beta_0, \beta_1 \dots, \beta_p\} = \{\beta_0, \boldsymbol{\beta}_{\setminus 0}\}$ is the vector of linear coefficients and ε_i the error term. The model is rewritten $Y_i = \widetilde{\mathbf{x}}_i^T \boldsymbol{\beta} + \varepsilon_i$ (for any $i = 1, \dots, n$) in order to integrate the intercept β_0 in the linear combination, i.e. matricially $Y = \widetilde{\mathbf{X}}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with the vector of errors $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_n)^T$.

The ℓ_2 loss corresponds to the log-likelihood when assuming the response $(Y_i)_{i=1:n}$ to be i.i.d.[1] Gaussian variables, i.e. $Y_i \sim \mathcal{N}(\widetilde{\mathbf{x}}_i^T \boldsymbol{\beta}, \sigma^2)$ with $\sigma^2 > 0$ constant across $i = 1, \dots, n$. The concept of Gaussian linear model can be extended to the case of a non-Gaussian response, especially to any distribution in the exponential family[2]. The GLM framework (Nelder & Wedderburn, 1972; McCullagh & Nelder, 1989) covers numerous usual distributions, including the Bernoulli and multi-categorical distributions suitable for qualitative and discrete outcomes. We will especially focus on classification problems where the purpose is to predict the label of the response depending on the covariates.

The logistic regression model

In GLMs, the conditional expectation of the response is supposed to depend on a linear combination of predictors $\widetilde{\mathbf{x}}_i^T \boldsymbol{\beta}$ (following the notation previously introduced) through an invertible link function g, i.e.

$$g^{-1}(\mathbb{E}[Y_i \,|\, \mathbf{x}_i]) = \widetilde{\mathbf{x}}_i^T \boldsymbol{\beta}\,,$$

for any $i = 1, \dots, n$. The link function is determined by the parametrization of the response distribution in the exponential family. When considering the case of a binary classification problem, the label variables Y_i takes their values in $\{0, 1\}$ (for instance tumorous versus non-tumorous tissue). The response is therefore assumed to follow a Bernoulli distribution $Y_i \,|\, \mathbf{x}_i \sim \mathcal{B}\big(\pi(\mathbf{x}_i)\big)$ conditionally to the observation $\mathbf{x}_i$. The probability $\pi(\mathbf{x}_i) \in [0, 1]$ depends on $\mathbf{x}_i$, however to simplify the notation it will be denoted as π_i. Following the GLM framework, the random response variable Y_i is related to the predictors through the logistic (or logit) link function:

$$\text{logit}(\pi_i) = \widetilde{\mathbf{x}}_i^T \boldsymbol{\beta}\,,$$

[1] independent and identically distributed

[2] c.f. Appendix Chapter A

with π_i being the conditional probability $\mathbb{P}[Y_i = 1 \,|\, \mathbf{x}_i]$. The logit function is $\text{logit}(x) = \log(x/(1-x))$. The log-likelihood of the model is derived as:

$$\log \mathcal{L}(Y \,|\, \mathbf{X}\,;\, \boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \, \widetilde{\mathbf{x}}_i^T \boldsymbol{\beta} - \log\{1 + \exp\left(\widetilde{\mathbf{x}}_i^T \boldsymbol{\beta}\right)\} \right] \,. \tag{1.1}$$

The coefficients $\boldsymbol{\beta}$ are estimated by Maximum Likelihood Estimation (MLE)[3] (McCullagh & Nelder, 1989). We will also consider the multi-class model, also known as multinomial logistic regression, in the next chapter. It constitutes a direct generalization of the binary case and will be detailed later.

An example of high dimensionality issue

We present a small example illustrating why the high dimensional case is an issue for standard statistical tools.

In the linear model defined previously, the coefficients $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$ can be estimated by Ordinary Least Squares (OLS), i.e. the estimate $\widehat{\boldsymbol{\beta}}_{OLS}$ minimizes the quadratic error[4] $\|\mathbf{y} - \widetilde{\mathbf{X}}\boldsymbol{\beta}\|_2^2$. This problem admits a closed-form solution: $\widehat{\boldsymbol{\beta}}_{OLS} = (\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^T\mathbf{y}$. However, as soon as $p+1 > n$, the matrix $\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}$ becomes singular and the estimate $\widehat{\boldsymbol{\beta}}_{OLS}$ is not defined. In practice, this corresponds to an identifiability issue, indeed, $\widehat{\mathbf{y}}$ defined as $\widetilde{\mathbf{X}}\widehat{\boldsymbol{\beta}}$ is the projection of $\mathbf{y}$ onto the subspace generated by the columns of $\widetilde{\mathbf{X}}$. Thus, if $p+1 > n$ (or if the covariates are highly correlated), the columns of $\widetilde{\mathbf{X}}$ are not linearly independent and the coefficients $\widehat{\boldsymbol{\beta}}$ are not uniquely defined. In this case, another approach is needed to estimate the coefficients.

Regularization

A first option is to consider penalized approaches. It consists in adding a penalty on the vector of parameter $\boldsymbol{\beta}$ when optimizing the loss function associated to a considered methods, e.g. the log-likelihood. Penalized problem are generally written as:

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\text{argmin}} \left\{ \; \log \mathcal{L}(Y \,|\, \mathbf{X}\,;\, \boldsymbol{\beta}) + \nu \, \text{Pen}(\boldsymbol{\beta}) \; \right\},$$

with $\nu > 0$ being a constant of penalization to be tuned. The penalty term $\text{Pen}(\boldsymbol{\beta})$ is defined depending on the purpose of the penalization.

[3] The reader may refer to Aldrich (1997) for an historical review regarding the introduction of the MLE by R.A. Fisher in the 1920s.

[4] $\|\cdot\|_2$ is the ℓ_2 norm, i.e. $\|\mathbf{a}\|_2^2 = \mathbf{a}^T\mathbf{a}$ for any vector $\mathbf{a} \in \mathbb{R}^n$

For instance, regularization methods based on ℓ_2 penalty were developed to overcome numerical singularities. In this context, the Ridge regression (Hoerl & Kennard, 1970) also known as the Tikhonov regularization is defined with a penalty on the ℓ_2-norm of the vector of parameters, i.e. $\text{Pen}(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_2^2$. The Ridge-penalized least squares problem for regression[5] admits a closed-form solution[6]. The matrix $\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}$ is regularized to avoid the singularity issue.

1.2.2 Dimension reduction methods

Different other methodologies based on different paradigms exist to resolve the issues related to high dimensionality. We first recall some of the main strategies for dimension reduction and then introduce a framework known as the Partial Least Squares (PLS) regression that we will adapt to our classification problem.

Compression methods

Compression approaches are based on the assumption that high dimensional data may be represented in a much lower dimensional space. Their aim is to retrieve this lower dimensional structure. The principle is to project the observations into a lower dimensional space and to find the directions that summarize the information contained in the different variables. The standard example is the Principal Component Regression (PCR) where the response is regressed on a few principal components instead of the covariates directly (Jolliffe, 1982). These components are obtained thanks to the Principal Component Analysis (PCA) that construct directions of maximal variability in the data (Abdi & Williams, 2010).

In the first part of this manuscript, we will mainly work with the PLS regression (Wold, 1975; Wold et al., 1983). It solves a linear regression problem and is particularly suitable to deal with highly correlated covariates. PLS constructs new components as linear combinations of the predictors that maximize their covariance with the response. The advantage of the PLS regression over the PCR for instance is that it finds the latent directions within the data that explain the response at best. Such framework is an alternative to least squares estimation in high-dimensional case. Moreover, as in PCA, the PLS components can be used for data representation and visualization.

The PLS regression and its derivatives for classification will be more precisely introduced

[5] $\widehat{\boldsymbol{\beta}}_{Ridge} = \text{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \left\{ \|\mathbf{y} - \widetilde{\mathbf{X}}\boldsymbol{\beta}\|_2^2 + \nu\,\|\boldsymbol{\beta}\|_2^2 \right\}$

[6] $\widehat{\boldsymbol{\beta}}_{Ridge} = (\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}} + \nu\,\text{Id}_{p+1})^{-1}\widetilde{\mathbf{X}}^T\mathbf{y}$ where Id_{p+1} is the identity matrix.

in the following. Moreover, there is a wide diversity of methods that focus on dimension reduction by compression in the unsupervised context. This will be extensively considered in Part II.

Variable selection

Another option to reduce the dimension is to select only some variables that will be assumed to be relevant for the considered problem. In particular, variable selection methods are based on a hypothesis of parsimony within the data. It assumes that only a few relevant variables contribute to the model fit and a huge number of covariates are useless for the model. In this context, the purpose is to "select" these relevant variables and discard the non pertinent ones from the model. In this regard, it is possible to use another class of penalized methods which consist in sparsity-inducing approaches. An example is the Lasso (Tibshirani, 1996), with its ℓ_1 penalty constraint on the norm of the coefficients. It was first introduced in the case of linear regression. When considering the MLE, the optimization problem is written such as:

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\operatorname{argmin}} \left\{ \log \mathcal{L}(Y \mid \mathbf{X}\,;\,\boldsymbol{\beta}) + \nu\,|\boldsymbol{\beta}|_1 \right\},$$

where $|\boldsymbol{\beta}|_1 = \sum_j |\beta_j|$. The interest of this penalization is that coefficients of the less relevant variables are shrunk to zero (Bach et al., 2012). It can be noted that, since the Lasso, other penalized approaches have been developed for variable selection, e.g. SCAD (Fan & Li, 2001), the Fused Lasso (Tibshirani et al., 2005), the adaptive Lasso (Zou, 2006) or the Group Lasso (Yuan & Lin, 2006).

Combination of compression and variable selection

More recently, some methodologies have been developed based on a combination of a compression approach and a variable selection procedure, in order to benefit from the abilities of both framework. In this regard, the sparse PLS regression (Lê Cao et al., 2008; Chun & Keleş, 2010) is a combination of compression and variable selection approaches. It introduces a selection step based on the Lasso in the PLS framework, constructing new components as sparse linear combinations of predictors. Chong & Jun (2005) have empirically shown that using the PLS to rank covariates for variable selection gives better results regarding the accuracy of selection than the Lasso. It occurs as well that combining compression with a "sparse" approach improves the efficiency of prediction and the accuracy of selection. Chun & Keleş (2010) and Chung & Keleş (2010) obtained empirical results that showed better performance of the sparse PLS over the Lasso regarding prediction and selection.

In Part I (Chapters 1 to 3), we will consider the extension of sparse PLS to classification and its incorporation as a dimension reduction approach into the GLM framework (especially for logistic regression). In this context, the main issue is related to the convergence of the estimation algorithm and especially how to ensure the stability of the procedure when combining a dimension reduction approach as sparse PLS (that is already a combination of two approaches) and the estimation procedure for the GLMs.

1.2.3 Definition of the sparse Partial Least Squares regression

Before introducing their extension to the framework of GLMs, we present the PLS and sparse PLS approaches in the context of regression.

PLS regression

The PLS regression was first introduced in the field of chemometrics (Wold, 1975; Wold et al., 1983). It has been then widely used in different domains, including genomic data analysis (Boulesteix & Strimmer, 2007). The PLS is a compression method suitable for linear regression (Höskuldsson, 1988; Tenenhaus, 1998), particularly in the case of correlated covariates. The PLS constructs new components $\mathbf{t}_k \in \mathbb{R}^n$ (for $k = 1, \dots, K$) as linear combinations of the predictors, i.e. $\mathbf{t}_k = \mathbf{X}\mathbf{w}_k$ with the weight vectors $\mathbf{w}_k \in \mathbb{R}^p$, or equivalently $t_{ik} = \sum_j x_{ij}\, w_{jk}$ for any $i = 1, \dots, n$.

The weight vectors $\mathbf{w}_k$ are defined to maximize the empirical covariance between the new components $\mathbf{t}_k$ and the continuous response $\mathbf{y}$, defined as $\widehat{\mathrm{Cov}}(\mathbf{t}_k, \mathbf{y}) \propto \mathbf{w}_k^T (\mathbf{X}_c)^T \mathbf{y}_c$, where $\mathbf{X}_c$ and $\mathbf{y}_c$ are the respective centered versions of $\mathbf{X}$ and $\mathbf{y}$. The optimization problem associated to the PLS also assumes that $\|\mathbf{w}_k\|_2 = 1$ and orthogonality between components, i.e.

$$\begin{cases} \mathbf{w}_k = \underset{\mathbf{w}\in\mathbb{R}^p}{\operatorname{argmin}}\ \mathbf{w}^T (\mathbf{X}_c)^T \mathbf{y}_c\,, \\ \|\mathbf{w}_k\|_2 = 1\,, \\ \mathbf{t}_k \text{ orthogonal to } \mathbf{t}_1, \dots, \mathbf{t}_{k-1}\,. \end{cases} \tag{1.2}$$

Some implementations consider an objective defined by the squared covariance (leading to the same solution), i.e. $\mathbf{w}^T (\mathbf{X}_c)^T \mathbf{y}_c (\mathbf{y}_c)^T \mathbf{X}_c \mathbf{w}$ (Boulesteix & Strimmer, 2007).

Using matrix notations, $\mathbf{t}_k$ and $\mathbf{w}_k$ are the respective columns of the matrix $\mathbf{T}_{n\times K}$ and the matrix $\mathbf{W}_{p\times K}$ so that $\mathbf{T} = \mathbf{X}\mathbf{W}$. The PLS algorithm is iterative, the first weight vector $\mathbf{w}_1$ is computed by using the covariance maximization problem with $\mathbf{X}_c$ and $\mathbf{y}_c$.

Then, the second weight vector $\mathbf{w}_2$ is computed using the "deflated" version of $\mathbf{X}_c$ and $\mathbf{y}_c$, i.e. the residuals of the respective regression of $\mathbf{X}_c$ and $\mathbf{y}_c$ onto $\mathbf{t}_1$, and so on until the components K. At each step k, the covariance maximization problem admits a closed-form solution, $\mathbf{w}_k$ is the dominant singular vector of the empirical covariance matrix $\widehat{\mathrm{Cov}}(\mathbf{X}^{(k)}, \mathbf{y}^{(k)}) \propto (\mathbf{X}^{(k)})^T \mathbf{y}^{(k)}$ where $\mathbf{X}^{(k)}$ and $\mathbf{y}^{(k)}$ are the respective deflated counterparts of $\mathbf{X}_c$ and $\mathbf{y}_c$ at step $k > 1$.

Eventually, the response $\mathbf{y}$ is regressed against the K components $(\mathbf{t}_k)_{k=1:K}$, by considering the linear model $\mathbf{y} = \mathbf{T}\,\mathbf{q} + \widetilde{\boldsymbol{\varepsilon}}$, where $\mathbf{q} \in \mathbb{R}^K$ is the vector of coefficients and $\widetilde{\boldsymbol{\varepsilon}} \in \mathbb{R}^n$ the vector of errors. When plug-in the relation $\mathbf{T} = \mathbf{XW}$ in the previous model, i.e. $\mathbf{y} = \mathbf{XW}\,\mathbf{q} + \widetilde{\boldsymbol{\varepsilon}}$, the estimation of the coefficients in the original linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ is given by $\widehat{\boldsymbol{\beta}} = \mathbf{W}\widehat{\mathbf{q}}$.

When the matrix $\mathbf{X}$ is singular[7], $\boldsymbol{\beta}$ cannot be estimated by least squares optimization, however the dimension K is generally chosen so that $K < \min(n, p)$, therefore the regression of $\mathbf{y}$ onto $\mathbf{T}$ is more likely to be resolvable and used to estimate the linear coefficients in the high dimensional model. Another interest of the PLS is that the components $\mathbf{T}$ and weights $\mathbf{W}$ describe the individuals and covariates (respectively) in a lower dimensional subspace (Phatak & De Jong, 1997), highlighting directions that explain the response. Hence, the PLS components can be used for data representation[8].

The PLS algorithm is able to handle a multivariate response, however we will restrain our analysis to univariate response (a less complex case when defining sparse PLS) as we will see in the next paragraph.

Sparse PLS

The covariates that are not relevant to explain the response are characterized by near zero weights in the PLS components. However, in high-dimension the cumulative contribution of numerous non-pertinent variables may exceed the contributions of the few important covariates, introducing an inherent noise in the model. Selecting the pertinent predictors in the data $\mathbf{X}$ can be a solution to avoid this problem. For instance, Dai et al. (2006) showed that selecting covariates before running PLS improve the results regarding prediction[9]

More recently, it was proposed to directly integrate the selection process in the PLS framework by using a sparsity-inducing approach. Indeed, Lê Cao et al. (2008) or Chun

[7] For instance when the covariates are correlated or in high dimension (i.e. when $p > n$)

[8] similarly to the components from PCA (Abdi & Williams, 2010)

[9] Their application concerned classification of gene expression data.

& Keleş (2010) introduced the sparse PLS regression. The principle is to add a variable selection step within the PLS algorithm. The components are constructed from sparse weight vectors, whose coordinates are required to be null for covariates that are irrelevant to explain the response. The shrinkage of these weights to zero is achieved by using a ℓ_1 norm penalty in the covariance maximization problem, following the Lasso principle (Tibshirani, 1996), i.e.

$$\begin{cases} \widehat{\mathbf{w}}(\nu) = \underset{\mathbf{w}\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ -\widehat{\operatorname{Cov}}(\mathbf{X}\mathbf{w},\mathbf{y}) + \nu_S\, |\mathbf{w}|_1 \right\}, \\ \|\mathbf{w}\|_2 = 1\,, \end{cases} \tag{1.3}$$

where $\nu_S > 0$ is the sparsity penalty parameter. For the moment, we will focus on the computation of the first weight vector $\mathbf{w}_1$. The computation of the weights $\mathbf{w}_2, \dots, \mathbf{w}_K$ is also constrained so that the SPLS components are orthogonal. In particular, the following components $\mathbf{t}_2, \dots, \mathbf{t}_K$ are derived by replacing $\mathbf{X}$ and $\mathbf{y}$ by their deflated counterparts in the problem (1.3) as in the standard PLS algorithm. Since the deflated variables are the residuals from the regression of $\mathbf{y}$ and $\mathbf{X}$ onto the components $\mathbf{t}_1, \dots, \mathbf{t}_{k-1}$, it ensures that the components $\mathbf{t}_k$ will be orthogonal to $\mathbf{t}_1, \dots, \mathbf{t}_{k-1}$.

The main issue with the objective function of the problem (1.3) is that it is not convex[10] and quite difficult to optimize. To overcome this issue, Lê Cao et al. (2011) proposed to use a sparse Singular Value Decomposition (SVD) of the covariance matrix $(\mathbf{X}_c)^T\mathbf{y}_c$ at each iteration of the PLS algorithm. Their approach is inspired from the sparse PCA by Shen & Huang (2008). The SVD[11] of a matrix $\mathbf{A}$ is a procedure that finds the eigenvector of $\mathbf{A}^T\mathbf{A}$. Therefore, the PLS weight vector $\mathbf{w}$ can be derived by computing the SVD of $(\mathbf{X}_c)^T\mathbf{y}_c$. The idea of the sparse SVD is to find sparse approximation of the eigenvectors, so the algorithm from Lê Cao et al. (2011) uses sparse SVD to approximate the dominant eigenvectors of $(\mathbf{X}_c)^T\mathbf{y}_c$ under sparsity constraint. However, in this procedure, the covariance maximization problem does not admit a closed-form solution, and the construction of each SPLS component $\mathbf{t}_k$ requires an iterative optimization to compute the weight vector $\mathbf{w}_k$.

Another method was proposed by Chun & Keleş (2010). They used the alternate direction method (Eckstein & Yao, 2012) to decompose the problem (1.3). The loss in Equation (1.3) is the sum of two terms, a concave loss and a convex penalty, that can be easily optimized separately. The approach consists in separating each term in the objective function with two different arguments. The difference between these two arguments is penalized so that they remain close to each other. The optimization problem

[10] as a sum of a convex term and a concave term

[11] c.f. Chapter 4

associated to the sparse PLS by Chun & Keleş (2010) is therefore:

$$\underset{\mathbf{a}\in\mathbb{R}^p,\,\mathbf{c}\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ -\tau\,\mathbf{a}^T\mathbf{M}^T\mathbf{M}\mathbf{a} + (1-\tau)(\mathbf{c}-\mathbf{a})^T\mathbf{M}^T\mathbf{M}(\mathbf{c}-\mathbf{a}) + \nu_S \sum_{j=1}^{p} |c_j| \right\}, \tag{1.4}$$

where $\mathbf{c} \in \mathbb{R}^p$ and $\mathbf{a} \in \mathbb{R}^p$ are the two arguments separating the objective function. The matrix $\mathbf{M} = (\mathbf{X}_c)^T\mathbf{y}_c$ is proportional to the empirical covariance matrix, the product $\mathbf{M}^T\mathbf{M}$ corresponds to the squared covariance. The parameter $\tau \in [0.5, 1]$ regulates the penalization on the difference between the two arguments $\mathbf{a}$ and $\mathbf{c}$. The parameter $\nu > 0$ is as usual the penalty parameters on the ℓ_1 norm of the vector $\mathbf{c}$. Chung & Keleş (2010) introduce weighted sparse PLS algorihtm that extended this formulation to the weighted ℓ_2 metric case, taking into account heteroskedasticity by using a weighted matrix product, i.e. $\mathbf{M} = (\mathbf{X}_c)^T\mathbf{V}\mathbf{y}_c$, where $\mathbf{V} \in \mathbb{R}^{n\times n}$ is a weighting matrix.

The sparse weight vector $\mathbf{w}$ is given by the optimal $\mathbf{c}$. When the response is univariate, the problem in Equation (1.4) admits a closed-form solution independent from the parameter τ, based on the soft-thresholding operator applied to the coordinates of the covariance vector[12] $\mathbf{M} = (m_j)_{j=1:p} = (\mathbf{X}_c)^T\mathbf{y}_c$. Hence, the coordinates of the vector $\mathbf{w}$ are defined as

$$w_j = \operatorname{sgn}(m_j)\left(|m_j| - \nu\right)_+,$$

where the soft-thresholding operator is defined as $x \mapsto \operatorname{sgn}(x)\,(|x| - \nu)_+$ for any $x \in \mathbb{R}$ and $(\cdot)_+ = \max(0, \cdot)$.

In Appendix Chapter B, we present a new demonstration of this results, based on proximal operators.

1.3 Computational challenges for dimension reduction in GLMs

The PLS and sparse PLS have shown excellent performance in the case of regression with a continuous response (Chun & Keleş, 2010). Therefore, it seems quite natural to question the possible extension of the PLS framework to deal with non-continuous response. However, we will show that this adaptation to classification is not straightforward. We now present the state-of-the-art regarding the use of (sparse) PLS for classification and especially its integration into the GLM framework.

[12] The covariance matrix $\widehat{\operatorname{Cov}}(\mathbf{X}, \mathbf{y})$ is a vector when $\mathbf{y}$ is univariate.

1.3.1 Potential issues with logistic regression

We consider the model of logistic regression introduced in Section 1.2.1.

The IRLS algorithm

The optimization of the likelihood in logistic regression[13] relies on a gradient descent (McCullagh & Nelder, 1989). The iterative optimization constructs a sequence of vectors of coefficients $(\widehat{\boldsymbol{\beta}}^{(m)})_{m\geq 1}$, whose limit $\widehat{\boldsymbol{\beta}}^{\infty}$ (if it exists) is the estimation of $\boldsymbol{\beta}$. In particular, a Newton-Raphson-based algorithm gives an explicit formulation of $(\widehat{\boldsymbol{\beta}}^{(m)})_{m\geq 1}$. Each $\widehat{\boldsymbol{\beta}}^{(m)}$ corresponds to the coefficient in a weighted regression of a pseudo-response, denoted by $\boldsymbol{\xi}^{(m)}$, onto the predictors $\mathbf{X}$. At the iteration m, the pseudo-response is a continuous variable that depends linearly on the predictor thanks to $\widehat{\boldsymbol{\beta}}^{(m-1)}$. The Iteratively Reweighted Least Squares (IRLS) algorithm introduced by Green (1984) was therefore defined as the following successive weighted regression:

$$\left| \begin{aligned} \widehat{\boldsymbol{\beta}}^{(m+1)} &= (\widetilde{\mathbf{X}}^T \mathbf{V}^{(m)} \widetilde{\mathbf{X}})^{-1} \widetilde{\mathbf{X}}^T \mathbf{V}^{(m)} \boldsymbol{\xi}^{(m)} \,, \\ \boldsymbol{\xi}^{(m+1)} &= \widetilde{\mathbf{X}} \widehat{\boldsymbol{\beta}}^{(m)} + \left(\mathbf{V}^{(m)} \right)^{-1} \left[\mathbf{y} - \widehat{\boldsymbol{\mu}}^{(m)} \right] \,. \end{aligned} \right. \tag{1.5}$$

The pseudo-response $\boldsymbol{\xi}^{(m)}$ also depends on the vector of estimated probabilities for each observation $\widehat{\boldsymbol{\pi}}^{(m)} = (\widehat{\pi}_i^{(m)})_{i=1:n}$ with $\widehat{\pi}_i^{(m)} = \text{logit}^{-1}\big(\widetilde{\mathbf{x}}_i^T \widehat{\boldsymbol{\beta}}^{(m)}\big)$. The weighting matrix $\mathbf{V}^{(m)} = \text{diag}(v_i^{(m)})_{i=1:n}$ is the diagonal empirical variance matrix of the true response Y at the step m, i.e. $v_i^{(m)} = \widehat{\pi}_i^{(m)}(1 - \widehat{\pi}_i^{(m)})$, and the regression is weighted by the matrix $\mathbf{V}^{(m)}$.

Recalling the solution of the linear regression problem (c.f. Section 1.2.1), this iterative optimization achieves the successive resolution of a weighted linear regression of the pseudo-response $\boldsymbol{\xi}^{(m)}$ onto the covariates $\widetilde{\mathbf{X}}$. It is noticeable that following the definition of $\boldsymbol{\xi}^{(m)}$ computed at each iteration, the IRLS algorithm produces a pseudo-response $\boldsymbol{\xi}^{\infty}$ as the limit of the sequence $(\boldsymbol{\xi}^{(m)})_{m\geq 1}$, which is of the form $\boldsymbol{\xi}^{\infty} = \widetilde{\mathbf{X}} \widehat{\boldsymbol{\beta}}^{\infty} + \widetilde{\boldsymbol{\varepsilon}}$ where $\widehat{\boldsymbol{\beta}}^{\infty}$ is the solution of the likelihood optimization, and $\widetilde{\boldsymbol{\varepsilon}}$ is a noise vector of covariance matrix $(\mathbf{V}^{\infty})^{-1}$ with $\mathbf{V}^{\infty}$ the limit of the matrix sequence $(\mathbf{V}^{(m)})_{m\geq 1}$.

[13] c.f. Equation (1.1)

Convergence issues

The question of the convergence of the IRLS algorithm is a crucial issue when estimating parameters. Non-convergent methods may lead to unstable and unreliable estimations, impacting analysis interpretation and reproducibility. Even in the case $p < n$, the MLE may not exist (Albert & Anderson, 1984). Moreover, the definition of the IRLS algorithm is an issue in the high-dimensional case as it relies on the inversion of the singular matrix $\mathbf{X}^T\mathbf{X}$. Nonetheless, it is not a GLM-specific problem as other classification methods are affected by high-dimensionality such as k-Nearest Neighbours (k-NN) (Hinneburg et al., 2000) or Linear Discriminant Analysis (LDA) (Bickel & Levina, 2004). Using dimension reduction approaches is necessary to overcome the dimensionality issue. Especially, it has been proposed to use PLS to reduce dimension in the context of logistic regression. However, such combination raises questions regarding algorithmics to ensure the convergence and the stability of the procedure.

As mentioned, the (sparse) PLS is a tool built to solve linear regression problems. We now address the question of its extension to binary classification and logistic regression.

1.3.2 Combining PLS and GLMs

The integration of dimension reduction approaches in the GLM framework is a sensitive question (Antoniadis et al., 2003; Fort et al., 2005). Indeed estimation in the GLM is based on iterative optimization. There exist potential issues regarding convergence of the procedure.

To overcome the convergence issue in the IRLS algorithm, Marx (1996) proposed to solve the weighted least square problem at each iteration of the IRLS algorithm with a weighted PLS regression. This algorithm follows the IRLS scheme but defines $\widehat{\boldsymbol{\beta}}^{(m+1)}$ as estimated by a PLS regression of the pseudo-response $\boldsymbol{\xi}^{(m)}$ onto the covariates $\mathbf{X}$, i.e.

$$\left| \begin{array}{rcl} \widehat{\boldsymbol{\beta}}^{(m+1)} & \leftarrow & \mathrm{PLS}_w(\boldsymbol{\xi}^{(m)}, \widetilde{\mathbf{X}})\,, \\ \boldsymbol{\xi}^{(m+1)} & = & \widetilde{\mathbf{X}}\widehat{\boldsymbol{\beta}}^{(m)} + \left(\mathbf{V}^{(m)}\right)^{-1}\left[\mathbf{y} - \widehat{\boldsymbol{\mu}}^{(m)}\right]\,, \end{array} \right. \tag{1.6}$$

where PLS_w corresponds to the weighted PLS algorithm where the metric in the observation space, i.e. in $\mathbb{R}^n$, is weighted by the variance matrix $\mathbf{V}^{(m)}$. However, it is not clear if such iterative scheme corresponds to the optimization of an objective function, and if so, what would be this objective function. Therefore, there is no guarantee that a solution provided by this method is a good approximation of the MLE. Moreover, the lack of an optimization framework does not ensure that the sequence $(\widehat{\boldsymbol{\beta}}^{(m)})_{m\geq 1}$ will converge[14].

[14] This point is illustrated in the next chapter.

Another approach was proposed by Wang et al. (1999) and Nguyen & Rocke (2002). Instead of introducing the PLS inside the IRLS algorithm, they proposed to achieve the dimension reduction before the logistic regression. Their algorithm uses the PLS regression as a preliminary compression step. The components $(\mathbf{t}_k)_{k=1:K}$ in the subspace of dimension K are then used in the logistic regression instead of the predictors. Therefore, the IRLS algorithm does not deal with high dimensional data (as $K < p$). In this context, the PLS algorithm treats the discrete response as continuous. Such approach seems counter-intuitive as it neglects the definition of PLS to resolve a linear regression problem and it ignores the inherent heteroskedastic context. This algorithm is called PLS-log in the following. It can be noted that Nguyen & Rocke (2002) or Boulesteix (2004) also proposed to use discriminant analysis as a classifier after the PLS step[15]. This method, known as PLS-DA, is not directly linked to the GLM framework but we cite it as an alternative for classification with PLS-based approaches because we will use it as a baseline in our experimental study.

Ding & Gentleman (2005) proposed the Generalized PLS (GPLS) method. They introduced a modification in Marx's algorithm based on the Firth procedure (Firth, 1993), in order to avoid the non-convergence and the potential infinite parameter estimation in logistic regression. However, we will see in our simulations that this algorithm may not converge, since the iterative patterns does not correspond to the optimization of an objective function. Hence, on the contrary to the IRLS algorithm that optimizes the likelihood, the GPLS algorithm is not defined by an explicit optimization criterion over $\boldsymbol{\beta}$.

In order to overcome the optimization issue, Fort & Lambert-Lacroix (2005) proposed to integrate the dimension reduction step after the IRLS algorithm. Indeed, as previously introduced, the pseudo-response $\boldsymbol{\xi}^\infty$ produced by the IRLS algorithm depends on predictors through a linear relation $\boldsymbol{\xi}^\infty = \widetilde{\mathbf{X}}\widehat{\boldsymbol{\beta}}^\infty + \widetilde{\boldsymbol{\varepsilon}}$. Thus, the PLS is appropriate when considering the regression of $\boldsymbol{\xi}^\infty$ onto the predictors $\widetilde{\mathbf{X}}$. The interest of this framework is that the pseudo-response may be seen as a continuous approximation of the discrete outcome $\mathbf{y}$. Therefore, the dimension reduction step (by PLS) constructs latent directions that explain the qualitative response but without applying the PLS regression in a non-standard way. The dimension reduction step does not mess with the IRLS iterations and the procedure benefits from the properties of the standard PLS (Rosipal & Krämer, 2006; Krämer, 2007). Moreover, in order to ensure the convergence of the IRLS algorithm, Fort & Lambert-Lacroix used a Ridge regularization. In the following, we denote this algorithm by logit-PLS.

[15] Regarding this matter, Barker & Rayens (2003) investigated the link between PLS and LDA.

1.3.3 Combining sparse PLS and GLMs

More recently, the question of combining sparse PLS with the GLM framework was addressed in different works, as extensions of the previous PLS-based methods. Based on the sparse PLS algorithm by Chun & Keleş (2010), Chung & Keleş (2010) presented two different approaches. The first one that is called sparse Generalized PLS (SGPLS) is a direct extension from the GPLS algorithm by Ding & Gentleman (2005). It solves the successive weighted least square problems of IRLS using a sparse PLS regression, with the idea that variable selection reduces the model complexity and helps to overwhelm numerical singularities. Unfortunately, our simulations will show that convergence issues remain[16]. The second approach is a generalization of the PLS-log algorithm and uses sparse PLS to reduce the dimension before running the logistic regression on the SPLS components. This method will be called SPLS-log.

Similarly as what was done with the PLS, we cite the SPLS-DA method developed by (Chung & Keleş, 2010) or Lê Cao et al. (2011) based on the two different implementations of the sparse PLS that we introduced in the previous section by Chun & Keleş (2010) or Lê Cao et al. (2008) respectively. They used the sparse PLS as a preliminary dimension reduction step before a discriminant analysis.

1.3.4 Sparse PLS on a continuous pseudo-response?

Our approach that combines compression and variable selection for logistic regression will be an extension of the logit-PLS algorithm by Fort & Lambert-Lacroix (2005). Their framework is the only one that provides guarantees regarding the convergence of the IRLS algorithm, thanks to the Ridge regularization. Moreover, it allows to directly adapt the sparse PLS to the logistic regression problem, thanks to the regression of the continuous pseudo-response. Our method will be called logit-SPLS. An additional interest of such algorithm is that it can be straightforwardly generalized to other distributions in the exponential family. In this regard, we will eventually propose an extension of our logit-SPLS algorithm to the case of a multi-categorical response. Both algorithms will be detailed and discussed in Chapter 2. We will use the sparse PLS algorithm proposed by Chun & Keleş (2010) and introduce an adaptive penalization in order to improve the variable selection step. As a summary to conclude this chapter, the different construction of algorithms that integrate a dimension reduction step by (sparse) PLS in the framework of GLMs are summarized in Table 1.1.

[16] The use of sparse PLS does not resolve the issue link to the absence of an associated optimization problem.

Method	Algorithm	Sparse?	Reference
GPLS	(S)PLS inside the IRLS algorithm	×	Ding & Gentleman (2005)
SGPLS		✓	Chung & Keleş (2010)
PLS-log	(S)PLS before logistic regression	×	Wang et al. (1999), Nguyen & Rocke (2002)
SPLS-log		✓	Chung & Keleş (2010)
logit-PLS	(S)PLS on the pseudo-response after the Ridge IRLS (RIRLS) algorithm	×	Fort & Lambert-Lacroix (2005)
logit-SPLS		✓	**Our algorithm**

Table 1.1 – The different algorithms to process dimension reduction by (sparse) PLS in the framework of the logistic regression.

Chapter 2

Sparse PLS and logistic regression

In this chapter, we will introduce a specific approach that we developed in order to integrate a dimension reduction step in the framework of the logistic regression. In particular, we use the sparse PLS to combine compression and variable selection. Our approach extends the work by Fort & Lambert-Lacroix (2005) based on Partial Least Squares (PLS) regression. The motivation of our algorithmic choice is to ensure the convergence and stability of the estimation procedure and thus the reliability of the interpretation. We also propose an adaptive version of the sparse PLS, inspired from the adaptive Lasso (Zou, 2006), to improve the variable selection accuracy. Using simulations we show the stability and convergence of our method, compared with other state-of-the-art approaches. Especially, we empirically show the interest of our algorithm regarding prediction and selection accuracy. Our method is also more stable regarding the selection of variables and the choice of hyper-parameters by cross validation, on the contrary to other methods processing classification with sparse PLS. In particular, it appears that our approach is the only one that correctly performs considering all criteria (prediction, selection, stability), whereas all the other approaches present a weak spot. More generally, we illustrate the interest of both selection and compression over selection or compression only.

We will eventually present two applications of our approach. The first one will focus on the prediction of the relapse of breast cancer patients based on gene expression profiles. We will show interesting results regarding prediction and data visualization, we will also introduce a new framework regarding the choice of hyper-parameters based on the concept of stability selection developed by Meinshausen & Bühlmann (2010). The second application concerns on-going analysis of single-cell expression profiles. Our method will be used in the context of multi-group classification to identify the phenotype of lymphocyte T cells.

In addition, it can be noted that our methods are implemented in a new version of the `plsgenomics` R-package, released on the CRAN (`https://cran.r-project.org/`).

2.1 Compression and selection in the GLM framework

We define our algorithm that combines logistic regression and sparse PLS. Our objective is to ensure the convergence of the iterative optimization and achieve dimension reduction in the context of binary classification. Eventually, we address the question of hyper-parameter calibration and propose to use cross-validation or stability selection. Let us recall some notations (c.f. Chapter 1), we observe a sample of size n, denoted by $(\mathbf{x}_i, y_i)_{i=1}^n$, with y_i the label variables in $\{0, 1\}$ and $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$ a set of p covariates. In the following, the response and covariates are respectively denoted by $\mathbf{y} = (y_1, \dots, y_n)^T$ and $\mathbf{X} = [\mathbf{x}_1^T, \dots, \mathbf{x}_p^T]^T$. We also define the matrix $\widetilde{\mathbf{X}} = [(1, \dots, 1)^T, \mathbf{X}]$.

2.1.1 Ridge-based maximum likelihood estimation for logistic regression

In the Generalized Linear Model (GLM) framework, the Iteratively Reweighted Least Squares (IRLS) algorithm[1] produces a sequence of pseudo-responses $(\boldsymbol{\xi}^{(m)})_{m\geq 1}$ whose limit $\boldsymbol{\xi}^{\infty}$ (if it exists) linearly depends on the predictor $\widetilde{\mathbf{X}}$. Our first concern is to ensure the existence of the limit and that the algorithm converges toward it. We use an approach developed by Eilers et al. (2001) to regularize the logistic regression.

Stabilizing the IRLS algorithm with a Ridge penalty

When $p < n$, the IRLS algorithm may encounter convergence issues, giving infinite estimates in the case of completely separate or quasi-completely separate data (Albert & Anderson, 1984). If $p > n$, the $n \times (p + 1)$ design matrix $\widetilde{\mathbf{X}}$ is of rank n or less and therefore not full column-rank. Due to identifiability concerns, it implies that the Maximum Likelihood Estimation (MLE) is not unique when it exists, and even may not exist when minimal norm solution is infinite.

[1] Successive weighted linear regressions where $\widehat{\boldsymbol{\beta}}^{(m+1)} = (\widetilde{\mathbf{X}}^T\mathbf{V}^{(m)}\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^T\mathbf{V}^{(m)}\boldsymbol{\xi}^{(m)}$ and $\boldsymbol{\xi}^{(m+1)} = \widetilde{\mathbf{X}}\widehat{\boldsymbol{\beta}}^{(m)} + \left(\mathbf{V}^{(m)}\right)^{-1}\left[\mathbf{y} - \boldsymbol{\mu}^{(m)}\right]$

The convergence of the IRLS procedure can be guaranteed by a Ridge regularization, i.e. a constraint on the ℓ_2-norm of the coefficients. Le Cessie & Van Houwelingen (1992) introduced the Ridge penalized log-likelihood defined as:

$$\log \mathcal{L}(\boldsymbol{\beta}) - \frac{\nu_R}{2}\, \boldsymbol{\beta}^T \widehat{\boldsymbol{\Sigma}} \boldsymbol{\beta}\,, \tag{2.1}$$

with $\widehat{\Sigma}$ the diagonal empirical variance matrix of $\widetilde{\mathbf{X}}$ and $\nu_R > 0$ the Ridge penalty parameter. Eilers et al. (2001) developed the Ridge IRLS (RIRLS) algorithm to optimize the criterion (2.1), where the weighted regression at each iteration is replaced by a Ridge weighted regression, hence:

$$\widehat{\boldsymbol{\beta}}^{(m+1)} = (\widetilde{\mathbf{X}}^T \mathbf{V}^{(m)} \widetilde{\mathbf{X}} + \nu_R\, \widehat{\boldsymbol{\Sigma}})^{-1} \widetilde{\mathbf{X}}^T \mathbf{V}^{(m)} \boldsymbol{\xi}^{(m)}\,.$$

A unique solution that maximizes the penalized loss (2.1) always exists and is computed as the limit of the sequence $(\widehat{\boldsymbol{\beta}}^{(m)})_{m\geq 1}$. Thus, the Ridge penalization regularizes the optimization of the logistic likelihood.

2.1.2 Adaptive sparse PLS on a continuous pseudo-response

The pseudo-response $\boldsymbol{\xi}^\infty$ produced by the RIRLS algorithm depends on the predictors through a linear model, and thus becomes suitable for the sparse PLS regression, following the approach of Fort & Lambert-Lacroix (2005) that uses standard PLS instead. In this heteroskedastic case, the ℓ_2 metric (in the observation space) is weighted by the empirical inverse covariance matrix $\mathbf{V}^\infty$, to account for the heteroskedasticity of the noise. In order to neglect the intercept in the SPLS step, we now consider the centered version of $\mathbf{X}$ and $\boldsymbol{\xi}^\infty$, denoted by $\mathbf{X}_c$ and $\boldsymbol{\xi}_c^\infty$, regarding the metric weighted by $\mathbf{V}^\infty$. The intercept β_0 will be estimated after the dimension reduction.

Adaptive sparse PLS regression

We propose to adjust the ℓ_1 constraint in the covariance maximization problem[2] to further penalize the less significant variables. This is expected to reduce the bias induced by the sparsity penalty and to produce a more accurate selection process, hence improving the compression. Such an approach is inspired by component wise penalization as adaptive Lasso (Zou, 2006) and to our knowledge has not been proposed for sparse PLS yet.

[2] $\text{argmin}_{\mathbf{w}\in\mathbb{R}^p} \Big\{ -\widehat{\text{Cov}}(\mathbf{X}\mathbf{w}, \boldsymbol{\xi}) + \nu_S\, |\mathbf{w}|_1 \Big\}$

We use the weights $\mathbf{w}_k^{\text{PLS}}$ from the classical PLS (without sparsity constraint) to adapt the ℓ_1 penalty constraint on the weight vector $\mathbf{w}_k^{\text{SPLS}}$. The penalized criterion considered by Chun & Keleş (2010) becomes:

$$\underset{\mathbf{a}\in\mathbb{R}^p,\,\mathbf{c}\in\mathbb{R}^p}{\text{argmin}} \left\{ -\tau\, \mathbf{a}^T\mathbf{M}^T\mathbf{M}\mathbf{a} + (1-\tau)(\mathbf{c}-\mathbf{a})^T\mathbf{M}^T\mathbf{M}(\mathbf{c}-\mathbf{a}) + \nu_S \sum_{j=1}^{p} \nu^j \, |c_j| \right\}, \qquad (2.2)$$

where $\mathbf{M} = (\mathbf{X}_c)^T\mathbf{V}^\infty\boldsymbol{\xi}_c^\infty$ is proportional to the weighted covariance matrix and $\nu^j = 1/|w_{jk}^{\text{PLS}}|$ accounts for the significance of the predictor j in the component k, higher weights in absolute values corresponding to more important variables.

The sparse weight vector $\mathbf{w}$ is given by the optimal $\mathbf{c}$. The closed-form solution is still based on the soft-thresholding operator[3] applied to the dominant singular vector of $\mathbf{M}$ but takes into account the adaptive penalty with a penalty term $\nu_S \times \nu^j$ for the j^{th} predictor. We call this method adaptive sparse PLS. It is here presented with a weighted matrix product to fit our heteroskedastic model, but it can be rewritten as classical sparse PLS by replacing $\mathbf{V}^\infty$ by the $n \times n$ identity matrix.

The active set of selected variables up to the component k is a subset of $\{1,\dots,p\}$, defined as the variables with a non null weight in at least one of the weight vectors $\mathbf{w}_1,\dots,\mathbf{w}_k$. It is denoted by:

$$\mathcal{A}_k = \cup_{\ell=1}^{k}\{j, w_{j\ell} \neq 0\}\,.$$

At the first step, the weight vector $\mathbf{w}_1$ is computed as the effective solution of the problem (2.2). At the step $k > 1$, $\mathbf{w}_k$ is computed by solving the adaptive covariance maximization problem with $\mathbf{M} = (\mathbf{X}_c)^T\mathbf{V}^\infty(\boldsymbol{\xi}_c^\infty)^{(k)}$, where $(\boldsymbol{\xi}^\infty)^{(k)}$ is a deflated pseudo-response. It is defined as the residuals in the PLS regression of the response $\boldsymbol{\xi}_c^\infty$ onto the selected variables in $\mathcal{A}_k$, following the algorithm from Chun & Keleş (2010).

Finally, the estimation $\widehat{\boldsymbol{\beta}}_{\backslash 0}^{\text{SPLS}}$ of $\boldsymbol{\beta}_{\backslash 0}$ in the model $\boldsymbol{\xi}_c^\infty = \mathbf{X}_c\boldsymbol{\beta}_{\backslash 0} + \boldsymbol{\varepsilon}$ is given by the PLS regression of $\boldsymbol{\xi}_c^\infty$ onto the selected variables in the active set $\mathcal{A}_K$. The coefficient $\widehat{\beta}_j^{\text{SPLS}}$ is set to zero if the predictor $j \in \{1,\dots,p\}$ is not in the active set. Indeed, following the definition of the sparse PLS regression, the sparse structure of the weight vectors $(\mathbf{w}_k)_{k=1:K}$ directly induces the sparse structure of $\widehat{\boldsymbol{\beta}}_{\backslash 0}^{\text{SPLS}}$. The variables selected to construct the new components $(\mathbf{t}_k)_{k=1:K}$ are the ones that contribute the most to the response, corresponding to those with non-null entries in the true vector $\boldsymbol{\beta}_{\backslash 0}$.

Eventually, the estimates $\widehat{\boldsymbol{\beta}}_{\backslash 0}^{\text{SPLS}}$ are renormalized to correspond to the non-centered and non-scaled data[4], i.e.

$$\widehat{\boldsymbol{\beta}}_{\backslash 0} = \widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\beta}}_{\backslash 0}^{\text{SPLS}}\,,$$

[3] c.f. Appendix Chapter B

[4] where $\widehat{\boldsymbol{\Sigma}}$ was previously defined as the diagonal empirical variance matrix of $\widetilde{\mathbf{X}}$.

where $\widehat{\boldsymbol{\beta}}_{\setminus 0}$ is the estimation of $\boldsymbol{\beta}_{\setminus 0}$ in the original logistic model $\mathbb{E}[Y_i] = \text{logit}^{-1}(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}_{\setminus 0})$. The intercept β_0 is estimated by $\widehat{\beta}_0 = \overline{\xi^\infty} - \overline{\mathbf{x}}\widehat{\boldsymbol{\beta}}_{\setminus 0}$ where $\overline{\xi^\infty}$ and $\overline{\mathbf{x}}$ are respectively the sample average of the pseudo-response and the sample average vector of predictors regarding the metric weighted by $\mathbf{V}^\infty$. Our method can be summarized as follow:

1. $(\boldsymbol{\xi}^\infty, \mathbf{V}^\infty) \leftarrow \text{RIRLS}(\mathbf{X}, \mathbf{y}, \nu_R)$
2. Center $\mathbf{X}$ and $\boldsymbol{\xi}^\infty$ regarding the scalar product weighted by $\mathbf{V}^\infty$
3. $\left(\widehat{\boldsymbol{\beta}}_{\setminus 0}^{\text{SPLS}}, \mathcal{A}_K, \mathbf{T}\right) \leftarrow \text{SPLS}(\mathbf{X}, \boldsymbol{\xi}^\infty, K, \nu_S, \mathbf{V}^\infty)$
4. Renormalization of $\widehat{\boldsymbol{\beta}} = \{\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}_{\setminus 0}\}$

The label $\widehat{y}_{\text{new}}$ of new observations $\mathbf{x}_{\text{new}} \in \mathbb{R}^p$ (non-centered and non-scaled) is predicted through the logit function thanks to the estimation $\widehat{\boldsymbol{\beta}} = \{\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}_{\setminus 0}\}$.

Our method estimates the predictor coefficients $\boldsymbol{\beta}$ in the logistic model by using the sparse PLS regression on a pseudo-response, considered as continuous and therefore in accordance with the theoretical framework of PLS. It completes compression and variable selection simultaneously. Our approach will be denoted by logit-SPLS in the following while the method by Fort & Lambert-Lacroix (2005) that inspired us will be denoted by logit-PLS.

2.1.3 Cross-validation versus stability selection

Calibration of the parameters by cross-validation

Choosing good values for the hyper-parameters of a statistical method is crucial. The quality of a model depends directly on these hyper-parameters. For instance, the sparsity parameter induces the degree of sparsity in the coefficient estimates. If wrongly set, the relative model would be a poor choice to fit the data. Such question of calibration needs to be cautiously treated.

Our approach depends on a sparsity penalty parameter $\nu_S > 0$, a Ridge penalty parameter $\nu_R > 0$ and the number of components $K \in \mathbb{N}$. A common procedure to choose these parameter values is cross-validation: for each possible value of hyper-parameters, learning the model on a sub-part of the training set, calculating the prediction error rate on the remaining observations, and taking the values that minimize it. To reduce the sampling dependence, we tune all the parameters by 10-fold cross-validation, meaning that we average the prediction error rate over 10 resamplings of the train set with respective sizes of 90%/10% of observations in sample for learning and testing (Boulesteix, 2004).

Stability selection to avoid parameter calibration

We also propose an alternative to cross-validation in order to evaluate the selection accuracy of the different approaches, based on the concept of stability selection developed by Meinshausen & Bühlmann (2010). In the context of methods combining compression and variable selections, applying such framework is innovative. It has been used in an unsupervised context by Sill et al. (2015), however, it has not been yet applied in the context of supervised compression methods for classification. We first introduce some notations to explain the principle of this approach.

The grid of all hyper-parameter values is composed of the sparse parameter ν_S, the Ridge parameter ν_R and the number of components K (depending on the methods). It is denoted by Λ. The concept of stability selection is the following. When considering a statistical method that fits a model, instead of choosing the best point λ in the hyper-parameter grid Λ that defines the best model, the procedure fits the model for all the point $\lambda \in \Lambda$, which is done anyway in the cross-validation step, and selects the variables depending on the majority vote among all the models.

More explicitly, the model is learned on resamplings of size $n/2$ (n being the sample size, here 294) for all points $\lambda \in \Lambda$. Then, depending on λ, the probability π_j^λ for the covariate j to be selected is estimated (on the resamplings). Indeed, π_j^λ is the probability for the covariate j to be in the set:

$$\widehat{S}_\lambda = \left\{ j \;:\; \widehat{\beta}_j(\lambda) \neq 0 \right\},$$

where $\widehat{\beta}_j(\lambda)$ is the corresponding coefficient estimated by the considered method for the hyper-parameter value λ.

Eventually, the variables that are selected by more than a certain proportion of models are defined as stable selected variables. Formally, the set of stable selected variables is defined as

$$\widehat{S}_{\text{stable}} = \left\{ j \;:\; \max_{\lambda \in \Lambda} \{\widehat{\pi}_j^\lambda\} \geq \pi_{\text{thr}} \right\},$$

where $\widehat{\pi}_j^\lambda$ is the estimation of π_j^λ over the resamplings and π_{thr} is a threshold value, meaning that the covariates with high selection frequency over the grid λ are kept and the covariates with low selection frequency are disregarded.

The average number of selected variables over the entire grid Λ is denoted by q_Λ, and defined as $q_\Lambda = \mathbb{E}\left[\#\{\cup_{\lambda \in \Lambda} \widehat{S}_\lambda\}\right]$. Meinshausen & Bühlmann (2010) provided a bound on the expected number of wrongly stable selected variables, i.e. the expected number of false positives in $\widehat{S}_{\text{stable}}$. This bound only depends on the threshold π_{thr}, the expectation q_Λ and the number p of covariates, it is defined as:

$$\mathbb{E}[\text{FP}] \leq \frac{1}{2\pi_{\text{thr}} - 1} \frac{q_\Lambda^2}{p} , \tag{2.3}$$

where FP is the number of false positives i.e. $\text{FP} = \#\{S_0^c \cap \widehat{S}_{\text{stable}}\}$, S_0 being the unknown set of true relevant variables and S_0^c its complementary. This results is derived for $\Lambda \subset \mathbb{R}^+$ under two conditions: 1) assuming that the indicators $(\mathbf{1}_{\{j \in \widehat{S}_\lambda\}})_{j \in S_0^c}$ are exchangeable for any $\lambda \in \Lambda$; 2) the original procedure of selection is not worst than random guessing. The first assumption assumes that the considered method does not "prefer" to select some covariates rather than some other in the set of the non-pertinent predictors. This hypothesis seems reasonable in our sparse PLS framework. The second one is verified according to the results on our simulation (c.f. previous section). Moreover, in the methods that we consider, the grid of hyper-parameters lies in $(\mathbb{R}^+)^2$ or $(\mathbb{R}^+)^3$, however the parameter that truly influences the sparsity of the estimation is the parameter $\nu_S \in \mathbb{R}^+$. Therefore, the sparse PLS appears to be a reasonable framework to apply the concept of stability selection.

Equation (2.3) determines how wide should be the parameter grid Λ in order to control the number of false positives (corresponding to a weak ℓ_1 penalization). In our study, the grid Λ is restrained so that $q_\Lambda = \sqrt{(2\pi_{\text{thr}} - 1)p \times \rho_{\text{error}}}$ leading to $\mathbb{E}[\text{FP}] \leq \rho_{\text{error}}$, where ρ_{error} is the maximum number of false positives in the stable selected variable set $\widehat{S}_{\text{stable}}$. For instance, when the threshold probability π_{thr} is set to 0.9, Λ is defined as a subset of the parameter grid, so that $q_\Lambda = \sqrt{0.8\, p\, \rho_{\text{error}}}$. In practice, q_Λ is unknown, but estimated by the empirical average number of selected variables over all $\lambda \in \Lambda$. In this context, the expected number of false positives will be lower than ρ_{error}.

This control on the averaged number of false positives is very useful when dealing with experimental data, in which the variables that should be selected are unknown. Thus, it can be viewed as a quality control of the selection process.

The procedure based on stability selection will be used in the analysis of experimental breast cancer data.

2.2 Simulation study

Due to the absence of theoretical results concerning sparse PLS, simulations appear to be necessary to assess the performance of our method. Thus, we run our approach and compare it to others on simulated data. The purpose is to control the model design to evaluate in which data configuration compression and selection are appropriate for classification. We assess whether our approach performs better or worse than previously proposed procedures. We also aimed at verifying if our method respects the two crucial questions about convergence and suitability for prediction and selection.

2.2.1 Design of the experiment

Performance evaluation

In order to assess the performance of our method, we compare it to other state-of-the-art approaches taking into account sparsity and/or performing compression. We eventually use a "reference" method, called GLMNET (Friedman et al., 2010), that performs variable selection, by solving the GLM likelihood maximization penalized by ℓ_1 norm penalty for selection and ℓ_2 norm penalty for regularization, also known as the Elastic Net approach (Zou & Hastie, 2005). In particular, GLMNET is supposed to be appropriate to handle correlated covariates. Computations were performed using the software environment for statistics `R`. The GPLS approach used in our computation comes from the archive of the former `R`-package `gpls`, the methods logit-PLS and PLS-DA from the package `plsgenomics`, SGPLS, SPLS-log and SPLS-DA from the `R`-package `spls`, GLMNET from the `R`-package `glmnet`. The hyper-parameters of the different approaches are tuned by using the cross-validation procedures supplied in each package and by using the range of hyper-parameters recommended by their respective authors.

Block design and logit model

Our simulated data are constructed to assess the interest of compression and variable selection. The simulations are inspired from Zou et al. (2006), Shen & Huang (2008) or Chung & Keleş (2010). The purpose is to control the redundancy within predictors, meaning the degree of multicollinearity, and the relevance of each predictor to explain the response, meaning the degree of sparsity in the model.

We consider a design matrix $\mathbf{X}$ of dimension $n \times p$, with $n = 100$ fixed, and $p = 100, 500, 1000, 2000$, so that we examine different high dimensional models. To simulate redundancy within predictors, $\mathbf{X}$ is partitioned into k^* blocks (10 or 50 in practice) denoted by $\mathcal{G}_k$ for block k. Then for each j in the block $\mathcal{G}_k$, X_{ij} is generated depending on a latent variable H_k as $X_{ij} = H_{ik} + F_{ij}$, with $H_{ik} \sim \mathcal{N}(0, \sigma_H^2)$ and some noise $F_{ij} \sim \mathcal{N}(0, \sigma_F^2)$. The covariate matrix is therefore defined as:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots\cdots & x_{1j} & \cdots\cdots & \cdots\cdots & \cdots & x_{1p} \\ \vdots & \vdots & & \vdots & & & & \vdots \\ x_{i1} & x_{i2} & & x_{ij} & & & & x_{ip} \\ \vdots & \vdots & & \vdots & & & & \vdots \\ x_{n1} & x_{n2} & \cdots\cdots & x_{nj} & \cdots\cdots & \cdots\cdots & \cdots & x_{np} \end{bmatrix}$$
$$\underbrace{\hphantom{x_{n1}\ x_{n2}\ \cdots}}_{\mathcal{G}_1}\ \underbrace{\hphantom{\cdots\ x_{nj}\ \cdots}}_{\mathcal{G}_2}\ \cdots\ \underbrace{\hphantom{\cdots\ \cdots\ x_{np}}}_{\mathcal{G}_{k^*}}$$

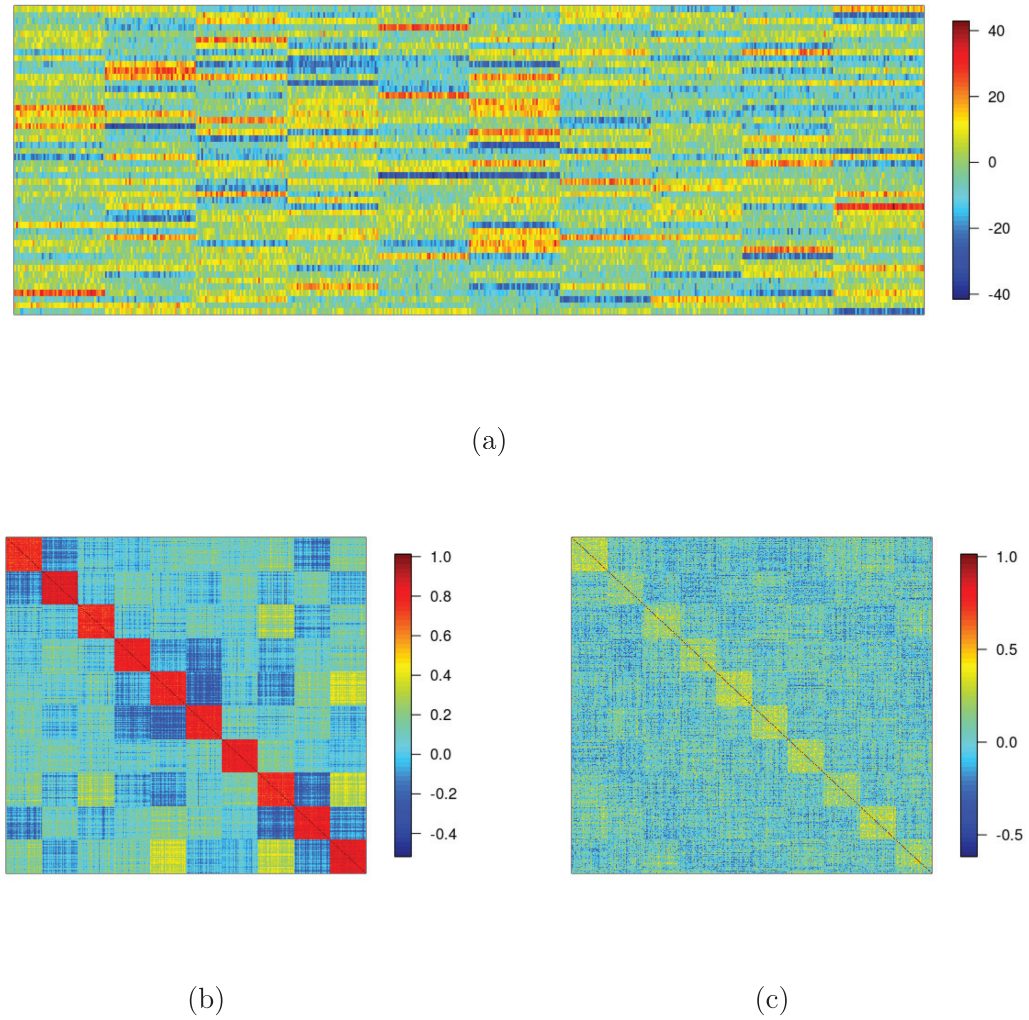
For example, Figure 2.1a shows a heatmap of the matrix $\mathbf{X}_{100\times10000}$ generated from $k^* = 10$ blocks. In this framework, each H_k is introduced to control the correlation within the block $\mathcal{G}_k$ which is proportional to the ratio σ_H^2/σ_F^2. The correlation between the blocks is regulated by σ_H^2, the higher σ_H^2 the less dependency. Figure 2.1b and Figure 2.1c show the heatmap of the correlation between the p covariates depending on different values of the ratio σ_H/σ_F. Here, $k^* = 15$, the blocks on the diagonal of the correlation matrix represent the groups $\mathcal{G}_k$ where the correlation between covariates is higher. In the following we consider $\sigma_H/\sigma_F = 2$ or $1/3$.

The true vector of predictor coefficients $\boldsymbol{\beta}^*$ is structured according to the blocks of $\mathbf{X}$. Actually, ℓ^* blocks in $\boldsymbol{\beta}^*$ are randomly chosen among the k^* ones to be associated with non null coefficients (with $\ell^* = 1$ or $k^*/2$), e.g.

$$\boldsymbol{\beta}^* = \Big(\underbrace{b, b, \ldots,}_{\mathcal{G}_1}\ \underbrace{0, \ldots, 0,}_{\mathcal{G}_2}\ \underbrace{b, \ldots, b,}_{\mathcal{G}_3} \ldots, \underbrace{0, \ldots, 0}_{\mathcal{G}_{k^*}}\Big)^T,$$

with $b \neq 0$. In practice, all the coefficients within the ℓ^* designated blocks are constant (with value b=1/2). In our model, the relevant predictors contributing to the response will be those with non zero coefficient, and our purpose will be to retrieve them. The response variable Y_i is sampled as a Bernoulli variable, with parameter π_i that follows a logistic model: $\pi_i = \text{logit}^{-1}(\mathbf{x}_i^T \boldsymbol{\beta}^*)$.

The parameter values that are tuned by cross-validation are the following: the number of components K varies from 1 to 8, the Ridge parameter ν_R in RIRLS are 31 points that are $\log_{10}$-linearly spaced in the range $[10^{-2}; 10^3]$, the sparse parameter ν_S for all sparse PLS approach are 10 points that are linearly spaced in the range $[0.05; 0.95]$.



(a)

(b)

(c)

Figure 2.1 – (a) Heatmap of a data matrix $\mathbf{X}_{n\times p}$ generated from $k^* = 10$ latent variables, with $n = 100$ and $p = 10000$. (b) Heatmap of the correlation between the covariates from a data matrix $\mathbf{X}$ ($n = 100$, $p = 1000$ and $k^* = 15$) when $\sigma_H > \sigma_F$, i.e. the latent structure is stronger than the noise. (c) Correlation between the covariates from a data matrix $\mathbf{X}$ ($n = 100$, $p = 1000$ and $k^* = 15$) when $\sigma_H < \sigma_F$, i.e. the noise overtakes the latent structure.

2.2.2 Guarantee about convergence and stability

Ridge penalty ensures convergence

Convergence is an important issue associated with the IRLS algorithm when estimating GLM parameters. It is especially a crucial issue when combining PLS and IRLS algorithm as pointed out by Fort & Lambert-Lacroix (2005). With the analysis of high dimensional data and the use of selection in the estimating process, it becomes even more essential to ensure the convergence of the optimization algorithm. As we will see, convergence issues lead to unreliable results. In particular, non-convergent algorithms will appear unstable. Thy will induce instability in the prediction and in the variable selection. The link between convergence and stability will be highlighted in the following. In order to check the convergence of the different algorithms, we consider the ℓ_2 convergence criterion between two iterations: $\|\widehat{\boldsymbol{\beta}}^{(m+1)} - \widehat{\boldsymbol{\beta}}^{(m)}\|_2$. In the following, the algorithm is assumed to converge if the ℓ_2 norm gap becomes lower than 10^{-12} with a maximum number of a hundred iterations in order to limit computation time.

Our simulations show that Ridge regularization systematically ensures convergence of the IRLS algorithm before performing sparse PLS in our method (logit-SPLS), whatever the configuration of simulation: $p = n$, $p > n$, high or low sparsity, high or low redundancy (see Table 2.1 for an example). On the contrary, approaches that use (sparse) PLS before or within the IRLS algorithm (resp. SPLS-log and (S)GPLS) do not converge quite often in some configurations (Table 2.1). To illustrate these convergence issues we studied the convergence path of $\|\widehat{\boldsymbol{\beta}}^{(m+1)} - \widehat{\boldsymbol{\beta}}^{(m)}\|_2$ which reveals that our method converges within fifteen iterations on average whereas other methods do not often converge, and even encounter cyclic singularities. For instance, Figure 2.2 shows different trajectories of the criterion $\|\widehat{\boldsymbol{\beta}}^{(m+1)} - \widehat{\boldsymbol{\beta}}^{(m)}\|_2$ when running the algorithm logit-SPLS and sparse Generalized PLS (SGPLS) on simulated data where $n = 50$, $p = 500$, $k^* = 10$, $\ell^* = 1$ and $\sigma_H/\sigma_F = 2$.

This point confirms that performing (sparse) PLS before or within the IRLS algorithm does not avoid convergence issues. On the contrary, it confirms the interest of the Ridge regularization to ensure the convergence of the IRLS algorithm. Moreover, this convergence seems to be fast, which depicts an interesting outcome for computational time.

Method	$p = 100$	$p = 500$	$p = 1000$	$p = 2000$
gpls	52	38	40	38
sgpls	68	72	72	68
spls-log	98	42	20	6
logit-spls	**100**	**100**	**100**	**100**

Table 2.1 – Percentage of model fitting that converged over 75 simulations for different values of p, when $\sigma_H/\sigma_F = 2$, $\ell^* = 1$ and $k^* = 50$

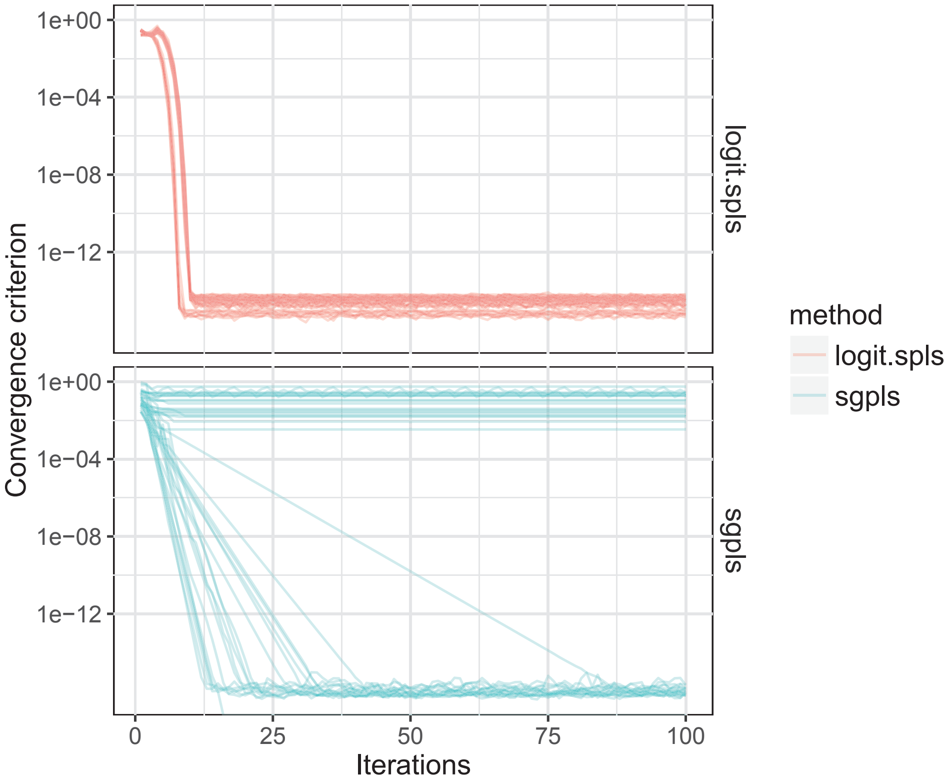


Figure 2.2 – Trajectories of the convergence criterion in 40 different runs of the algorithms logit-SPLS and SGPLS on simulated data where $n = 50$, $p = 500$, k^*10, $\ell^* = 1$ and $\sigma_H/\sigma_F = 2$. Each algorithm ran 3000 iterations, only the first 100 hundred are shown (the trajectories that do not converge after 100 iterations are still non convergent after 3000).

Adaptive selection improves cross-validation stability

When choosing the hyper-parameter values for the different methods considered here, one can expect a certain stability, meaning that when running a procedure many times on a same sample, the cross-validation process is supposed to return the same values for parameters. Otherwise, the label prediction and the variable selection become almost uncertain, hence not suitable for experiment reproducibility. For each configuration of our simulated data, we consider the precision of the sparse hyper-parameter values returned by cross-validation, i.e. the inverse of its standard deviation over repetitions of tuning procedure (the higher precision, the less variability). This scheme shows (Figure 2.3) that our adaptive method is more stable than other sparse PLS approaches, meaning that the cross-validation procedure almost always chooses the same sparse parameter λ_S values for a given sample (i.e. smaller standard deviation over repetition). It appears that the choice of components number K and Ridge parameter ν_R are also very stable (Figure not shown). On the contrary, cross-validation for methods such as SPLS-log or SGPLS is more unstable, returning different values, depending on the run.

On the one hand, the cross-validation stability can be linked to the consideration of convergence. It appears that the procedures (SGPLS, SPLS-log) which do not converge on our simulations present a higher cross-validation instability, whereas our method (logit-SPLS) converges efficiently and shows a better cross-validation stability. On the other hand, the variable selection accuracy defined as the proportion of rightly selected and rightly non-selected variables (Chong & Jun, 2005) is also influenced by the cross-validation stability, as the accuracy precision (inverse of the standard deviation over 75 repetitions) increases with the cross-validation stability (c.f. Figure 2.3).

Another interesting point is that the cross-validation procedure almost always returns an optimal number of components K equal to 1. In order to reduce the computation time, we fixed the number of components to one in our following simulation and performed the tuning only on the sparsity parameter ν_S and the Ridge parameter ν_R. Nonetheless tuning a supplementary parameter does not account for a bigger time in execution since our method converges fast, whereas the other ones do not converge hence iterating farther (until the limit on the number of iterations, fixed in the implementation).

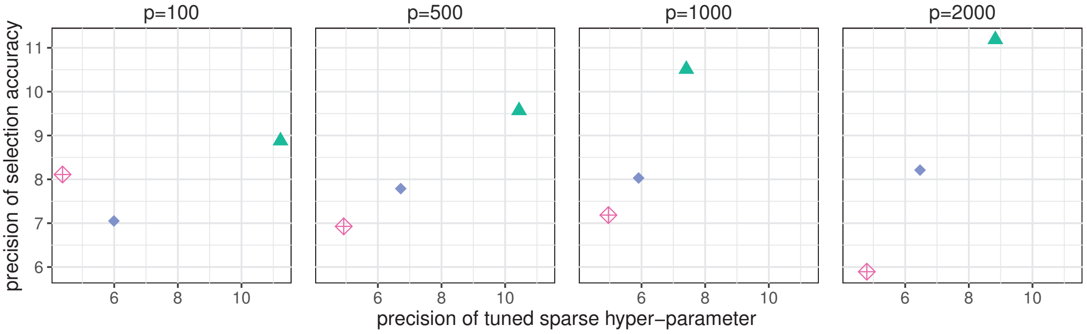


Figure 2.3 – Precision is the inverse of standard deviation (the higher precision, the less variability). Precision on the sparse parameter values, i.e. ν_S, chosen by cross-validation versus precision on selection accuracy over 75 simulations for different number p of predictors (from 100 to 2000). Methods: logit-spls-adapt (▲), sgpls (◆), spls-log (⊕).

2.2.3 Performance in prediction and selection

Selection increases prediction accuracy

To study the importance of combining compression and variable selection, we now focus on the prediction accuracy defined as the rate of correct classification, evaluated through the prediction error rate. Thus we compete sparse PLS approaches and their PLS (non sparse) matching: our method logit-SPLS versus the logit-PLS from Fort & Lambert-Lacroix (2005), with sparse or non Partial Least Squares after the IRLS algorithm, and others that perform (sparse) PLS within IRLS loop (GPLS vs SGPLS), or before a discriminant analysis (PLS-DA vs SPLS-DA); so that we assess the impact of selection for different methods of compression. In every configuration of simulations (see Table 2.2 for example when $p = 2000$), the prediction performance of compression methods is stable or increased by the addition of a selection step, meaning that in any case compression and selection should be considered for prediction. However methods that are not converging or not suitable for qualitative response (SGPLS, SPLS-DA, SPLS-log) achieve the same prediction performance than converging and suitable ones (GLMNET, our logit-SPLS). This indicates that checking prediction accuracy only may not be a sufficient criterion to assess the relevance of a method. The GPLS method is however a good example of non-convergent method (c.f. Table 2.1) that presents a high variability and poor performance regarding prediction.

Actually, the combination of Ridge IRLS and sparse PLS in our method ensures convergence and provides good prediction performance (prediction error rate at 10% on average) even in the most difficult configurations $n = 100$ and $p = 2000$, which makes it

Method	Av. prediction error	(Stand. dev.)
gpls	0.47	(0.32)
pls-da	0.20	(0.09)
logit-pls	0.17	(0.10)
glmnet	0.13	(0.12)
logit-spls	**0.10**	**(0.10)**
sgpls	0.10	(0.11)
spls-da	0.12	(0.12)
spls-log	0.11	(0.12)

Table 2.2 – Prediction error rate averaged over all simulation configurations when $p = 2000$ (and standard deviation), for approaches using sparsity principle or not (delimited by the line). The resulting variance is not too large except for GPLS which also presents the worst performance. Our approach logit-SPLS is as good or better than any other methods

an appropriate framework for classification.

Compression increases selection accuracy

The prediction performance are nevertheless not very useful if the selected variables do not match with the genuine important predictors to explain the response. To assess the selection accuracy, we compare the pool of selected predictors returned by sparse methods (performing compression or not) to the set of relevant ones used to construct the response, i.e. with a non zero coefficient β_j^* in our model. We thus evaluate the effect of the compression step on variable selection. To determine if one method selects too many or not enough variables, we consider sensitivity and specificity (Chong & Jun, 2005), respectively proportion of true positive and true negative regarding correctly or wrongly selected variables, which illustrates under or over selection phenomenon. We especially focus on the true positive versus false positive rate, i.e. sensitivity versus "$1-$specificity", the first one is expected to be close to one, and the second one to be close to zero.

On our simulations (see Figure 2.4), especially when the number of covariates p grows, our method logit-SPLS selects less irrelevant predictors as the false positive rate is lower, compared to other sparse PLS approaches (SGPLS and SPLS-log). These two select

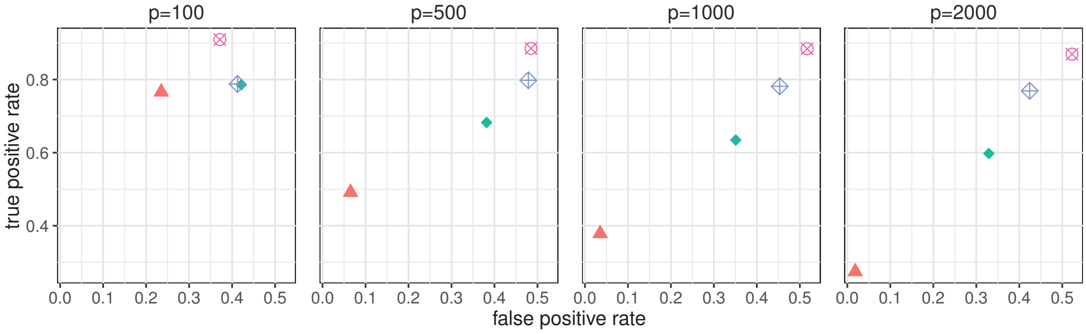


Figure 2.4 – True positive average rate (i.e. selection sensitivity, the higher the better) versus average false positive rate (i.e. 1-specificity, the lower the better) over 75 simulations, for different values of p from 100 to 2000 (average over all repetitions and configurations of simulations). Methods: glmnet (▲), logit-spls-adapt (◆), sgpls ($\oplus$), spls-log ($\otimes$). The trade-off between true positive rate and false positive rate is better with our method logit-SPLS as we select not too much variables (less false positives) compared to other sparse PLS approaches, but not too few variables (more true positives) compared to glmnet.

more true positives as their sensitivity is closer to one; however they tend to select too many variables (with their higher false positive rate), which is confirmed by a higher averaged number of selected variables for SGPLS and SPLS-log, and especially higher than the number of true relevant variables in our model (Figure not shown), defined as $\#\{j, \beta_j^* \neq 0\}$.

Our baseline is the GLMNET procedure, which processes selection without compression, conversely to sparse PLS approach. It shows a lower accuracy, and especially high specificity (low false positive rate) and very low sensitivity, meaning that it selects only few variables, avoiding false positives, but excluding to many pertinent variables. The relative good sensitivity of other sparse PLS approach (SGPLS and SPLS-log) is also balanced by a selection process less stable than our method as the accuracy standard deviation is higher (as previously mentioned, see Figure 2.3).

In any case, combining compression and variable selection has a true impact on selection accuracy, compared to sparse only approach such as GLMNET, which supports our idea of using sparse PLS over other methods.

2.3 Classification of breast tumors by sparse PLS for logistic regression

The problem that motivated the development of our dimension reduction approach for classification is the analysis of gene expression profiles. The questions raised in this context concerns for instance the characterization of disease as cancer at the molecular level. Our objective in this section is to assess the performance of our method in a life-sized classification situation. In particular, we consider the analysis of an experimental data set. We especially use our approach to predict the fate of cancer patients based on the expression of genes coming from a tumorous tissue.

We use a publicly available data set on breast cancer, published by Guedj et al. (2012). It contains the expression level of 54613 genes for 357 patients affected by breast cancer. The original work consisted in classifying breast tumors according to the expression of genes. We rather focus on the relapse after 5 years, considering a $\{0, 1\}$ valued response, if the relapse occurred or not. The design matrix $\mathbf{X}$ contains the gene expression levels for the corresponding patients. The outcome is already known for all patients. The experiment is designed as follows, the model is trained on a part of the data and used to predict the response on the rest of the data (we consider many resamplings to avoid over-fitting). Then, it is possible to compare the prediction with the true value of the response.

Data preprocessing

We restrict the analysis on 294 patients for whom the relapse situation is known. We also reduce the number of genes by taking away the less differentially expressed genes between the two conditions (relapse or not). To do so, we determine the p-values associated to the t-test on the expression of each gene for each condition, then correct these p-values with the method by Benjamini & Hochberg (1995) for multiple testing. Finally, the genes are ranked according to the p-values[5]. We take the 5000 most differentially expressed genes, corresponding to a confidence level of approximately 70% (not too sharp). The matrix of pre-selected gene expression is finally centered and scaled to avoid that the most differentially expressed genes (with higher variance) hide the effect of any other potential relevant gene.

[5] The more expressed genes correspond to the smaller p-values

Method	Conv. perc.	sparse param. precision
logit-spls-adapt	**100**	**6.44**
sgpls	5	5.63
spls-log	1	5.37

Table 2.3 – Convergence percentage versus precision on sparse parameter values chosen by cross-validation (i.e. inverse of standard deviation, the higher the less variability) when fitting the model over 100 resamplings with each method.

2.3.1 Convergence and accuracy in prediction

To assess the performance of the different approaches for prediction and dimension reduction, we applied the methods GLMNET, logit-PLS, logit-SPLS (adaptive or not), SGPLS and SPLS-log to our data set. Each method is trained and tested over 100 resamplings, where observations are randomly split into training and test sets with a 70%/30% ratio. On each resampling, the parameter values of each method are tuned by 10-fold cross-validation on the training set, respecting the following grid $K \in \{1, \dots, 8\}$, ν_R in RIRLS are 31 $\log_{10}$-linearly spaced points in the range $[10^{-2}; 10^3]$, the sparse parameter ν_S for all sparse PLS approach are 10 linearly spaced points in the range $[0.05; 0.95]$. The procedure from the package `glmnet` determines by itself the grid of hyper-parameters.

Convergence and stability with RIRLS and adaptive sparse PLS

As seen in the simulation experiment, the convergence of the different methods is an important issue. The IRLS algorithm regularized by Ridge (RIRLS) confirms its usual convergence (see Table 2.3). The other approaches that use sparse PLS within the IRLS iterations (SGPLS) or before logistic regression (SPLS-log) encounter severe issues and almost never converge. Following a similar pattern, our adaptive selection is far more stable under the tuning of the sparsity parameter ν_S by cross-validation than any other approach using sparse PLS (Table 2.3). Indeed, the precision[6] on the chosen ν_S (over the different runs) is the highest for our method. It illustrates the lower variability in the hyper-parameter tuning over repetitions.

[6] Inverse of the standard deviation.

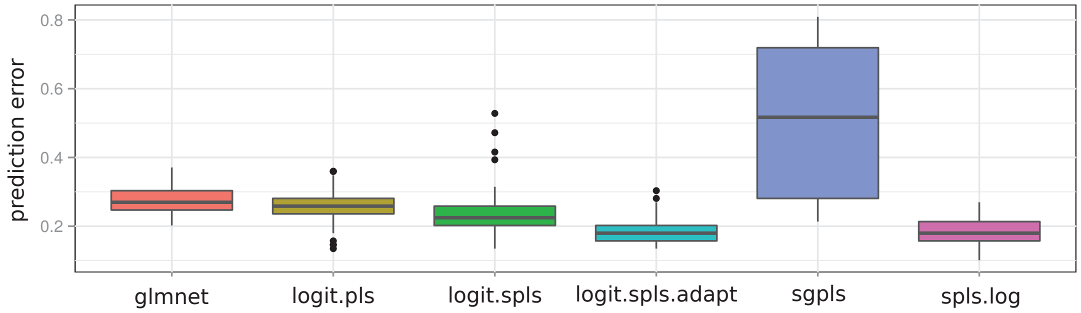


Figure 2.5 – Prediction error rate over 100 resamplings when tuning and fitting the model on a train subset of the data and predicting the outcome on a different test subset with each method (the ratio train/test is 70%/30% of the observations).

Adaptive selection increases prediction accuracy

Our approach logit-SPLS performed better for prediction (see Figure 2.5) than its predecessor logit-PLS without variable selection. This point confirms that the combination of variable selection and compression increases the prediction accuracy. Moreover, the adaptive version is even better and reaches an average prediction error rate below 20%. The SGPLS method does not confirm its performance on our simulation with poor and highly variable results. The instability induced by non-convergent methods is illustrated here[7]. A first striking point is that SPLS-log performs as well as our adaptive method. However this point will be counterbalanced in the following (lower performance regarding other criteria).

Compression is more efficient to discriminate the response

We now illustrate the interest of our method for data visualization. As in standard PLS, we represent the coordinates of the first two components constructed by compression methods, i.e. the observation scores. The points are colored according to their Y-labels. An efficient compression technique would separate the Y-classes with fewer components. We compare the logit-PLS, our logit-SPLS procedure, the SGPLS and the SPLS-log approaches, by tuning and fitting the model on different resamplings of our data set, the number of components is not tuned and fixed to $K = 2$. We use an unsupervised compression method, i.e. the Principal Component Analysis (PCA)[8], as a reference for

[7] In particular, this variability is not caused by a wrong set of hyper-parameter candidate values, since we used the same cross-validation procedure in the simulations.

[8] The principle of PCA is explained in Chapter 4.

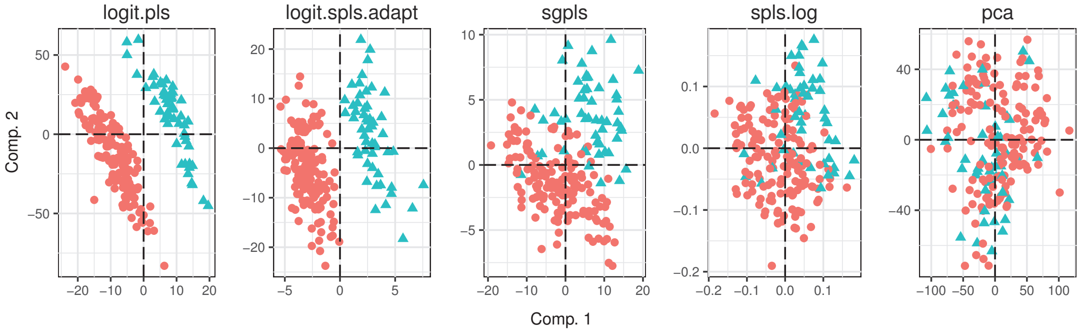


Figure 2.6 – Individual scores of each observation on the first two components for the different methods. The points are shaped according to the value of the response: 0 (●) and 1 (▲). Our approach logit-SPLS and the non sparse related logit-PLS discriminate the two classes with the first two components whereas other methods do not, including the PCA. The scale of the components depends on the different normalization in the output of each methods and are therefore not comparable.

compression and data visualization. Figure 2.6 represents the first two components computed by each methods when fitting a single model. It appears that the first component produced by our method (logit-SPLS) discriminates the observations between the two conditions. This is particularly consistent with the fact that the tuning procedure always chooses $K = 1$ (as previously mentioned). The first two components from the corresponding non sparse approach (logit-PLS) are sufficient to easily separate the two Y-classes. However, the other methods combining sparse PLS and logistic regression differently (SGPLS and SPLS-log) do not achieve a similar efficiency in the compression process. The first two components does not separate the Y-labels, indicating that these two methods need more components to discriminate properly the Y-classes, leading to a less efficient compression process. Therefore, our method turns out to be very effective for data visualization, even compared to principal component analysis, whose first two components explain less than 30% of the total variability in $\mathbf{X}$ and do not discriminate the two classes (c.f. Figure 2.6).

2.3.2 Calibration by stability selection

In order to evaluate the selection accuracy of the different approaches in our experimental case, we use the concept of stability selection as introduced in Section 2.1.3. The interest here is to avoid choosing a value for the hyper-parameters. The different model are learned for all hyper-parameters in a restrained grid of values so that the averaged number of false positives among selected variables is controlled.

The stability selection analysis (see Figure 2.7) shows that, when the averaged number of false positives is fixed, our approach logit-SPLS selects more genes than any other approach (SGPLS, SPLS-log and GLMNET). This means that, on average, our method discovers more true positives (because the number of false positives is bounded at the same level), hence unraveling more relevant genes than other approaches. This again illustrates the good performance in selection of our procedure. More generally, approaches that use sparse PLS, i.e. performing selection and compression, select more variables than GLMNET to reach the same number of false positives, hence retrieving more true positives than GLMNET which performs only selection. This supports our idea that combining compression and selection is very suitable for high dimensional data analysis.

2.4 Sparse PLS for multi-group classification

In the two previous sections, we proposed a computational framework for the logistic regression by using sparse PLS. After highlighting the interest of our method for the binary case, we consider the generalization of the logit-SPLS algorithm for the classification of multi-categorical data. We first present the model and the estimation procedure, then we present an on-going data analysis that motivated the development of this approach.

2.4.1 Multinomial sparse PLS

Thanks to the GLM framework, our procedure for logistic regression can be generalized to handle a multi-categorical response. This problem is known as multinomial logistic regression or polytomous regression (McCullagh & Nelder, 1989). In this section, we present the extension of our algorithm, that we called multinomial sparse PLS.

Multinomial logistic regression

The model for multinomial logistic regression is the following (Fahrmeir & Tutz, 2001). The response y_i takes its values in a discrete set $\{0, \dots, G\}$ corresponding to $G + 1$ groups or classes of observations. The associated variable Y_i ($i = 1, \dots, n$) follows a multi-categorical distribution where $\mathbb{P}(Y_i = g \,|\, \mathbf{x}_i) = \pi_{ig}$ for any class g. Based on a direct generalization of the logistic model, a class of references is set (generally the class 0) and for each class $g \neq 0$, the probability π_{ig} that $Y_i = g$ depends on a linear

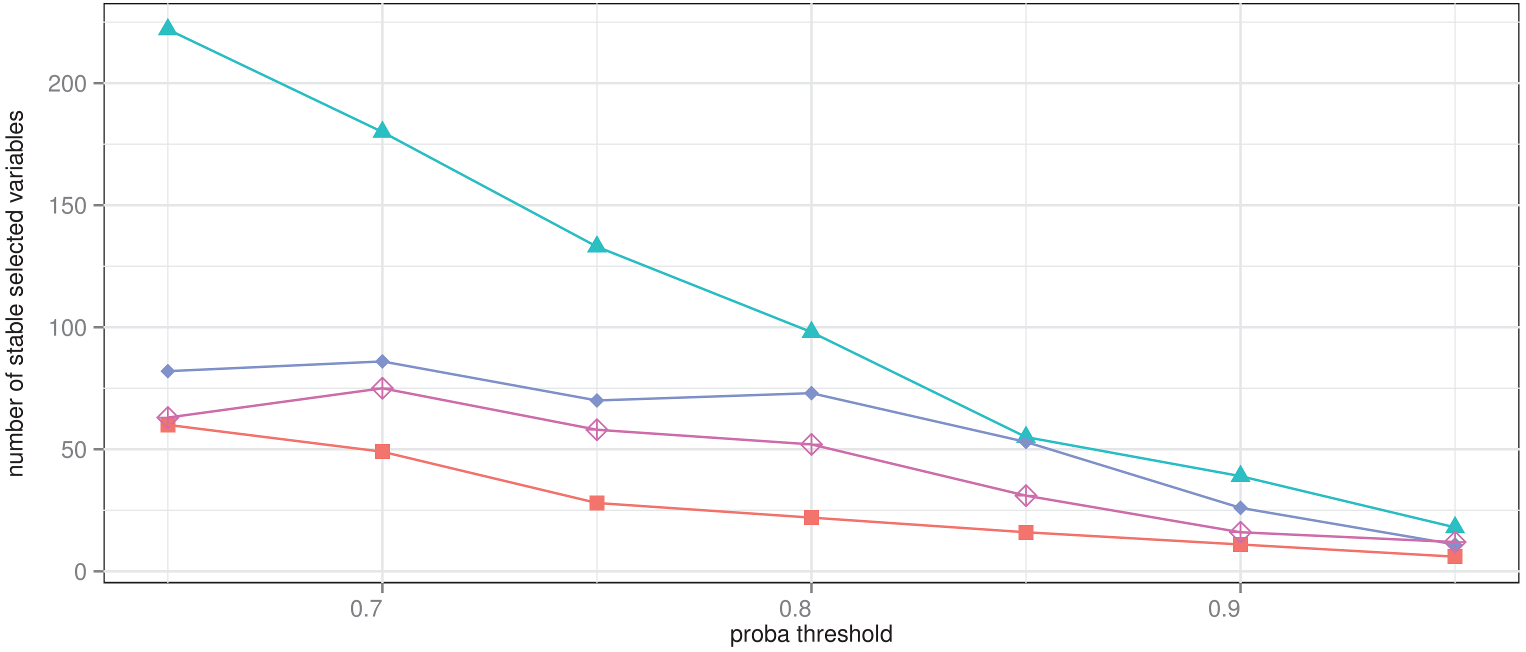


Figure 2.7 – Number of variables in the set of stable selected variables versus the threshold π_{thr}, when forcing the average number of false positives to be smaller than $\rho_{\mathrm{error}} = 10$. Methods: glmnet (−■−), logit-spls-adapt (−▲−), sgpls (− ◆ −), spls-log (− ⊕ −).

combination of predictor such as:

$$\log\left(\frac{\pi_{ig}}{\pi_{i0}}\right) = \widetilde{\mathbf{x}}_i^T \boldsymbol{\beta}_g, \tag{2.4}$$

with a specific vector of coefficient $\boldsymbol{\beta}_g \in \mathbb{R}^{p+1}$ for each class $g = 1, \ldots, G$. Indeed, the probabilities $(\pi_{ig})_{g=1:G}$ determine the probability π_{i0} since $\sum_{g=0}^{G} \pi_{ig} = 1$. As introduced in Chapter 1, a column of 1 is added in the matrix $\widetilde{\mathbf{X}}$ to incorporate the intercept in the linear combination $\widetilde{\mathbf{x}}_i^T \boldsymbol{\beta}_g$. The log-likelihood can be explicitly formulated:

$$\log \mathcal{L}\big((\boldsymbol{\beta}_g)_{g=1:G}\big) = \sum_{i=1}^{n} \left\{ \sum_{g=1}^{G} y_{ig}\, \widetilde{\mathbf{x}}_i^T \boldsymbol{\beta}_g - \log\Big(1 + \textstyle\sum_{g=1}^{G} \exp(\widetilde{\mathbf{x}}_i^T \boldsymbol{\beta}_g)\Big) \right\}, \tag{2.5}$$

where the binary variable $y_{ig} = \mathbf{1}_{\{y_i = g\}}$ indicates[9] the class of the observation i.

It is possible to rearrange the data in order to formulate a vectorized version of the loss (2.5), and express the multinomial logistic regression as a logistic regression of a binary response $\mathcal{Y} \in \{0,1\}^{n\,G}$ against a matrix of rearranged covariates $\widetilde{\mathcal{X}} \in \mathbb{R}^{n\,G \times (p+1)\,G}$. The response vector $\mathcal{Y}$ of length $n\,G$ is defined as follows:

$$\mathcal{Y} = \Big((y_{1g})_{g=1:G}, (y_{2g})_{g=1:G}, \ldots, (y_{ig})_{g=1:G}, \ldots, (y_{ng})_{g=1:G}\Big)^T,$$

where $y_{ig} = \mathbf{1}_{\{y_i = g\}}$ as previously mentioned. The new covariate matrix $\widetilde{\mathcal{X}}$ of dimension $n\,G \times (p+1)\,G$ is defined by blocks as:

$$\widetilde{\mathcal{X}} = \begin{bmatrix} \widetilde{\mathcal{X}}_1 \\ \vdots \\ \widetilde{\mathcal{X}}_i \\ \vdots \\ \widetilde{\mathcal{X}}_n \end{bmatrix},$$

where each block i is constructed by G diagonal repetitions of the row $\widetilde{\mathbf{x}}_i$ from the original covariate matrix $\widetilde{\mathbf{X}}$, i.e.

$$\widetilde{\mathcal{X}}_i = \left.\begin{pmatrix} 1\ x_{i1}\ \ldots\ x_{ip} & & 0 \\ & \ddots & \\ 0 & & 1\ x_{i1}\ \ldots\ x_{ip} \end{pmatrix}\right\} G \text{ repeats of the row } \mathbf{x}_i^T,$$

[9] $\mathbf{1}_{\{\mathcal{A}\}}$ is the indicator function valued in $\{0,1\}$, indicating if the statement $\mathcal{A}$ is true (1) or false (0).

The coefficient vectors $\boldsymbol{\beta}_g \in \mathbb{R}^{p+1}$ (for $g = 1, \ldots, G$) are also reorganized in the vector $\mathbf{B} \in \mathbb{R}^{(p+1)\,G}$ as:

$$\mathbf{B} = \Big((\beta_{0g})_{g=1:G}, (\beta_{1g})_{g=1:G}, \ldots, (\beta_{jg})_{g=1:G}, \ldots, (\beta_{pg})_{g=1:G} \Big)^T,$$

where $(\beta_{jg})_{j=0:p}$ are the coordinates of $\boldsymbol{\beta}_g$, so that the response $\mathcal{Y}$ depends on the linear combination $\widetilde{\mathcal{X}}\,\mathbf{B}$.

Thanks to this reformulation, it is possible to adapt the IRLS algorithm to estimate the coefficients $\mathbf{B}$ and infer the probabilities π_{ig} that observations y_i belongs to the class g. The algorithm that we call Multinomial IRLS (mIRLS) is detailed in Fort et al. (2005) and was also regularized by a Ridge penalty to avoid optimization issues.

Dimension reduction with sparse PLS

The vectorized formulation of the multinomial logistic regression allows to use the dimension reduction approach introduced at the beginning of this chapter. As in the binary case, the Multinomial IRLS algorithm (penalized by Ridge) produces a continuous pseudo-response (at the convergence) that is suitable for the sparse PLS regression. Thus, our approach, called multinomial-SPLS, extends directly our algorithm logit-SPLS to the multinomial logistic regression. It estimates the linear coefficients $\mathbf{B}$ by sparse PLS, processing compression and variable selection simultaneously. Then, these estimated coefficients are used to get an estimation of the probabilities π_{ig}. Our procedure is directly inspired from the approach by Fort et al. (2005) that extended the algorithm logit-PLS (Fort & Lambert-Lacroix, 2005) to the multi-categorical cases.

It can be noted that Ding & Gentleman (2005) presented a version of the Generalized PLS suitable for multinomial logistic regression, i.e. the linear regression inside the iteration of the mIRLS algorithm are processed by weighted PLS regression. Chung & Keleş (2010) introduced a similar algorithm based on sparse PLS (extension of the sparse Generalized PLS algorithm). However, we used exclusively our multinomial-SPLS algorithm in the data analysis. Indeed, based on the conclusions from the binary case, our approach showed better results regarding prediction performance on an experimental data set. Moreover, the dimension of the data is drastically increased because of the rearrangement since the number of observations becomes $n\,G$ and the number of covariates becomes $p\,G$. It is therefore necessary to account for the computational cost and to give priority to computationally efficient methods. In particular, thanks to the Ridge penalty, we showed that our approach converges quickly, hence reducing the time of computation.

2.4.2 Characterization of T lymphocyte types

Single-cell data

The question of multi-group classification was motivated by a collaboration with Jeff Mold, a Biologist from the *Karolinska Institutet* (Stockholm, Sweden). The whole project aims at studying, at the single-cell level (Stegle et al., 2015; Gawad et al., 2016), the transcriptomic response of a population of lymphocyte T cells to a vaccine shot. Our team is in charge of the statistical analysis of the data. Here we will focus on a specific question that we had to address during the project, concerning the classification of single cells into different cell types. Indeed, as part of the immune system, the T lymphocyte can be grouped into different categories, depending on their function.

Like the B cells, the T cells are part of the adaptive immune system. However, the mechanisms of an efficient T cells response are still largely unknown. Therefore, the fine understanding of this adaptive immune response is of great interest for the creation of new vaccines. After a vaccine shot, we expect the formation of two categories of T cells: the "effector" T cells carrying the immune response against the pathogen and the long-lasting "memory" T cells that will constitute a repertoire for later secondary immune responses. In the literature, T cells are described as 4 sub-groups : CM, TSCM ("Memory"), TEMRA, EM, ("Effector") based on the measurement of two surface markers[10]. However, those 4 sub-groups are defined by drawing non-overlapping gates on the space defined by those two markers.

This approach ignores the complexity of a T cell population that is sampled from real blood. This rule, based on a few variables, leads to the selection of a fraction of cells only, that correspond to cells with the most extreme values of markers. In order to refine this classification, and to be able to classify (and to study) more cells, we proposed to develop a multi-categorical classification in order to infer the cell type of the non-identified cells. To proceed, we considered the measurements of 11 surface markers, along with the expression of the corresponding genes[11]. All these measurements were available on the single-cell basis. Hence the application of the multi-class sparse PLS here is not the in the high dimensional setting, however these data are expected to be heavily correlated. Even in this low dimensional case, the use of variable selection will help to improve the accuracy of the results.

[10] The surface markers are proteins present in the membrane of the cells.

[11] The genes that encodes the proteins associated to these markers.

Data analysis

The full data set is composed of the expression levels of ~ 20000 genes in ~ 1000 single T cells sampled at three different time points after the vaccine shots: 15, 136 and 908 days. At each time point, only a portion of the cells were annotated, i.e. classified in the different sub-groups of T cells during the experiments. The annotated cells present the most extremal phenotype[12], i.e. the cells that are easily classified. In this context, this study was based on multinomial SPLS to predict the phenotype of the remaining cells. The analysis was run on a specific subset of genes as we will explain in the following. Despite this application not being in high-dimension, variable selection will be useful to ensure the accuracy of the results. Indeed, we did not consider the whole 20000 genes because the data are very noisy and numerous genes are not informative regarding the classification of the cells according to their cell types. As previously mentioned, not only expression profiles but also measures of other phenotypic markers[13] are available for each cell.

Since this work is part of an international collaboration, we did the methodological development for the analysis but not the analysis directly that was handled by other members of our laboratory. The results presented here are not published yet, we just highlight the main conclusions to illustrate the usefulness of the multinomial-SPLS in the analysis of present genomic data.

The purpose was to classify the unidentified T cells (i.e. predict their types) and to find the genes that are associated to the partitioning of the cells into these different groups. The main issue was that it remained difficult to pre-select the genes that explain the cell types since a significant number of cells are unidentified. Thus, we first had to classify the cells according to the original 11 markers that are generally used to predict the cell types. We did also consider the level of expression of the 11 identified genes that encode for these markers (for a total of 22 predictors). Based on this prediction step, it was possible to order and pre-select the genes that are the most differently expressed between the groups of cells. Eventually, based on this pre-selection, we were able to perform a second round of classification by using 61 genes in addition to the 11 markers and the 11 associated genes (for a total of 82 predictors in the model). The pipeline of phenotype prediction is decomposed as follows:

1. A first round of prediction was performed by considering the measures of the 11 surface markers and the expression of the 11 associated genes. The model based on the multinomial-SPLS is trained on the subset of cells that are annotated and used to predict the types of the unknown cells. On the training set, a 5-fold cross-validation procedure is used to tune the hyper-parameters. The cross-validation

[12] The phenotype represents the observable characteristics of a cell or an organism.

[13] i.e. the quantification of the surface markers previously mentioned.

error[14] over the resamplings was $\sim$ 6%. The interest of the resampling in the V-fold cross-validation is to avoid over-fitting.

2. Following this first prediction, a Differential Expression Analysis (DEA)[15] is run to find the most differentially expressed genes between the different cell types, by using the predicted groups from the step 1. The genes are ordered according to the DEA to pre-select the ones that are more associated with the difference of phenotype[16].

3. Based on the differentially expressed genes, a second round of prediction was performed on all the cells with the multinomial-SPLS. This second analysis is restricted to 61 differentially expressed genes, plus the original 11 markers and 11 associated genes. The cross-validation error rate over resamplings (again 5-fold cross-validation) reaches $\sim$ 4% on this second run.

This application highlights the interest of dimension reduction by compression and variable selection, even when dealing with low dimensional data. It can also be noted that, even when using sparse approaches, a step of pre-selection is always useful, since the number of observations[17] remains small compared to the number of genes, i.e. $n = O(10^2)$ versus $p = O(10^5)$. Indeed, as we showed, variable selection is an interesting tool, however it cannot make miracles when the data are too noisy because of the numerous irrelevant covariates.

[14] i.e. the error rate associated to the best values of hyper-parameters tuned by cross-validation.

[15] The model used is introduced in Chapter 4.

[16] since the cell type is considered as a phenotypic trait.

[17] even if n grows in the most recent applications

Chapter 3

Conclusion and perspectives about the sparse PLS

In the previous chapters, we focused on dimension reduction approaches for supervised problems, especially when the response is categorical. Such statistical problems represent a huge interest for the integration of genomic data, in particular for the characterization of diseases or for the exploration of the genetic diversity between cells.

In this statistical context, important questions are raised regarding modeling or estimation algorithms. We especially discussed the integration of dimension reduction schemes in the framework of Generalized Linear Models (GLMs). Issues concerning convergence, calibration or computation time have to be handled, especially to guarantee the stability of the methods and thus the reliability of the interpretation of the results.

We proposed a method that performs compression and variable selection to solve a classification problem. It combines the Ridge regularized Iteratively Reweighted Least Squares (IRLS) algorithm and the sparse PLS in the context of the logistic regression. It is particularly appropriate for high dimensional data, which appears to be a crucial issue, especially in genomics. Our main consideration was to ensure the convergence of the IRLS algorithm, which is a critical point in logistic regression. Another concern was to properly incorporate a dimension reduction approach such as sparse PLS into the framework of GLMs. In particular, the algorithmic choice has a direct impact on the convergence and the stability of the method and thus on the interpretation of the results. We especially showed that non-convergent methods are unstable and unreliable regarding prediction and selection accuracy. On the contrary, the Ridge regularization ensures the convergence of the IRLS algorithm, which is confirmed in our simulations and tests on experimental data sets.

Applying adaptive sparse PLS as a second step on the pseudo-response produced by the IRLS respects the definition of sparse PLS regression to handle continuous response. Moreover, combining compression and variable selection increases the prediction performance and the selection accuracy of our method, which turns out to be more efficient than state-of-the-art approaches. Such combination also improves the dimension reduction, illustrated by the efficiency of our method for data visualization compared to standard supervised or unsupervised approaches. Furthermore it appears that previous procedures using sparse PLS with logistic regression encounter convergence issues linked to a lack of stability in the cross-validation. This point highlights the crucial importance of convergence when dealing with iterative algorithms.

The performance of our approach were assessed in a life-size situation. We ran an analysis of an experimental data set where the aim was to predict the relapse for breast cancer based on gene expression. The results of the simulation experiments were confirmed in this analysis, especially regarding the prediction performance and the stability of our method. Moreover, we introduced the use of stability selection to assess the accuracy of the variable selection in statistical methods combining compression and variable selection. Eventually, our approach was used in an on-going work to analyze and characterize single cell data. Single cell sequencing is very recent in the field of genomics and the results highlight the interest of our dimension reduction method in this context.

Nonetheless, a limitation of our approach is the lack of knowledge regarding the sparse PLS regression and especially the absence of theoretical results concerning its consistency or any oracle properties. Deriving such properties would be an interesting point to assess the underlying statistical properties of our method,

The Partial Least Squares (PLS) regression has been widely studied in the past. A wide variety of theoretical properties regarding the PLS have been derived. In particular, Phatak & de Hoog (2002) worked on the link between the optimization problem in the PLS and the conjugate gradient method (Hestenes & Stiefel, 1952) or with the method of Lanczoz (Lanczos, 1950) to approximate the extremal eigenvalues of large matrices. Phatak et al. (2002) studied the asymptotic variance of the PLS estimator. Krämer (2007) presented insights on the shrinkage properties of the PLS estimator (compared to the Ordinary Least Squares (OLS) estimator). Eventually, Blazère et al. (2014) introduced a new framework based on orthogonal polynomials, that allow to retrieve the previous known results regarding the PLS regression.

However, all this results concern the PLS regression and to our knowledge there does not exist any work regarding the theoretical properties of the sparse PLS yet, especially regarding the variable selection. Therefore, an interesting perspective would be to work on a theoretical characterization of the sparse PLS, first in the Gaussian framework, in order to support our empirical results.

A first potential direction to characterize the selection by sparse PLS would be to use the reformulation of the covariance maximization problem that defines the sparse PLS as a least squares problem (on the empirical covariance matrix) with an Elastic Net (EN) penalty. As introduced by Zou & Hastie (2005), it consists in a sum of penalties on the ℓ_1 and ℓ_2 norms of the coefficients. The question of the consistency of the EN regarding prediction and selection has been addressed by Ghosh (2007) and De Mol et al. (2009). Zou & Zhang (2009) or Jia & Yu (2010) also considered the high-dimensional case. These results mainly consider the case of an adaptive penalty. The interest here would be to benefit from the theoretical properties of the Elastic Net in the framework of the adaptive sparse PLS. However, such formulation would require to explicitly define a statistical model associated with the PLS. We hopefully will be able to investigate this point in the next months.

Another possibility to derive some properties about the sparse PLS would be to consider a particular formulation of the standard PLS regression. In practice, the coefficients[1] estimated by the PLS regression (c.f. Chapter 1) are actually the solution of a very specific least squares problem $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ when restraining the potential solutions to lie in a Krylov subspace, this point is detailed in Rosipal & Krämer (2006) or Krämer (2007). It could be possible to consider this least squares problem with a ℓ_1 penalty as a reformulation of the sparse PLS. However, such formulation raises different issues regarding the existence of the solution. Moreover, such framework is not constructive, in the sense that it does not expose a solution. Eventually, it requires a non-convex optimization procedure that especially works on convex subspaces. Following these different considerations, we will consider the characterization of sparse PLS by using Krylov subspaces on a longer term.

[1] from the linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$.

Part II

Unsupervised

Chapter 4

Introduction to matrix factorization for high dimensional data representation

In the second part of this manuscript, we will focus on unsupervised statistical analysis. Indeed, the question of data exploration and representation is crucial when analyzing high-dimensional data. We will focus on the issues related to the analysis of non-Gaussian data like single-cell expression profiles. We will develop statistical tools to explore the underlying geometry that is associated to the data, especially regarding the diversity between individuals and the dependency between variables.

We will present dimension reduction approaches, suitable to handle count data and based on matrix factorization. In particular, the development of data-specific framework for data exploration remains a challenging field of computational statistics. Considering more complex models allows to get a refined understanding of the data, however such framework raises important questions concerning the model inference, particularly regarding the algorithms, the optimization and the computational cost.

4.1 Matrix factorization for dimension reduction

We first present the interest of matrix factorization for dimension reduction and data exploration. We then describe how the standard Principal Component Analysis (PCA) solves a matrix factorization problem. It has to be noted that the questions about unsupervised approaches for data exploration are far from being recent. In particular, the PCA was introduced by Pearson (1901) and Hotelling (1933). The concepts about matrix factorization and low rank approximation were already discussed by Eckart & Young (1936). However, for fifteen years, these subjects have been back in the spotlight. In many different fields (genomics, text mining, signal processing, etc.), different issues have been raised regarding the scale and the dimensionality of the data, but also concerning the appropriate geometric representation.

4.1.1 Why factorize matrix?

We consider a data matrix $\mathbf{X} = [x_{ij}] \in \mathbb{R}^{n \times p}$ of dimension $n \times p$. Each row $\mathbf{x}_i \in \mathbb{R}^p$ of the matrix represents a vector of observations (among n) of p variables (e.g. gene expression). For instance, the entry x_{ij} stores the read counts (i.e. expression level) of gene j in sample i. The reader should bring attention to this point: in our formulation, rows correspond to observations and columns to the recorded variables[1] in the data matrix $\mathbf{X}$.

Factorizing a matrix consists in representing the observations and variables as linear combinations of latent directions. These latent components or factors lie in a lower dimensional space of dimension K and are assumed to be a good approximation of the data in a lower and more representable dimension. Singh & Gordon (2008) present an overview about the definition and interest of matrix factorization, sometimes called dictionary learning in the literature (Mairal et al., 2012).

In particular, the factorization of the matrix $\mathbf{X}_{n \times p}$ consists in searching for two factor matrices, $\mathbf{U} = [u_{ik}] \in \mathbb{R}^{n \times K}$ of dimension $n \times K$ and $\mathbf{V} = [v_{jk}] \in \mathbb{R}^{p \times K}$ of dimension $p \times K$ such that:

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}^T\,, \tag{4.1}$$

i.e. for each observation:

$$x_{ij} \approx \sum_{k=1}^{K} u_{ik}\, v_{jk}\,,$$

see Figure 4.1. In this decomposition the columns of $\mathbf{U}_{n \times K}$ represent the coordinates of the observations in the latent subspace and the columns $\mathbf{V}_{p \times K}$ are the contributions of

[1] This convention is sometimes reversed in the literature regarding matrix factorization.

the variables to the latent components. Therefore, the columns of $\mathbf{U}$ and $\mathbf{V}$ summarize the structure and organization of respectively individuals and variables in the original (and possibly high-dimensional) space.

The main question about matrix factorization concerns the definition of the approximation "$\approx$" in Equation (4.1) (Singh & Gordon, 2008; Févotte & Cemgil, 2009). This point will be the main concern of Part II (Chapters 4 to 7). At first, a reasonable choice could be the least squares approximation, nonetheless we will see different ways to define the approximation $\mathbf{X} \approx \mathbf{U}\mathbf{V}^T$ and so to compute the factor matrices $\mathbf{U}$ and $\mathbf{V}$. The dimension reduction is considered to be efficient when the approximation stands with $K \ll \min(n, p)$. At least, due to algebraic concern, we generally have the relation $K \leq \operatorname{rank}(\mathbf{X}) \leq \min(n, p)$.

The usefulness of matrix factorization to process dimension reduction can be manifold. First, this is inherently a compression technique as the cumulative dimension of the factor matrices $\mathbf{U}$ and $\mathbf{V}$, i.e. $n \times K + p \times K$, is smaller than the dimension $n \times p$ of $\mathbf{X}$ as soon as $K \leq 0.5 \times \min(n, p)$. Hence storing $\mathbf{U}$ and $\mathbf{V}$ instead of $\mathbf{X}$ is less costly. Then, the columns of $\mathbf{U}$ or $\mathbf{V}$ can be used to understand the organization of respectively individuals and variables, as they expose hidden structures within the data. For instance, the latent factors can be a good basis to perform clustering of the variables (e.g. genes) or of the observations (e.g. cells) in a lower dimensional space (Yeung & Ruzzo, 2001; Xu et al., 2003; Ding et al., 2005; Lee et al., 2010; Wang et al., 2013). If K is very small, i.e. $K \ll \min(n, p)$, the columns of $\mathbf{U}$ and $\mathbf{V}$ represent a particular interest for data visualization and data exploration (Bishop & Tipping, 1998). Finally it can be used for matrix completion (Witten et al., 2009) or matrix reconstruction (Meng & De La Torre, 2013).

Matrix factorization has been applied in many domains as image processing and text mining (Lee & Seung, 1999; Xu et al., 2003), collaborative filtering and user recommendation system (Salakhutdinov & Mnih, 2011; Gopalan et al., 2014), spectral analysis and signal unmixing (Hoffman et al., 2010; Dikmen & Févotte, 2012), and even transcriptomics (Brunet et al., 2004; Kim & Park, 2007; Wang et al., 2013; Yang & Michailidis, 2016).

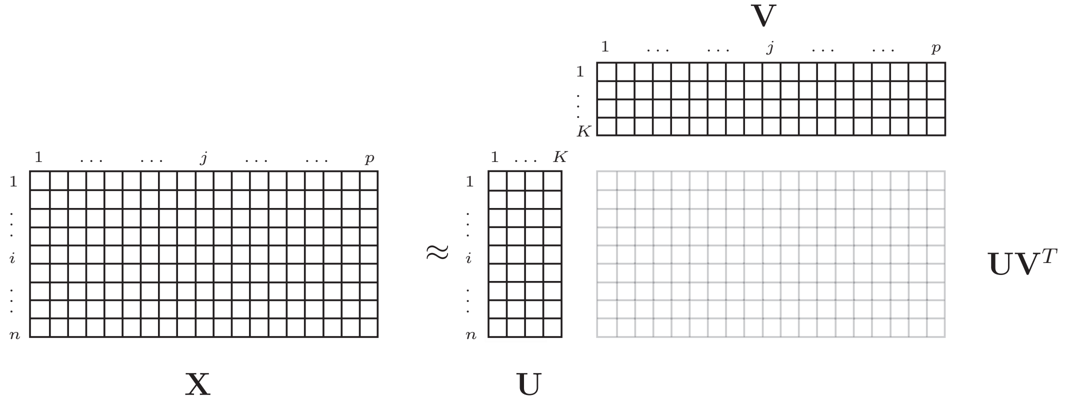


Figure 4.1 – Illustration of the dimension reduction by matrix factorization. The dimensions of the factors $\mathbf{U}$ and $\mathbf{V}$ are lower than the dimension of the data matrix $\mathbf{X}$.

4.1.2 Behind Principal Component Analysis

A standard approach for data exploration and visualization is the well known and widely used PCA (Hotelling, 1933, or see Abdi & Williams, 2010 for a review). We first recall the definition of PCA and then highlight its link with matrix factorization.

The purpose of PCA is to transform the data thanks to a linear projection into a lower dimensional subspace by maximizing the variance of the projection coordinates. It seeks for K orthogonal directions called principal components that explain most variability in the data. The principal components are constructed (for $k = 1, \ldots, K$) as linear combination of the data $\mathbf{t}_k = \mathbf{X}\mathbf{w}_k$, i.e. $t_{ik} = \sum_j x_{ij}\, w_{jk}$ for $i = 1, \ldots, n$. These relations can be summarized with the following matrix notation $\mathbf{T} = \mathbf{XW}$ where the matrix $\mathbf{T} = [t_{ik}] \in \mathbb{R}^{n\times K}$ stores (in columns) the component coordinates or score $(\mathbf{t}_k)_{k=1:K}$ and the matrix $\mathbf{W} = [w_{jk}] \in \mathbb{R}^{p\times K}$ stores (in columns) the variable weights or loadings $(\mathbf{w}_k)_{k=1:K}$. This weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_K$ are defined such that the empirical variance $\widehat{\text{Var}}(\mathbf{t}_k)$ is maximal. The empirical variance is expressed depending on the centered data matrix $\mathbf{X}_c$ and the objective function is therefore:

$$\mathbf{w}_k = \underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmax}}\ (\mathbf{w}_k)^T(\mathbf{X}_c)^T\mathbf{X}_c\mathbf{w}_k\,, \tag{4.2}$$

under the constraint of orthogonality between $\mathbf{t}_1, \ldots, \mathbf{t}_K$. The resolution gives a closed-form solution for $\mathbf{w}_1, \ldots, \mathbf{w}_K$, that are the K first dominant eigenvectors of the empirical covariance matrix $(\mathbf{X}_c)^T\mathbf{X}_c$.

Singular Value Decomposition

In practice, the resolution of the eigen-problem in the PCA appears to be equivalent to a matrix factorization problem. We now explain the link between PCA and Singular Value Decomposition (SVD). The SVD is an algebraic decomposition of any matrix (Klema & Laub, 1980). The SVD of the matrix $\mathbf{X}_{n\times p}$ is defined as follows:

$$\mathbf{X} = \widetilde{\mathbf{U}}\mathbf{D}\widetilde{\mathbf{V}}^T .$$

The matrices $\widetilde{\mathbf{U}}$, $\widetilde{\mathbf{V}}$ and $\mathbf{D}$ verify the following properties (c.f. Figure 4.2a):

- $\mathbf{D}_{r\times r} = \mathrm{diag}(\delta_1, \dots, \delta_r)$ is the diagonal matrix of ordered singular values $\delta_1 > \dots > \delta_r$ of the matrix $\mathbf{X}$. The non-zero singular values correspond to the square roots of the non-zero eigenvalues of the matrices $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$.

- The matrices $\widetilde{\mathbf{U}}_{n\times r}$ and $\widetilde{\mathbf{V}}_{p\times r}$ are orthonormal, i.e. $\widetilde{\mathbf{U}}^T\widetilde{\mathbf{U}} = \mathrm{Id}_r$ and $\widetilde{\mathbf{V}}^T\widetilde{\mathbf{V}} = \mathrm{Id}_r$. The columns of $\widetilde{\mathbf{U}}$ (resp. $\widetilde{\mathbf{V}}$) are the left-sided (resp. right-sided) singular vectors of the matrix $\mathbf{X}$, and correspond to the eigenvectors of $\mathbf{X}\mathbf{X}^T$ (resp. $\mathbf{X}^T\mathbf{X}$).

- r is the rank of the matrix $\mathbf{X}$ (and therefore the rank of $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$).

Solving the covariance maximization problem defining the PCA, as defined in Equation (4.2), is actually equivalent to processing the SVD of the centered data matrix $\mathbf{X}_c$. Indeed, the weight vectors $(\mathbf{w}_k)_{k=1:K}$ are the dominant eigen-vectors of the empirical covariance matrix $(\mathbf{X}_c)^T\mathbf{X}_c$, which correspond to the columns of $\widetilde{\mathbf{V}}$ when applying the SVD to $\mathbf{X}_c$. Hence, when denoting respectively by $\widetilde{\mathbf{U}}_{1:K}$, $\widetilde{\mathbf{V}}_{1:K}$, $\mathbf{D}_{1:K}$ the matrices composed of the respective K first columns of $\widetilde{\mathbf{U}}$, K first columns of $\widetilde{\mathbf{V}}$, and K first rows and columns of $\mathbf{D}$, the principal components are computed as $\mathbf{T}_{n\times K} = \mathbf{X}_c\widetilde{\mathbf{V}}_{1:K}$, i.e. $\widetilde{\mathbf{U}}_{1:K}\mathbf{D}_{1:K}$ by orthogonality of $\widetilde{\mathbf{V}}_{1:K}$.

A least-squares formulation

The SVD of the matrix $\mathbf{X}_{n\times p}$ leads to an exact decomposition $\mathbf{X} = \mathbf{U}\mathbf{V}^T$ where $\mathbf{U} = \widetilde{\mathbf{U}}\mathbf{D}$ and $\mathbf{V} = \widetilde{\mathbf{V}}$ are respectively the latent components (or scores) and the variable coefficients (or loadings).

In fact, the SVD has a geometrical interpretation and can be used to find a matrix $\mathbf{U}\mathbf{V}^T$ that approximates the matrix $\mathbf{X}$ (c.f. Figure 4.2b). When setting the dimension $K < \mathrm{rank}(\mathbf{X})$, the matrices $\mathbf{U} = \widetilde{\mathbf{U}}_{1:K}\mathbf{D}_{1:K}$ and $\mathbf{V} = \widetilde{\mathbf{V}}_{1:K}$ verify the following property: $\mathbf{U}\mathbf{V}^T$ is the matrix of rank K that minimizes its Frobenius distance to the matrix $\mathbf{X}$ as stated by Eckart & Young (1936). Therefore, the matrix $\mathbf{U}\mathbf{V}^T$ verifies:

$$\mathbf{U}\mathbf{V}^T = \underset{\mathrm{rk}(\mathbf{M})=K}{\mathrm{argmin}} \ \|\mathbf{X} - \mathbf{M}\|_F^2 , \tag{4.3}$$

where the Frobenius norm of a matrix $\mathbf{X} = [x_{ij}]$ is defined by

$$\|\mathbf{X}\|_F^2 = \text{trace}(\mathbf{X}^T\mathbf{X}) = \textstyle\sum_{i,j} (x_{ij})^2 \, .$$

Hence, when expending $\|.\|_F^2$, the rows of $\mathbf{UV}^T$ minimize the ℓ_2 distance:

$$\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{m}_i\|_2^2$$

between the rows $\mathbf{x}_i$ of $\mathbf{X}$ and the rows $\mathbf{m}_i$ of $\mathbf{M}$. The vectors $\mathbf{m}_i$ are interpreted as the approximations of the observation vectors $\mathbf{a}_i$ in the lower dimensional subspace of dimension K.

The PCA applied to a data matrix $\mathbf{X}$ (or the SVD of $\mathbf{X}_c$) is therefore a least square (and low rank) approximation. This point needs to be kept in mind when thinking about applying the PCA to non-Gaussian data as we will see in Chapters 5 and 6.

Sparse PCA

Following the concept introduced in Part I, it is also possible to consider the hypothesis of parsimony when developing unsupervised statistical approaches suitable for compression. In particular, it is possible to combine matrix factorization with variable selection. When analyzing high-dimensional data, the variable selection process will improve the dimension reduction since only the pertinent variables will be considered to construct the latent directions that explain the structure of the data.

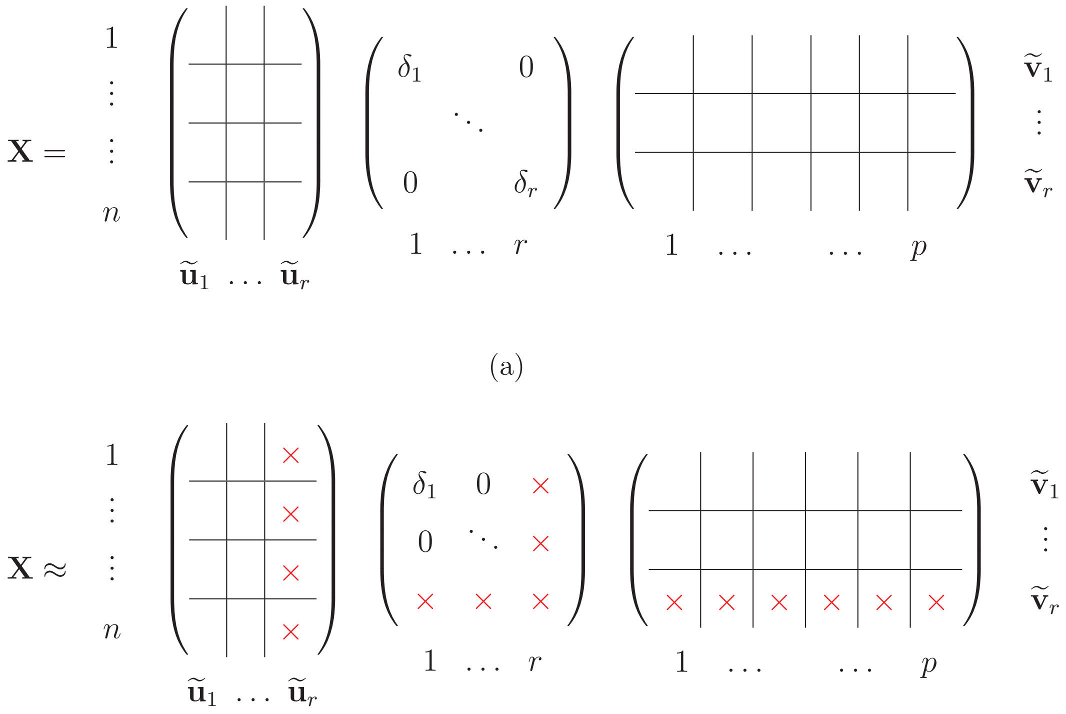
In particular, the sparse PCA is formulated as a penalized variance maximization problem (Jolliffe et al., 2003) with a sparsity-inducing penalty on the weight vectors $\mathbf{w}_k$, hence defining sparse principal components $\mathbf{t}_k$. The penalty explicitly concerns the ℓ_1-norm of the weight vectors, i.e.

$$\mathbf{w}_k = \underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmax}} \; (\mathbf{w})^T(\mathbf{X}_c)^T\mathbf{X}_c\mathbf{w} + \nu \sum_{j=1}^{p} |w_j| \, , \tag{4.4}$$

where $\nu > 0$ is a penalty constant. As in the case of the sparse PLS (see Chapter 1), the optimization of (4.4) is tricky.

Zou et al. (2006) reformulated the problem (4.4) as a matrix approximation problem. Similarly, Shen & Huang (2008) introduced the sparse SVD as a low rank approximation based on a least squares loss function with an ℓ_1 penalty i.e.

$$\underset{\mathbf{u}\in\mathbb{R}^n,\mathbf{v}\in\mathbb{R}^p}{\text{argmin}} \; \|\mathbf{X} - \mathbf{uv}^T\|_F^2 + \sum_{j=1}^{p} \text{pen}_\nu(v_j) \, ,$$


$\mathbf{X} =$
1
⋮
⋮
n
$\widetilde{\mathbf{u}}_1 \dots \widetilde{\mathbf{u}}_r$
δ_1
0
⋱
0
δ_r
$1 \dots r$
$1 \dots \dots p$
$\widetilde{\mathbf{v}}_1$
⋮
$\widetilde{\mathbf{v}}_r$
$\mathbf{X} \approx$
1
⋮
⋮
n
$\widetilde{\mathbf{u}}_1 \dots \widetilde{\mathbf{u}}_r$
δ_1 0 ×
0 ⋱ ×
× × ×
$1 \dots r$
$1 \dots \dots p$
$\widetilde{\mathbf{v}}_1$
⋮
$\widetilde{\mathbf{v}}_r$

(a)

(b)

Figure 4.2 – SVD and low rank approximation. (a) SVD of the matrix $\mathbf{X}$ or rank r, i.e. $X = \widetilde{\mathbf{U}}\mathbf{D}\widetilde{\mathbf{V}}^T$. (b) Low rank approximation based on the SVD, i.e. $\mathbf{X} \approx \widetilde{\mathbf{U}}_{1:K}\mathbf{D}_{1:K}\widetilde{\mathbf{V}}_{1:K}$. The × refers to the columns with index $K+1,\dots,r$ that have been removed.

where $\text{pen}_\nu(\cdot)$ is a sparsity-inducing penalties (depending on $\nu > 0$), for instance, based on the ℓ_1-norm. Witten et al. (2009) also presented a reformulation of the problem associated to the SVD under a sparsity constraint. Other approaches similar to sparse PCA or sparse SVD have been proposed in parallel in the machine learning community, especially sparse coding and matrix factorization (Bach et al., 2008; Mairal et al., 2009, 2012).

4.2 New data specificity

The renewal of interest for exploratory statistics is linked to the recent evolution regarding the scale and the specific types of the data in many fields. The questions about exploration and representation of the data are central in many analysis, especially to get an insight on the structures organizing the observations and the variables or to explicitly visualize the data. Such context constitutes an important issue when dealing with large-scale and high dimensional data that need to be represented in a tangible way to be interpreted. For instance, in genomic data analysis, the aim is generally to investigate the complex dependencies between genes or the influence of the environmental conditions. As the experimental validation is heavy and expensive in Biology (e.g. to investigate the role of a pool of genes), the statistical exploration of genomic data is used to get a precise understanding of the latent configuration of the data.

Moreover, we will deal with a very recent type of genomic data, especially single-cell data. Whereas standard RNA sequencing (RNA-seq) used to capture the average expression of genes across a population of cells (called bulk sequencing), it is now possible to monitor the expression of genes at the level of single cells. As a deeper recording of the molecular activity in cells, it represents a unique insight on the individual diversity between cells from a same tissue or organism. The choice of the statistical methods to analyze single-cell expression profiles depends on the question that needs to be addressed: how the cells are organized and related to each other? Do they express the same genes in the same environment and condition? These questions will motivate the use of matrix factorization to explore the data.

In this section, we briefly introduce a specific single-cell data set and the biological questions that are related. Then, we explain the specificity of single-cell data from a statistical point of view. These two points will motivate the development of a specific dimension reduction framework. In particular, gene expression data are not just concerned by high dimensionality issues. Indeed, in the framework of Next-Generation Sequencing (NGS), transcriptomic data are also composed of non continuous signal as counts that quantify the abundance of some nucleic sequences (like RiboNucleic Acids (RNAs)). For instance, in the context of heteroskedastic data, the geometry induced by the standard Euclidean metric may not be suitable to catch the structure of the data. This point will be discussed in Chapter 5.

4.2.1 An example of single-cell expression data

Single-cell sequencing is a recent technology (Gawad et al., 2016). The cells are isolated from each other and their genetic material is individually sequenced. Characterizing the expression of genes at the single-cell level gives a unique insight on the genomic diversity between cells of the same organism. As introduced in Chapter 2, our interest for statistical exploration was motivated by a collaboration with Jeff Mold from the *Karolinska Institutet* (Stockholm, Sweden). The project focuses on the study of the transcriptomic landscape (at the single-cell level) in a population of lymphocyte T cells that were sampled after a vaccine shot at three time points (15, 136 and 908 days after the inoculation). The interest is to characterize an immune response from the T cell point of view, in particular to understand the organization of the population of T cells through times and qualify its genetic homogeneity or heterogeneity.

Lymphocyte T cells are divided into different groups, tagged as Effector or Memory cells. These discrimination is based on phenotypic markers. However, T cells are also structured by genealogical links. Indeed, a unique genetic marker allows to identify all the cells that originate from a common ascendant after successive divisions. Such group of cells constitute a clone. One of the main question raised in this study is the differences between the clonal and phenotypic organization of the cells, especially regarding their evolution across time. After the vaccine shot, the number of T cells specific to the antigen in the vaccine rises quickly during a few days, and then this number decreases within a few hundred of days. However, these T cells does not disappear, a pool of cells remains so that they can be activated if they encounter their specific antigen during a future infection. We will focus on the exploration and representation of single-cell expression profiles to characterize the different levels of structures between cells (clone versus phenotype) and monitor the evolution of these structure through time.

Single-cell data present different characteristics and their analysis appears to be more challenging compared to RNA-seq data (Stegle et al., 2015). One of the most specific pitfall is the abundance of drop-out events in the gene expression profile. Indeed, a zero count may refers to an absence of read or to a failure in the experiment due to the short amount of genetic material available in a single cell. Therefore, an unknown and random proportion of zeros corresponds to unobserved values. In particular, the sequencing technology used in this project, *SMARTseq2*[2] is assumed to capture between 40 and 50% of the genetic material in a single cell. Moreover, the proportion of cells from the same type that express the same genes is very low, because the transcription is a stochastic process (Marinov et al., 2014). This reflects the high variability of expression

[2] This technology was published in Picelli et al. (2013), however in this experiment, the sequencing is based on custom reagents that are not published yet.

between cells. For instance, Figure 4.3 shows the empirical distribution of the expression level of four different genes across all cells in our single-cell data set. All these genes are characterized by a huge number of zeros. However, the distributions may present different characteristics (uni-modal or bi-modal for instance).

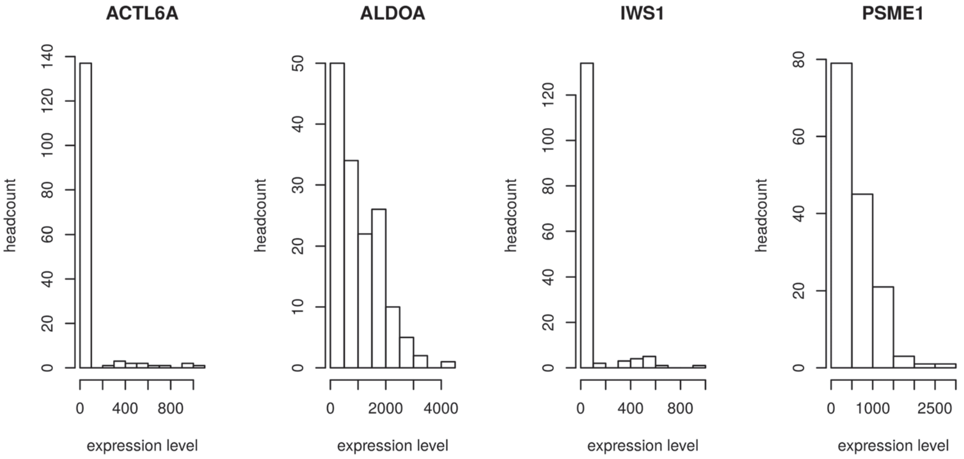


Figure 4.3 – Amplification of the zeros in the read count distribution of different genes

4.2.2 Distributions to model count expression profiles

As previously mentioned, gene expression profiles are count matrices. A first idea to model such data is to use the Poisson distribution. It has been especially used for NGS data analysis (Marioni et al., 2008; Srivastava & Chen, 2010; Witten, 2011). However, the Poisson model is restrictive as a single parameter determines the moments of first and second orders (i.e. position and dispersion). It lacks flexibility especially since NGS data are often over-dispersed regarding what would be expected in a Poisson model[3] (Anders & Huber, 2010). Based on this assessment, the Negative Binomial distribution is an alternative to model over-dispersion in NGS count data (Anders & Huber, 2010; Bonafede et al., 2015). Considering a Negative Binomial $X \sim \mathcal{NB}(r, \pi)$ with $r > 0$ and $\pi \in (0, 1)$, the probability mass function of X is defined[4] by:

$$p(x\,;\, r, \pi) = \frac{(x + r - 1)\,!}{x\,!\,(r-1)\,!}\,\pi^r\,(1-\pi)^x\,,$$

for any integer $x > 0$. Some estimation methods for the Negative Binomial distribution are introduced in Johnson et al. (2005) or Karlis (2005).

[3] In Poisson distributions, the variance is equal to the expectation.

[4] c.f. appendix Chapter C

An interesting point is that the Negative Binomial distribution can be formulated as a hierarchical model[4]. Indeed, if we consider a Poisson variable $X \,|\, \lambda \sim \mathscr{P}(\lambda)$ with $\lambda > 0$, we may add a prior distribution on the parameter λ. When choosing a Gamma[5] prior, the model becomes:

$$\begin{aligned} &\lambda \sim \Gamma(\alpha_1, \alpha_2)\,, \\ &X \,|\, \lambda \sim \mathscr{P}(\lambda)\,. \end{aligned} \tag{4.5}$$

The marginal distribution of X, i.e. $p(X\,;\Omega)$ with $\Omega = (\alpha_1, \alpha_2)$, can be derived by integrating the joint likelihood $p(X,\lambda) = p(X \,|\, \lambda)\, p(\lambda)$ over λ. In this case, the marginal is especially the Negative Binomial distribution $\mathcal{NB}(\alpha_1, \alpha_2/(\alpha_2+1))$, hence accounting for over-dispersion[4]. This framework was for instance applied by Christiansen & Morris (1997) for Poisson regression. Such hierarchical model is called the Gamma-Poisson (GaP) model. We will develop the concept of matrix factorization in this framework in Chapters 5 and 6. We will especially see how to process dimension reduction in over-dispersed count data.

4.2.3 Zero-inflated count data

As previously stated, single-cell transcriptomic data are even more specific than over-dispersed count. They are indeed characterized by drop-out events, i.e. an amplification of the zeros in the data. This phenomenon defined as zero-inflation (Hall, 2000) may have a huge impact on statistical analysis and have been a huge concern in single data analysis with the development of specific methods (Pierson & Yau, 2015).

In zero-inflated data, the signal is a superposition of two sources. In our data, non-null values especially correspond to an effective signal whereas null values may refer to a true zero (absence of read) or to an artificial zero (failure of the experiment). The origin of the zeros (from one source or another) cannot be determined. The underlying distribution modeling such behavior is therefore a mixture of two distributions, here a count-generative distribution and a zero-generative distribution. For example, the reader may refer to Lambert (1992) for an introduction of the zero-inflated Poisson model. In such context, the count data X is supposed to follow a Dirac-Poisson mixture, i.e.

$$X \sim (1-\pi)\,\delta_0 + \pi\,\mathscr{P}(\lambda)\,.$$

[5] We use the following standard parametrization of the Gamma distribution $\Gamma(\alpha_1,\alpha_2)$ with parameters $\alpha_1, \alpha_2 > 0$ and with the density:

$$p(u\,|\alpha_1,\alpha_2) = u^{\alpha_1-1}\frac{(\alpha_2)^{\alpha_1}\,e^{-\alpha_2\,u}}{\Gamma(\alpha_1)}$$

where $\Gamma(\cdot)$ is the Gamma function $\Gamma : x > 0 \mapsto \int_{\mathbb{R}+} t^{x-1}\,e^{-t}\,\mathrm{d}t$

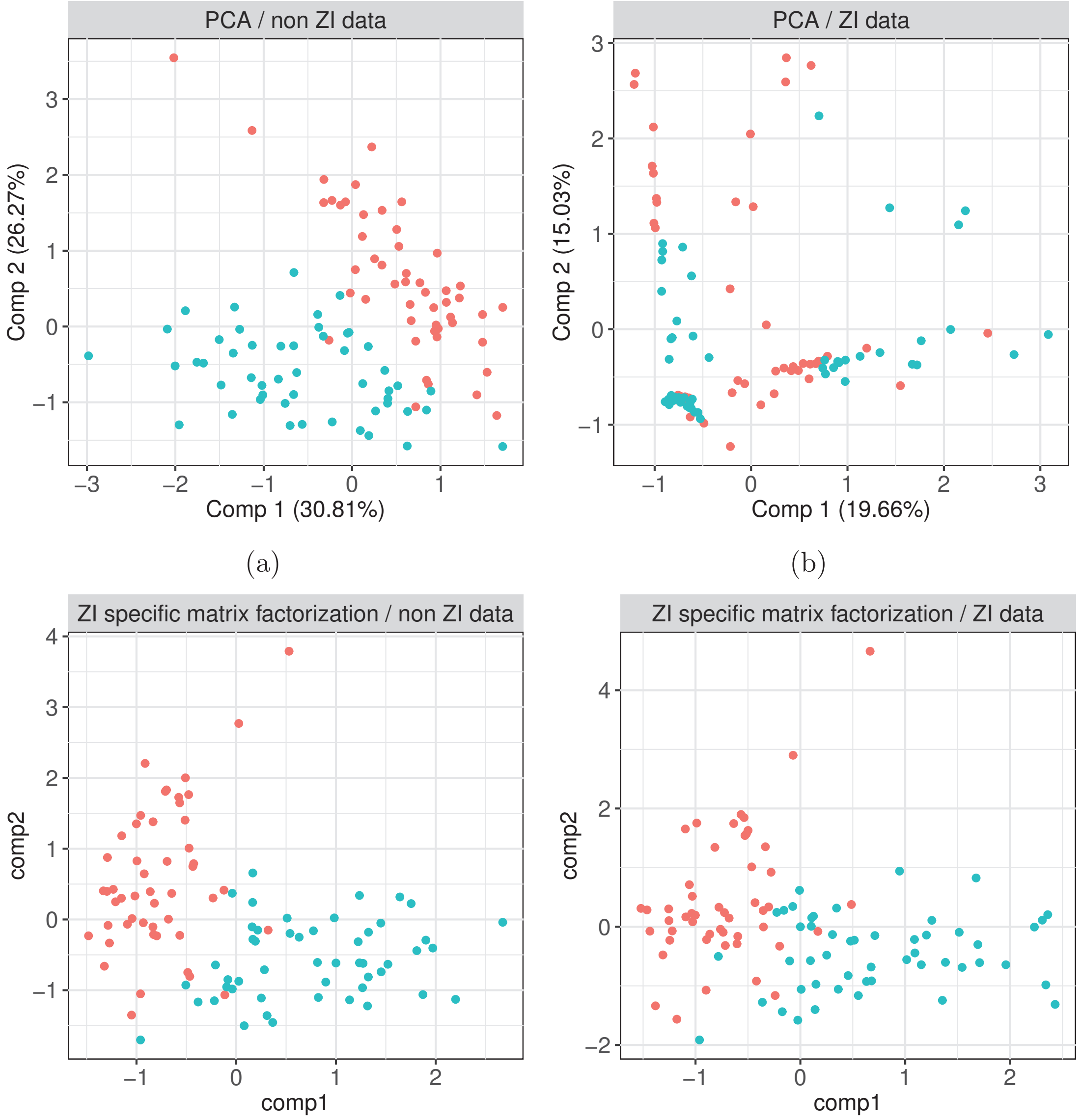
The parameter $\pi \in [0, 1]$ is the probability that the observation is drawn from the Poisson distribution of parameter $\lambda > 0$. We precise that we consider only a zero-inflated model and not a hurdle model (Dalrymple et al., 2003) because a null value may be a true observation in the data.

Standard approaches may not respond well to data following such generative process. For instance, the PCA is based on the measure of covariance, and the addition of many zeros may totally change the covariance structure in the data. In fact, the PCA is very sensitive to corrupted observations. Different approaches have been proposed to robustify the PCA (De La Torre & Black, 2003; Candès et al., 2011; Meng & De La Torre, 2013; Brooks et al., 2013), however they are based on metrics inappropriate for count data (c.f. Chapter 5).

We take a small example to illustrate this point. We generate a data matrix with $n = 100$ observations and $p = 10$ variables according to the GaP model with latent factors that we will study in the following chapters. We specifically set different values to the hyper-parameters to define two groups of observations. We use the PCA to visualize the data. We plot the first two components (see Figure 4.4a) that explain more than 55% ($\sim 30.8\% + \sim 26.3\%$) of the variability in the data[6]. The latent structure associated with the first two axis discriminate clearly between the two classes of observations. Now, we simulate random drop-out events and replace some observations by null values at random positions in the data (so that the proportion of artificial zeros is between 0.3 and 0.6). Again, we use the PCA and visualize the first two components (c.f. Figure 4.4b). We observe that the percentage of explained variability plunges to $\sim 32\%$ (versus $\sim 55\%$ previously) and the two groups are totally mixed.

Zero-inflated (ZI) data must be handled with caution. In particular, the PCA and the SVD may not be appropriate to explore such data. For instance, Pierson & Yau (2015) proposed a latent factor model suitable for zero-inflated Gaussian data. Our questioning concerns particularly the validity of standard criteria such as least squares or covariance when the data are far from being Gaussian. This point will be developed in Chapters 5 and 6. As a teaser, we show the results on the same data set obtained by the matrix factorization method that we developed, where zero-inflation is accounted for (see Figures 4.4a and 4.4b). We notice that the data structure is retrieved even in the case of ZI data with drop-out events, on the contrary to the PCA.

[6] The variability explained by a component is the ratio between the empirical variance of the projection of the data on this components and the empirical variance in the data.



(a)

(b)

(c)

(d)

Figure 4.4 – Example of PCA on ZI data. (a) Observation scores from PCA on a data set with two groups of observations. (b) PCA score with the same data but random drop-out events add a huge proportion of zero in the data (proportion between 0.3 and 0.6). (c) Example of specific matrix factorization for ZI data on the standard data. (d) Example of specific matrix factorization for ZI data on the ZI data. (In Fig (c) and (d), we do not consider the percentage of explained variance since it is not an appropriate criterion for our approach as we will see in Chapters 5 and 6).

4.2.4 First developments regarding the single-cell project

Before introducing data-specific dimension reduction methodologies in the following chapters, we present an overview of some results regarding the analysis of the expression profiles of single T cells with PCA-based methods. Indeed, the PCA has been used in previous studies regarding single-cell data, see for instance Buganim et al. (2012) or Gaublomme et al. (2015), thus we did focus on the PCA as a preliminary step for data exploration.

Although the point of the following chapters will be to explain why it could be appropriate to use specific dimension reduction methods in the exploratory analysis of non-Gaussian data, the PCA remains useful as a fast procedure to visualize the latent organization of large-scale data set. In particular, the single-cell transcriptomic data are non-Gaussian, however it is possible to transform the counts into a pseudo-continuous signal and apply standard dimension reduction approaches. For instance, Anders & Huber (2010) or Cleynen et al. (2013) discussed the transformation of count expression profiles. Johnson et al. (2005) detailed useful transformations depending on the data underlying distribution, especially variance stabilization transformation. The two main transformations used in the context of count data are the log transform, i.e. $x \mapsto \log(x+1)$ or the Anscombe transform, i.e. $x \mapsto 2\sqrt{x+3/8}$. Such transforms are appropriate when the averaged signal is high in the data. In this case, the transformed counts are expected to be approximately Gaussian. In our experimental analysis, we did use the Anscombe transform. However, the transformed signal remains non-Gaussian because of the zero-inflation in single-cell data.

Adaptive sparse PCA

The question addressed now mainly concerns the characterization of the different organization of the cells: clonal versus phenotypic. A first issue concerns the number of genes in the data set (around 20000). Among these thousands of genes, we would like to focus on the ones that carry an interesting signal and not just noise. To do so, it is possible to process a differential expression analysis and rank the genes based on their difference of expression between the conditions: differentially expressed (DE) genes associated to the clonal effect or DE genes associated to the phenotypic effect. Such analysis if based on the inference of complex Generalized Linear Models (GLMs)[7] and the optimization process is time consuming. In order to reduce the computation time, we used sparse PCA as a preselection step to withdraw non-relevant genes.

Based on the formulation of sparse PCA by Witten et al. (2009) and the work introduced

[7] In particular zero-inflated Negative-Binomial models.

in Part I, we developed an adaptive version of sparse PCA where the penalty on the ℓ_1-norm of the weight vector $\mathbf{w}_k$ is adjusted to penalize more the less pertinent variables. The motivation of such principle is that the contribution of the pertinent variables in the data representation are softened by the small yet non null contributions of the numerous non-relevant variables, particularly in high dimensional data. To process a more precise selection, we want to adapt the selection regarding the relative importance of each variable. The associated problem can be written as follows:

$$\mathbf{w}_k = \underset{\mathbf{w}\in\mathbb{R}^p}{\operatorname{argmax}}\ (\mathbf{w})^T(\mathbf{X}_c)^T\mathbf{X}_c\mathbf{w} + \nu\sum_{j=1}^{p}\nu^j|w_j|\,, \tag{4.6}$$

where the penalty constant $\nu^j > 0$ depends on the variable j. In particular, the ℓ_1-penalty on the weight of the variable j in the component k, i.e. w_{jk}^{spca}, is regulated by the term $\nu^j = 1/|w_{jk}^{\text{pca}}|$ where w_{jk}^{pca} is the weight of the variable j in the component k in standard PCA. Thus, the variables with small weights w_{jk}^{pca} in absolute and then non-informative will be more penalized and therefore discarded.

In practice, we focus on each day separately: "D15" and "D136"[8]. When dealing with the full data set on both days conjointly, the sparse PCA will select mostly genes that explain the day variability, as gene expression seems to be very variable between days. Moreover, in that case, it would not be possible to identify a situation where genes that discriminate clones are not the same at each day. Eventually, the idea behind the separation of the days is to enforce robustness in our results as we repeat the analysis. For each day, we select (by sparse PCA) among all genes those that contribute the most to the variability: we obtain 1974 genes for D15 and 1932 genes for D136. An interesting point is that $\sim$ 99% for D15 and $\sim$ 90% for D136 of the selected genes are within the top 10% most differentially expressed genes between clones, hence processing pre-selection with sparse PCA seems consistent with the results of differential expression analysis, and a less time-consuming alternative. However, we did not investigate this direction any further in our experimental study. Indeed, at the moment (as mentioned in the case of the sparse PLS in Chapter 3), there does not exist any theoretical characterization of the properties of sparse PCA regarding variable selection. Moreover, as explained in the next chapter, we decided to focus on model-based approaches for matrix factorization. Besides this preliminary analysis, we will present some other results regarding data visualization in Chapter 6.

[8] The data at "D908" were not available at that time.

Chapter 5

Principle of PCA on counts

In regression problems, when the response is not Gaussian, we consider Generalized Linear Models because the linear regression would not comply with the constraints inherent to the data. For instance, the least squares regression cannot be used to predict a binary response. The same remark holds in the context of matrix factorization. The Principal Component Analysis and Singular Value Decomposition are based on a least squares approximation. Therefore, it cannot be sure that the reconstructed matrix $\mathbf{U}\mathbf{V}^T$ that approximate the data matrix $\mathbf{X}$ will respect the specificity of the data. We are considering count data, hence discrete and non-negative. To account for the non-negativity, the Non-negative Matrix Factorization was developed to estimates the factors $\mathbf{U}$ and $\mathbf{V}$ under constraints of non-negativity. As we will see, the geometry induced by the Euclidean metric is not the most appropriate for count, thus we will rather consider model-based matrix factorization approaches, that are based on the concept of Generalized PCA.

As introduced in the previous chapter, genomic data are generally over-dispersed and this characteristic has to be accounted for when analyzing such data. Therefore, we will not consider models for matrix factorization based on the Poisson distribution. We will present a formulation based on the hierarchical Gamma-Poisson model that accounts for over-dispersion. In this scheme, the factors $\mathbf{U}$ and $\mathbf{V}$ are considered as latent variables. Moreover, the standard approaches for estimation as the Maximum Likelihood Estimation or the Expectation-Maximization algorithm are not appropriate because the marginal distribution of the data or the posterior distributions of the factors are intractable. To overcome the estimation problems, we use the framework of variational inference, a method that approximates the posterior distribution of a model. A standard procedure to infer the posterior would be to use Markov Chain Monte Carlo approaches, however the variational inference is computationally more efficient.

Since we generally study high dimensional data, the question of sparsity remains central especially in the context of dimension reduction procedure. After introducing the model and the inference scheme, we will eventually present some works regarding variable selection approaches that will guide us in the next chapter to develop a sparse model for matrix factorization.

Before going any further, in the next two boxes, we introduce some notations about probability distributions that will be useful in the following.

Likelihood and distributions
For the purpose of notations, the densities of continuous random variables and the probability mass functions of discrete random variables are similarly denoted. For instance if a variable X follows a Poisson distribution, i.e. $X \sim \mathscr{P}(\lambda)$, the associated likelihood is $p(x\,;\,\lambda)$. Similarly if a random variable U follows a Gamma distribution, i.e. $U \sim \Gamma(\alpha_1, \alpha_2)$, then its likelihood will be denoted by $p(u\,;\,\alpha_1, \alpha_2)$. The semi-colon "$;$" separates the random variables from the parameters. The lower-case letter argument always refers to the corresponding upper-case variable, i.e. $p(x)$ is the likelihood of X. When considering a conditional distribution, the variable are separated from the conditioning by a vertical line "$\,|\,$". For instance, if we consider λ as random, the conditional distribution associated to $X \,|\, \lambda \sim \mathscr{P}(\lambda)$ is $p(x \,|\, \lambda)$. A conditional distribution may also depend on some parameters. In this case, the notations are mixed. For example, let U be random and v a parameter, we define the conditional distribution of X as a Poisson distribution of parameters Uv, i.e. $\mathscr{P}(Uv)$, then the conditional likelihood of X is denoted by $p(x \,|\, U\,;\,v)$.

Expectation of random variables
We define the different notations that we will use when taking the expectation of random variables. We consider a hierarchical model where the conditional distribution of the data X depends on two random variables U and V, i.e. defined by $p(x \,|\, U, V)$. The prior on U is defined by $p(u\,;\,\alpha)$ and the prior on V by $p(v\,;\,\beta)$. The expectation of X regarding its conditional distribution is defined as $\mathbb{E}[X \,|\, U, V]$. The expectation of the marginal distribution of X, i.e. $p(x\,;\alpha, \beta)$ where the conditioning variables are integrated out, is $\mathbb{E}[X]$. The expectations of the U and V regarding their prior are respectively $\mathbb{E}[U]$ and $\mathbb{E}[V]$. The expectations of U and V regarding some probability distribution q are respectively $\mathbb{E}_q[U]$ and $\mathbb{E}_q[V]$. Eventually, the expectation of the posterior distribution of U and V, i.e. regarding $p(u \,|\, X)$ and $p(v \,|\, X)$ respectively, are denoted by $\mathbb{E}_{U|X}[U]$ and $\mathbb{E}_{V|X}[V]$ respectively. Sometimes, in order to have lighter notation, they may be denoted by $\mathbb{E}[U \,|\, X]$ and $\mathbb{E}[V \,|\, X]$ respectively.

5.1 Data-specific factorization of matrix

We now focus on model-based approaches and review the state-of-the-art regarding data-driven factorization of matrix, from generalized Principal Component Analysis (PCA) to Non-negative Matrix Factorization (NMF). We first identify potential issues when applying the PCA to non-Gaussian data. Then, we present the general framework of matrix factorization for count data and introduce some statistical and algorithmic considerations that will drive the development of our own approach.

5.1.1 Matrix factorization and non-Gaussian data

In exploratory statistics, the objective is to explore and represent the organization of the data, both in the observation and in the variable spaces. To do so, a strategy is to understand the geometry of the data, for instance to assess which individuals are closed or distant from each other. Another option is to apprehend how the data are distributed, i.e. to choose and infer a statistical model fitting the data. Although these two strategies do not appear to be related at first sight, the frontier between these two worlds is thin, especially in the case of the PCA (De Leuuw, 1986).

The PCA is defined as a geometrical method that seeks for orthogonal directions explaining most of the observed variability. It is a linear projection of the data and does not presume any assumption on the distribution of the data. As seen in Chapter 4, it can be interpreted as a least squares approximation thanks to the Singular Value Decomposition (SVD). The standard geometry is based on the Euclidean distance (ℓ_2 metric). In our framework, we want to find a matrix product[1] $\mathbf{U}\mathbf{V}^T$ that is the closest to the data matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$. The interest of the Gaussian distribution is that, in the case of homoskedastic multivariate Gaussian data, the geometric approach exactly corresponds to the model-based approach, since the likelihood associated to such data is the least squares approximation criterion $\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2$, where the Frobenius distance $\|\cdot\|_F$ is the ℓ_2 metric between matrices in the Euclidean space. The main question in this context is to determine if the Euclidean geometry is appropriate for non-Gaussian data, especially when heteroskedasticity is involved. In particular, Bailey (2012) or Han & Liu (2013) developed an approach to extend PCA in the case of the heteroskedastic data. More generally, the underlying geometry that drives count data may be different than the ones in the Gaussian case (this point will be illustrated in the following).

The geometric question is related to a question about the model. Although, the co-

[1] with $\mathbf{U} \in \mathbb{R}^{n\times K}$ and $\mathbf{V} \in \mathbb{R}^{p\times K}$

variance maximization problem in PCA is defined for any underlying distributions, it is highly associated with the Gaussian distribution. Indeed, the PCA only considers the mean (through the centering of the data) and the empirical covariance that are exactly the two sufficient statistics characterizing the multivariate Gaussian distribution. Therefore, considering a covariance criterion for non-Gaussian data may not be appropriate to characterize the data. Actually, the covariance (and therefore correlation) is a measure of independence in the Gaussian framework as non-correlation equals independence for multivariate Gaussian variables. This statement is not true anymore when considering any other distribution. In particular, the absence of correlation does not mean anything regarding the independence of non-Gaussian variables, hence the PCA may miss higher-order relations between non-Gaussian variables. For instance, the Independent Component Analysis (ICA) by Comon (1994) is a dimension reduction approach that searches for linear combinations of independent latent factors that fit the data, i.e. $x_{ij} \approx \sum_k u_{ik}\, v_{jk}$, based on higher-order statistics. Hyvärinen & Oja (2000) reviewed the different ICA algorithms. This approach is particularly suitable for non-Gaussian yet continuous data. To finally illustrate the link between Gaussian distribution and PCA, the reader may refer to the work of Linsker (1988) and Geiger & Kubin (2013) that studied the optimality of PCA regarding information theory. In the Gaussian case, the PCA appears to minimize the information loss in the dimension reduction process.

In the following, we will focus on matrix factorization approach that are suitable for count data. In particular, the NMF was developed to handle positive data. It consists in searching for factor matrices $\mathbf{U}$ and $\mathbf{V}$ with non-negative entries so that a positive data x_{ij} is decomposed as a sum of positive factors $\sum_k u_{ik}\, v_{jk}$. For instance, the SVD may lead to a decomposition with negative entries in $\mathbf{U}\mathbf{V}^T$ which cannot be a good approximation of the non-negative data $\mathbf{X}$. The first NMF method was proposed by Paatero (1997) and popularized by Lee & Seung (1999, 2001). It is nonetheless still based on the least squares problem but with non-negativity constraints on $\mathbf{U}$ and $\mathbf{V}$:

$$\begin{cases} \underset{\mathbf{U}\in\mathbb{R}^{n\times K},\mathbf{V}\in\mathbb{R}^{p\times K}}{\operatorname{argmin}} \ \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2\,, \\ \mathbf{U}\geq 0, \mathbf{V}\geq 0\,, \end{cases} \tag{5.1}$$

where $\mathbf{U} \geq 0$ and $\mathbf{V} \geq 0$ denote the non-negativity constraints[2]. Lee & Seung (2001) or Kim & Park (2007) proposed some optimization procedures for this criterion. The interest of such method that we denote by Least Squares NMF (ls-NMF) stands when deriving theoretical properties of the matrix factorization such as identifiability of the factors (Donoho & Stodden, 2003), complexity of the exact factorization (Vavasis, 2009) or uniqueness of the decomposition (Laurberg et al., 2008). However, it does not eman-

[2] Matrices with positive entries were specifically called “non-negative” because the concept of positive matrix has a different meaning in algebra.

cipate from the Euclidean geometry, therefore model-based NMF procedures were then introduced.

5.1.2 Poisson Non-negative Matrix Factorization

The Poisson-NMF introduced by Lee & Seung (1999) was the first approach that considered matrix factorization in the specific context of count data. It was motivated by applications in text mining or image analysis. It assumes the data to follow a Poisson model, i.e. each observation x_{ij} in the data matrix $\mathbf{X}$ is considered to be the realization of a Poisson random variable $X_{ij} \sim \mathscr{P}(\lambda_{ij})$. Instead of directly factorizing $\mathbf{X}$, the matrix of Poisson rates $\mathbf{\Lambda} = [\lambda_{ij}] \in (\mathbb{R}^{+})^{n \times p}$ is factorized as $\mathbf{\Lambda} = \mathbf{U}\mathbf{V}^T$. This corresponds to factorizing the expectation[3] $\mathbb{E}[\mathbf{X} \,|\, \mathbf{\Lambda}]$ of the matrix $\mathbf{X}$. Such formulation is similar to the path from linear regression to Generalized Linear Models, where the linear combination of predictors explains the expectation of the response, instead of the response directly (c.f. Chapter 1). In this regard, the concept of Generalized PCA (GPCA) was introduced to extend the PCA to the exponential family. If we consider the observations and variables to be independent, the log-likelihood of such Poisson model is therefore (when disrupting constant terms):

$$\log p(\mathbf{X}; \mathbf{U}, \mathbf{V}) = \sum_{i=1}^{n} \sum_{j=1}^{p} x_{ij} \log(\mathbf{u}_i^T \mathbf{v}_j) - \mathbf{u}_i^T \mathbf{v}_j \,, \tag{5.2}$$

where $\mathbf{u}_i \in \mathbb{R}^K$ and $\mathbf{v}_j \in \mathbb{R}^K$ represent the respective rows of $\mathbf{U}$ and $\mathbf{V}$, and then $\lambda_{ij} = \sum_k u_{ik}\, v_{jk} = \mathbf{u}_i^T \mathbf{v}_j$.

The Poisson-NMF is a dimension reduction method since the Poisson rates λ_{ij} are not directly estimated. The procedure estimates $\mathbf{U}$ and $\mathbf{V}$ which corresponds to $n \times K + p \times K$ parameters. As mentioned in Chapter 4, $n \times K + p \times K \ll n \times p$ when $K \ll \min(n, p)$, hence avoiding over-parametrization.

The factor matrices $\mathbf{U}$ and $\mathbf{V}$ are estimated by the Maximum Likelihood Estimation (MLE) procedure under non-negativity constraints, so that the rate matrix $\mathbf{\Lambda}$ remains non-negative. Lee & Seung (1999, 2001) proposed iterative optimization procedures that lead to a local optimum of the log-likelihood (5.2). In order to avoid the non-negativity constraint, Salmon et al. (2014) have proposed a formulation using the exponential function $X_{ij} \sim \mathscr{P}\big(\exp(\sum_k u_{ik}\, v_{jk})\big)$, however the optimization scheme seems to be less straightforward.

[3] In this matrix notation, the entries of $\mathbb{E}[\mathbf{X} \,|\, \mathbf{\Lambda}]$ are defined as $\mathbb{E}[X_{ij} \,|\, \lambda_{ij}]$ for any pair (i, j).

At this point, it has to be noted that the Poisson-NMF has a geometric interpretation. Sra & Dhillon (2005) or Févotte & Cemgil (2009) reviewed different statistical models for NMF that specifically correspond to different metric and thus different underlying geometry. In particular, in the exponential family, the log-likelihood can be interpreted as a Bregman divergence (Chen et al., 2008) between the data and the parameters of the data distribution. For instance, the metric suitable for Poisson data $X_{ij} \sim \mathscr{P}(\lambda_{ij})$ is not the Euclidean distance but rather a generalized Kullback-Leibler divergence, based on Bregman divergence, defined as:

$$D(\mathbf{X} \,|\, \boldsymbol{\lambda}) = \sum_{i=1}^{n} \sum_{j=1}^{p} x_{ij} \log\left(\frac{x_{ij}}{\lambda_{ij}}\right) - x_{ij} + \lambda_{ij} \,. \tag{5.3}$$

In this context, the Poisson-NMF finds the factors $\mathbf{U}$ and $\mathbf{V}$ that minimize the divergence between $\mathbf{X}$ and $\mathbf{U}\mathbf{V}^T$, i.e.

$$\begin{cases} \underset{\mathbf{U}\in\mathbb{R}^{n\times K}, \mathbf{V}\in\mathbb{R}^{p\times K}}{\text{argmin}} \quad D(\mathbf{X} \,|\, \mathbf{U}\mathbf{V}^T)\,, \\ \mathbf{U} \geq 0, \mathbf{V} \geq 0\,. \end{cases} \tag{5.4}$$

The Bregman divergence can be viewed as a generalization of the Euclidean metric to the exponential family (Banerjee et al., 2005).

The Poisson-NMF also lies in the framework of the GPCA proposed by Collins et al. (2001). As mentioned previously, based on the formulation of Generalized Linear Models (GLMs) in the case of non-Gaussian data (c.f. Part I), the PCA is generalized following the scheme of Poisson-NMF but for any distribution in the exponential family. This approach uses the Bregman divergence to quantify the proximity between $\mathbf{X}$ and $\mathbf{U}\mathbf{V}^T$. One of the main interest of such formulation is that it is suitable for any type of data. On the contrary, when using the least squares approximation $\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2$, the estimated matrix $\mathbf{U}\mathbf{V}^T$ is not guaranteed to satisfy the constraints defining X_{ij}, e.g. a positive count in our case. The GPCA can be interpreted as a GLM linking X_{ij} to the latent factors u_{ik} and v_{jk}. This formulation is related to the factor models Bartholomew (2004). In particular, the relation between factor models and matrix factorization have been investigated in the Gaussian case by Tipping & Bishop (1999) or in the Poisson case by Lee et al. (2013).

5.1.3 Gamma-Poisson matrix factorization

As presented in Chapter 4, we will focus on the Negative Binomial distribution in order to get a model that is more flexible and that accounts for over-dispersion (on the contrary to the Poisson model). If we consider a Negative Binomial matrix factorization, it is based on the model $X_{ij} \sim \mathcal{NB}(r_{ij}, \pi_{ij})$ with the parameters[4] $r_{ij} > 0$ and $\pi_{ij} \in (0, 1)$. It is necessary to define how to factorize the two sets of parameters $[r_{ij}] \in \mathbb{R}^{n\times p}$ and $[\pi_{ij}] \in \mathbb{R}^{n\times p}$, and especially to set the relation between the factors in each set of parameters. Instead, we consider an extension of the Gamma-Poisson (GaP) model introduced in Chapter 4 to the context of matrix factorization.

In the following, the matrix notations $\mathbf{X}$, $\mathbf{U}$ and $\mathbf{V}$ may refer, depending on the context, to the matrices of effective realizations, i.e. $[x_{ij}]_{n\times p}$, $[u_{ik}]_{n\times K}$ and $[v_{jk}]_{p\times K}$ respectively, or to the collection of associated random variables, i.e $[X_{ij}]_{n\times p}$, $[U_{ik}]_{n\times K}$ and $[V_{jk}]_{p\times K}$ respectively.

The Poisson model of NMF is extended as follows. The factors U_{ik} and V_{jk} are viewed as independent random variables with prior distributions. The set of parameters, denoted by $\mathbf{\Omega}$, of these priors are now the hyper-parameters of the model. We will therefore consider the Gamma-Poisson factor model. Canny (2004) or Buntine & Jakulin (2006) proposed a GaP factor model with Gamma priors on only one of the factors $\mathbf{U}$ or $\mathbf{V}$. We will rather consider a GaP factor model with Gamma priors on both factors $\mathbf{U}$ and $\mathbf{V}$ (Cemgil, 2009) that is defined as follows:

$$\begin{aligned} & X_{ij} \,|\, (U_{ik}, V_{jk})_{k=1:K} \;\sim\; \mathscr{P}(\textstyle\sum_k U_{ik}\, V_{jk})\,, \\ & U_{ik} \sim \Gamma(\alpha_{k,1}, \alpha_{k,2})\,, \\ & V_{jk} \sim \Gamma(\beta_{k,1}, \beta_{k,2})\,, \end{aligned} \tag{5.5}$$

with the hyper-parameters $\alpha_{k,1}, \alpha_{k,2}, \beta_{k,1}, \beta_{k,2} > 0$. The observations X_{ij} are assumed to be conditionally independent and the factors U_{ik} and V_{jk} to be independent. At this point, we introduce the following notations: the vectors $\boldsymbol{\alpha}_k$ and $\boldsymbol{\beta}_k$ ($k = 1, \dots, K$) store the corresponding prior hyper-parameters, i.e. $\boldsymbol{\alpha}_k = (\alpha_{k,1}, \alpha_{k,2})^T$ and $\boldsymbol{\beta}_k = (\beta_{k,1}, \beta_{k,2})^T$ respectively. The hyper-parameters of the prior over $\mathbf{U}$ and $\mathbf{V}$ are respectively gathered in $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_k] \in \mathbb{R}^{K\times 2}$ and $\boldsymbol{\beta} = [\boldsymbol{\beta}_k] \in \mathbb{R}^{K\times 2}$. The whole set of hyper-parameters is denoted as $\mathbf{\Omega} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$.

The interest to use prior distributions is to consider a refined model that fits the data more closely. Especially, the recorded variables in $\mathbf{X}$ are not supposed to be marginally independent as in the standard NMF, rather conditionally independent. The structure of dependency is directly included in the model through the specification of the prior on

[4] c.f. Appendix Chapter C

the latent factors. A drawback is that the marginal distribution of X_{ij} is intractable. Indeed, it would require to integrate the joint likelihood[5] $p(\mathbf{X}, \mathbf{U}, \mathbf{V}\,;\, \boldsymbol{\alpha}, \boldsymbol{\beta})$ over $\mathbf{U}$ and $\mathbf{V}$. Following the notations introduced at the beginning of this chapter, the marginal is therefore:

$$p(\mathbf{X}\,;\, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \int_{(\mathbf{U},\mathbf{V})} p(\mathbf{X} \,|\, \mathbf{U}, \mathbf{V})\, p(\mathbf{U}, \mathbf{V}\,;\, \boldsymbol{\alpha}, \boldsymbol{\beta})\, \mathrm{d}\mathbf{U}\, \mathrm{d}\mathbf{V}\,.$$

However, the conditional distribution of X_{ij} depends on the Poisson rate $\sum_k u_{ik}\, v_{jk}$ as:

$$p\big(x_{ij} \,|\, (u_{ik}, v_{jk})_k\big) \;=\; \frac{\exp\big(-\sum_k u_{ik}\, v_{jk}\big)\big(\sum_k u_{ik}\, v_{jk}\big)^{x_{ij}}}{x_{ij}\,!}$$

The term $\big(\sum_k u_{ik}\, v_{jk}\big)^{x_{ij}}$ may not be factorized because it involves $\log\big(\sum_k u_{ik}\, v_{jk}\big)$ that is not expandable. This point is an issue when estimating the parameters of the model as we will see in a moment. Moreover, the distribution of such multiplicative/additive combination of Gamma distributed random variable does not admit a distribution with a closed-form density (Moschopoulos, 1985; Coelho & Arnold, 2014). Nonetheless, it can be proven that such GaP factor model accounts for over-dispersion in the data (c.f. Appendix Section D.1.1).

The interest of the GaP factor model can be summarized as follows:

- It is a dimension reduction method that is suitable to handle over-dispersed data.
- Following the concept of matrix factorization, it exposes the latent structure within the data, especially in the observation and in the variable spaces. This point illustrates the concept of "count-specific PCA".
- The dependency between variables X_{ij} is integrated in the model, on the contrary to SVD or NMF-based approaches.
- The model can be adapted to account for zero-inflation or to enforce sparsity in the factors (this point will be discussed in Chapter 6).

We will especially derive a matrix factorization method based on the GaP factor model. A wide variety of such model have been developed in the literature, with different strategies regarding the statistical inference, see Canny (2004), Cemgil (2009), Hoffman et al. (2010), Dikmen & Févotte (2012), Zhou et al. (2012), Gopalan et al. (2014), Paisley et al. (2014) or Acharya et al. (2015). It can be noted that other GaP models were also proposed without an explicit formulation depending on latent factors[6]. Dunson &

[5] Also known as the complete likelihood. It is explicitly formulated in Appendix Section D.1.2.

[6] The algorithm from Dunson & Herring (2005) and Titsias (2008) have inspired some of the references concerning the GaP factor model, especially regarding the inference algorithms.

Herring (2005) presented a more generalized GaP model with Gamma latent variables and a regression term involving some covariates. Titsias (2008) worked on an infinite additive GaP model, i.e. the Poisson rate is an infinite sum of Gamma distributed latent variables, but without considering a latent factor model. In the next sections, we will detail the different approaches regarding the inference in the Gamma-Poisson factor model and we will explain the novelty of our procedure.

5.2 Inference in the Gamma-Poisson factor model

When considering a hierarchical model such as the GaP factor model, we have to choose between computing point estimates or inferring the distribution of $\mathbf{U}$ and $\mathbf{V}$. In this context, point estimation represent a snapshot of the latent variables in a particular data sample. On the contrary, the inference of the model would correspond to estimate the complete distribution of the factors $\mathbf{U}$ and $\mathbf{V}$. For instance, inferring the posterior, i.e. the distribution conditionally to the data, gives an insight on the behavior of the latent variables at the population level. In this section, we will discuss different strategies that were proposed in the context of the GaP factor model. In particular, some approaches estimate the factors with a MLE or a Maximum a Posteriori (MAP) procedure. We recall that the MLE is the mode of the marginal likelihood, as in the Poisson-NMF algorithm, while the MAP corresponds to the mode of the posterior regarding $\mathbf{U}$ and $\mathbf{V}$ (c.f. box below). Other algorithms directly infer the posterior of $\mathbf{U}$ and $\mathbf{V}$. In practice, our method will be based on a computationally efficient approach that is intermediate between point estimation and posterior inference.

Estimation and inference
In a model $p(x\,;\,\theta)$ with a parameter θ, the MLE is the most probable θ that explains the data:

$$\widehat{\theta}_{\mathrm{MLE}}(x) = \operatorname*{argmax}_{\theta}\, p(x\,;\,\theta)$$

If we consider a prior $p(\theta)$ over θ, the marginal distribution of the observations x becomes $p(x) = \int_\theta p(x\,|\,\theta)\,p(\theta)\,\mathrm{d}\theta$. Thanks to the Bayes rule, the posterior distribution is:

$$p(\theta\,|\,x) = \frac{p(x,\theta)}{p(x)} = \frac{p(x\,|\,\theta)\,p(\theta)}{\int_\vartheta p(x\,|\,\vartheta)\,p(\vartheta)\,\mathrm{d}\vartheta} \propto p(x\,|\,\theta)\,p(\theta)$$

The MAP is the most probable θ knowing the data. The marginal distribution of the observations $p(x)$ is independent from the parameter θ, hence the MAP estimation corresponds to maximizing the joint likelihood $p(x,\theta) = p(x\,|\,\theta)\,p(\theta)$:

$$\widehat{\theta}_{\mathrm{MAP}}(x) = \operatorname*{argmax}_{\theta}\, \frac{p(x\,|\,\theta)\,p(\theta)}{\int_\vartheta p(x\,|\,\vartheta)\,p(\vartheta)\,\mathrm{d}\vartheta} = \operatorname*{argmax}_{\theta}\, p(x\,|\,\theta)\,p(\theta)$$

The MLE and the MAP are point estimates. On the contrary, when inferring the posterior, the hyper-parameter α of the prior $p(\theta\,;\,\alpha)$ is accounted for. Thanks to the Bayes formula, the posterior can be written:

$$p(\theta\,|\,x,\alpha) = \frac{p(x\,|\,\theta)\,p(\theta\,;\,\alpha)}{p(x\,;\,\alpha)}$$

In this context, the aim of Bayesian inference is to infer the posterior $p(\theta\,|\,x,\alpha)$ and not just get a point estimation of θ.

5.2.1 Point estimation?

In the GaP factor model, the first interest is to estimate the factors $\mathbf{U}$ and $\mathbf{V}$. As the model considers some prior over the latent factors, the objective is to find the MAP estimation of $\mathbf{U}$ and $\mathbf{V}$, i.e. the mode of the joint likelihood $p(\mathbf{X},\mathbf{U},\mathbf{V})$. However, since the log of a sum is not expandable, the main issue when considering such model is the term $\log\left(\sum_k u_{ik}\,v_{jk}\right)$ that appears in the conditional distribution[7] of X_{ij}. It is especially a pitfall for differentiation in the case of direct optimization, or for integration when considering the Expectation-Maximization (EM) algorithm (see Dempster et al. (1977) or next box).

[7] c.f. Section 5.1.3 and Appendix Chapter D

EM algorithm
In a model with some data $\mathbf{X}$, some latent variables $\mathbf{Z}$ whose prior depends on some hyper-parameters $\boldsymbol{\Omega}$, the EM algorithm maximizes the expectation of $\log p(\mathbf{X}, \mathbf{Z} \,;\, \boldsymbol{\Omega})$ regarding the posterior, i.e.:

$$\underset{\boldsymbol{\Omega}}{\operatorname{argmax}}\ \mathbb{E}_{\mathbf{Z}\,|\,\mathbf{X}\,;\,\widetilde{\boldsymbol{\Omega}}}[\log p(\mathbf{X}, \mathbf{Z} \,;\, \boldsymbol{\Omega})]$$

where $\widetilde{\boldsymbol{\Omega}}$ are some fixed values for the hyper-parameters. Indeed, it can be proven that the optimal point $\widehat{\boldsymbol{\Omega}}$ that verifies $\nabla_{\boldsymbol{\Omega}}\ \mathbb{E}[p(\mathbf{X}, \mathbf{Z} \,;\, \boldsymbol{\Omega})|\mathbf{X}] = 0$ also verifies $\nabla_{\boldsymbol{\Omega}}\ p(\mathbf{X} \,;\, \boldsymbol{\Omega}) = 0$ (McLachlan & Krishnan, 2008), which corresponds to the MLE. The optimization is iterative and each iteration is divided into two steps. The E-step consists in computing the expectation of the joint log-likelihood regarding the posterior, i.e. $\mathbb{E}_{\mathbf{Z}\,|\,\mathbf{X}\,;\,\widetilde{\boldsymbol{\Omega}}}[\log p(\mathbf{X}, \mathbf{Z} \,;\, \boldsymbol{\Omega})]$ that is maximized in the M-step. The EM algorithm can be modified to estimate the MAP (McLachlan & Krishnan, 2008) by maximzing $\mathbb{E}_{\mathbf{Z}\,|\,\mathbf{X}\,;\,\widetilde{\boldsymbol{\Omega}}}[\log p(\mathbf{X}, \mathbf{Z} \,;\, \boldsymbol{\Omega})] + \log p(\mathbf{Z})$ in the M-step.

In order to overcome the computational issue, Canny (2004) defined a simplified GaP factor model. He only set the Gamma prior on the factor $\mathbf{V}$, i.e. on the variable contribution $V_{jk} \sim \Gamma(\beta_{k,1}, \beta_{k,2})$. The entries u_{ik} in the factor $\mathbf{U}$ are considered as hyper-parameters of the model. The joint log-likelihood of this model is therefore:

$$\log p(\mathbf{X}, \mathbf{V} \,;\, \mathbf{U}, \boldsymbol{\beta}) = \log p(\mathbf{X} \,|\, \mathbf{V} \,;\, \mathbf{U}) + \log p(\mathbf{V} \,;\, \boldsymbol{\beta})\,,$$

depending on the hyper-parameters $\mathbf{U}$ and $\boldsymbol{\beta}$. In the E-step of his EM algorithm, Canny used a sharp multiplicative approximation that allows to expand the term $\mathbb{E}\big[\log\big(\sum_k u_{ik}\, V_{jk}\big) \,|\, X_{ij}\big]$ based on $\mathbb{E}[V_{jk} \,|\, X_{ij}]$ and the Taylor-Young development of the function $x \mapsto \log(x+1)$. This trick leads to an explicit M-step to estimate u_{ik}. The MAP for V_{jk} is computed on the fly during the E-step.

The extension of this trick to the GaP factor model defined in Equation (5.5) is however not possible because the term $\mathbb{E}\big[\log\big(\sum_k U_{ik}\, V_{jk}\big) \,|\, X_{ij}\big]$ depends on both $\mathbb{E}[U_{ik} \,|\, X_{ij}]$ and $\mathbb{E}[V_{jk} \,|\, X_{ij}]$. Especially, it is not sure that $\mathbb{E}[U_{ik}\, V_{jk} \,|\, X_{ij}]$ can be factorized in the product of both posterior expectations. More generally, the EM algorithm cannot be used to find the MAP for the factors $\mathbf{U}$ and $\mathbf{V}$ or at least the MLE for the hyper-parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in the GaP factor model because the E-step depends on the following moments[8]:

$$\begin{aligned}
&\mathbb{E}[U_{ik} \,|\, X_{ij}]\,,\ \mathbb{E}[V_{jk} \,|\, X_{ij}]\,,\\
&\mathbb{E}[\log(U_{ik}) \,|\, X_{ij}]\,,\ \mathbb{E}[\log(V_{jk}) \,|\, X_{ij}]\,,\\
&\mathbb{E}\big[\log\big(\textstyle\sum_k U_{ik}\, V_{jk}\big) \,|\, X_{ij}\big]\,,
\end{aligned}$$

that cannot be derived for similar reasons. Other methodologies were proposed to overcome this issue.

[8] c.f. Appendix Chapter D

5.2.2 Posterior inference?

Another strategy would be to infer the posterior and consider $\mathbb{E}[U_{ik} \mid X_{ij}]$ and $\mathbb{E}[V_{jk} \mid X_{ij}]$ to estimate U_{ik} and V_{jk}. However, determining the posterior of the latent factors in the GaP factor model is not straightforward.

If a model is defined with conjugate prior[9] (Diaconis et al., 1979; Fink, 1997), the prior and posterior lie in the same class of distributions in the exponential family. For instance, the Gamma distribution is the conjugate prior to the Poisson distribution. Therefore, if the Poisson rate λ_{ij} follows a Gamma prior, its posterior will also be a Gamma distribution whose parameters depend on the hyper-parameters of the prior and on the observations X_{ij}. However, in the GaP factor model, the conjugacy relation is not verified as the combination $\lambda_{ij} = \sum_k U_{ik} V_{jk}$ does not follow an explicit distribution (c.f. previously). Therefore, the posterior of U_{ik} and V_{jk} cannot be determined through the conjugacy relation between the Gamma and the Poisson distributions. Moreover, it cannot be directly derived either, because, as previously mentioned, it would require to integrate the joint likelihood $p(\mathbf{X}, \mathbf{U}, \mathbf{V})$ over $\mathbf{U}$ and $\mathbf{V}$. In general, the potential non-existence of a closed-form relation mapping from the data to the posterior is a crucial issue (Orbanz, 2009).

In this context, it is necessary to use Markov Chain Monte Carlo (MCMC) approaches to infer the posterior (c.f. box below).

MCMC framework
MCMC is used to draw sample from a distribution for which it is not possible to directly sample or generate realizations. The principle is to iterate through a Markov Chain whose stationary distribution is the distribution of interest, i.e. the posterior in our inference problem. Observations are generated at each iteration, and as the iterations goes the distribution of the observations become closer to the objective distribution. If the Markov Chain is iterated far enough, it reaches its stationary distribution. MCMC procedures are the almost only way to sample exactly through the posterior.

Different approaches have been used in the context of the GaP factor model for matrix factorization. Cemgil (2009) and Dikmen & Févotte (2012) proposed a MCMC-based inference algorithm for the GaP factor model. They especially implemented a Gibbs sampling algorithm (Geman & Geman, 1984). In practice, Dikmen & Févotte considered a modified GaP factor model that we will introduce later. Zhou & Carin (2012) presented a unified framework for NMF, GaP factor model and Latent Dirichlet Allocation[10] that is

[9] Conjugate prior are based on conjugacy relationships in the exponential family (c.f. Appendix Chapter A)

[10] Matrix factorization method for categorical data Blei et al. (2003).

inspired from Dunson & Herring (2005). The inference in such framework is also based on MCMC, allowing to consider versatile distributions in the model. Acharya et al. (2015) even extended the GaP factor model to dynamic count matrices, thanks to an inference via Gibbs sampling. Eventually, it does not directly concern the GaP factor model but it can be noted that Schmidt et al. (2009) developed a Bayesian NMF procedure based on a Gaussian factor model (with exponential prior), again by using a MCMC algorithm to infer the posterior.

We will not detail any longer MCMC approaches as we will focus on another inference method. The reason of this choice will be discussed in the next section. In particular, it is linked to the convergence speed of Markov Chains. In the context of MCMC, it may require an important number of iterations to reach the stationary distributions.

5.2.3 Variational inference

The main issue when inferring posterior distributions is that the marginal likelihood of the data is not explicit. If we might be able to integrate:

$$\int_{(\mathbf{U},\mathbf{V})} p(\mathbf{X} \,|\, \mathbf{U},\mathbf{V})\, p(\mathbf{U},\mathbf{V}\,;\,\boldsymbol{\Omega})\, \mathrm{d}\mathbf{U}\, \mathrm{d}\mathbf{V}\,,$$

the posterior $p(\mathbf{U},\mathbf{V} \,|\, \mathbf{X}\,;\,\boldsymbol{\Omega})$ would be explicit. We recall that $\boldsymbol{\Omega} = (\boldsymbol{\alpha},\boldsymbol{\beta})$ are the hyper-parameters of the model.

Variational inference is a framework that solves this issue based on an approximation of the posterior. It was introduced by Jaakkola & Jordan (1997) and Jordan et al. (1999). A complete presentation of this framework can be found in Hoffman et al. (2013) or Blei et al. (2016). We introduce variational inference in the context of the GaP factor model defined in Equation (5.5). The complete inference process will be detailed in Chapter 6.

The Evidence Lower Bound

In the GaP factor model, two sets of latent variables are considered, the global ones $\mathbf{V}$ (depending on the p variables) and the local ones $\mathbf{U}$ (depending on the n observations). The main purpose of variational inference is to approximate the posterior $p(\mathbf{U},\mathbf{V} \,|\, \mathbf{X}\,;\,\boldsymbol{\Omega})$ by a factorizable distribution denoted by $q(\mathbf{U},\mathbf{V})$ and called the variational distribution.

The Kullback-Leibler divergence measures the “difference” between two probability distributions, hence the variational distribution $q(\mathbf{U},\mathbf{V})$ can be defined as the “closest” distribution to the posterior $p(\mathbf{U},\mathbf{V}\,|\,\mathbf{X}\,;\,\mathbf{\Omega})$ regarding the Kullback-Leibler divergence (Hoffman et al., 2013):

$$q(\mathbf{U},\mathbf{V}) = \underset{\text{distribution } \widetilde{q}}{\operatorname{argmin}} \ \mathrm{KL}\big(\widetilde{q}(\mathbf{U},\mathbf{V})\,\big|\,p(\mathbf{U},\mathbf{V}\,|\,\mathbf{X}\,;\,\mathbf{\Omega})\big)\,, \tag{5.6}$$

where the Kullback-Leibler divergence is defined as:

$$\mathrm{KL}\big(\widetilde{q}(\mathbf{U},\mathbf{V})\,\big|\,p(\mathbf{U},\mathbf{V}\,|\,\mathbf{X}\,;\,\mathbf{\Omega})\big) = \mathbb{E}_{\widetilde{q}}[\log \widetilde{q}(\mathbf{U},\mathbf{V})] - \mathbb{E}_{\widetilde{q}}[\log p(\mathbf{U},\mathbf{V}\,|\,\mathbf{X}\,;\,\mathbf{\Omega})]\,,$$

for any probability distribution $\widetilde{q}$.

As the posterior is not explicit, the problem (5.6) is reformulated. In practice, minimizing $\mathrm{KL}\big(q(\mathbf{U},\mathbf{V})\,\big|\,p(\mathbf{U},\mathbf{V}\,|\,\mathbf{X}\,;\,\mathbf{\Omega})\big)$ regarding q is equivalent to maximizing a specific lower bound on the marginal log-likelihood[11]. This bound, namely the Evidence Lower Bound (ELBO), is based on the variational distribution q and defined as (c.f. box below):

$$J(q) = \mathbb{E}_q[\log p(\mathbf{X},\mathbf{U},\mathbf{V})] - \mathbb{E}_q[\log q(\mathbf{U},\mathbf{V})]\,. \tag{5.7}$$

The interest is that maximizing the ELBO is equivalent to minimizing the Kullback-Leibler divergence between $q(\mathbf{U},\mathbf{V})$ and $p(\mathbf{U},\mathbf{V}\,|\,\mathbf{X})$ regarding q because:

$$J(q) = \ \ \log p(\mathbf{X}\,;\,\mathbf{\Omega}) - \mathrm{KL}\big(q(\mathbf{U},\mathbf{V})\,\big|\,p(\mathbf{U},\mathbf{V}\,|\,\mathbf{X}\,;\,\mathbf{\Omega})\big)\,,$$

where the marginal $p(\mathbf{X}\,;\,\mathbf{\Omega})$ is constant regarding q.

Derivation of the Evidence Lower Bound

For any distribution q, the marginal log-likelihood can be rewritten as:

$$\log p(\mathbf{X}\,;\,\mathbf{\Omega}) = \log \int_{(\mathbf{U},\mathbf{V})} p(\mathbf{X},\mathbf{U},\mathbf{V}\,;\,\mathbf{\Omega})\,\frac{q(\mathbf{U},\mathbf{V})}{q(\mathbf{U},\mathbf{V})}\,\mathrm{d}\mathbf{U}\,\mathrm{d}\mathbf{V}\,,$$

which is equivalent to the following formulation:

$$\log p(\mathbf{X}\,;\,\mathbf{\Omega}) = \ \ \log \mathbb{E}_q\left[\frac{p(\mathbf{X},\mathbf{U},\mathbf{V}\,;\,\mathbf{\Omega})}{q(\mathbf{U},\mathbf{V})}\right]\,.$$

The ELBO is derived by applying the Jensen's inequality on the log (which is concave):

$$\log p(\mathbf{X}\,;\,\mathbf{\Omega}) \geq \ \ \mathbb{E}_q\left[\log \frac{p(\mathbf{X},\mathbf{U},\mathbf{V}\,;\,\mathbf{\Omega})}{q(\mathbf{U},\mathbf{V})}\right]\,.$$

[11] The marginal likelihood is also called the evidence.

Mean-field variational family

To optimize the objective function $J(q)$, the distribution q is assumed to lie in the mean-field variational family (Hoffman et al., 2013), i.e. to be factorisable with independence between latent variables and between observations:

$$q(\mathbf{U}, \mathbf{V}) = \prod_{i=1}^{n} \prod_{k=1}^{K} q(u_{ik} \, ; \, \mathbf{a}_{ik}) \ \prod_{j=1}^{p} \prod_{k=1}^{K} q(v_{jk} \, ; \, \mathbf{b}_{jk})$$

where $\mathbf{a}_{ik}$ (resp. $\mathbf{b}_{jk}$) are the parameters of the variational distribution of U_{ik} (resp. V_{jk}).

The notations are heavy between the hyper-parameters of the model and the variational parameters. From now on, we apply the following convention, the model hyper-parameter are denoted by a Greek letter, and their corresponding variational parameter by the corresponding Roman letter. For instance, the Gamma prior on U_{ik} is parametrized by $\boldsymbol{\alpha}_k$, i.e. $p(u_{ik} \, ; \, \boldsymbol{\alpha}_k)$, and the corresponding variational distribution is $q(u_{ik} \, ; \, \mathbf{a}_{ik})$.

A second assumption states that each $q(u_{ik} \, ; \, \mathbf{a}_{ik})$ lies in the same exponential family as the complete conditional distribution on U_{ik}. Similarly, each $q(v_{jk} \, ; \, \mathbf{b}_{jk})$ is determined by the complete conditional on V_{jk}. The complete conditional of the latent variable U_{ik} is the conditional distributions of U_{ik} knowing the other latent variables and the data, it is denoted by $p(U_{ik} \,|\, —)$. Similarly, the complete conditional of V_{jk} is denoted by $p(V_{jk} \,|\, —)$.

Deriving the complete conditional is possible in our GaP factor model. Indeed, the complete conditional of U_{ik} is a Gamma distribution whose parameters $\boldsymbol{\eta}_{ik}(—) \in (\mathbb{R}^+)^2$. The "—" is here to recall that $\boldsymbol{\eta}_{ik}$ depends on the factor $\mathbf{V}$ and the data. Similarly, the complete conditional of V_{jk} is a Gamma distribution whose parameters $\boldsymbol{\eta}_{jk}(—) \in (\mathbb{R}^+)^2$ depends on the factor $\mathbf{U}$ and the data. The full derivation of the complete conditional $p(U_{ik} \,|\, —)$ and $p(V_{jk} \,|\, —)$ are joined in Appendix Section D.2.1. The proof is based on the conjugacy between the Poisson and the Gamma distribution. This property is characteristic of conditionally conjugate prior and such model are called conditionally conjugate model. As we will see, the complete conditional are also used to optimize the ELBO.

The variational distribution over $\mathbf{U}$ and $\mathbf{V}$ are therefore assumed to be Gamma distributed with parameters $\mathbf{a}_{ik} = (a_{ik,1}, a_{ik,2})$ and $\mathbf{b}_{jk} = (b_{jk,1}, b_{jk,2})$. The variational parameters regarding $\mathbf{U}$ and $\mathbf{V}$ are respectively stored in the following objects:

$$\mathbf{a} = [\mathbf{a}_{ik}] \in (\mathbb{R}^+)^{n \times K \times 2} \text{ and } \mathbf{b} = [\mathbf{b}_{ik}] \in (\mathbb{R}^+)^{p \times K \times 2} \, .$$

At this point, we have introduced many different distributions over U_{ik} and V_{jk} that are summarized in Table 5.1. To recap, the standard variational procedure is based on two assumptions:

- The variational distribution q is factorisable (independence).
- The variational distribution of each latent factor lies in the same exponential family as the complete conditional of this latent factor.

It can be noted that the validity of the mean field assumption and the quality of the variational approximation compared to the posterior are discussed in Blei et al. (2016).

Optimization algorithm

The two assumptions previously introduced are used in the optimization of the ELBO, defined in Equation (5.7), regarding the variational distribution q, i.e. regarding the set of variational parameters $(\mathbf{a}, \mathbf{b})$. Thanks to the formulation in the exponential family and by using the complete conditional distributions, it is possible to find the exact point $\mathbf{a}_{ik}$ and $\mathbf{b}_{jk}$ that set the gradient of the ELBO to zero (Hoffman et al., 2013). Indeed, when considering separately each parameter, here $\mathbf{a}_{ik}$ and $\mathbf{b}_{jk}$, the objective function respectively becomes:

$$\begin{aligned}\widetilde{J}(\mathbf{a}_{ik}) &= \mathbb{E}_q[\log p(U_{ik}\,|\,\text{—})] - \mathbb{E}[\log q(U_{ik}\,;\,\mathbf{a}_{ik})] + \text{const}\,,\\ \widetilde{J}(\mathbf{b}_{jk}) &= \mathbb{E}_q[\log p(V_{jk}\,|\,\text{—})] - \mathbb{E}[\log q(V_{jk}\,;\,\mathbf{b}_{jk})] + \text{const}\,,\end{aligned}$$

where “const” is a constant term. The complete conditional and the variational term are known (c.f. previously). Therefore, differentiating $J(q)$ regarding $\mathbf{a}_{ik}$ or $\mathbf{b}_{jk}$ corresponds to the explicit gradients $\nabla_{\mathbf{a}_{ik}}\widetilde{J}(\mathbf{a}_{ik})$ or $\nabla_{\mathbf{b}_{jk}}\widetilde{J}(\mathbf{b}_{jk})$ respectively. Based on some properties of the exponential family (Hoffman et al., 2013), the coordinates of the point in the space of variational parameters[12] that sets the gradient of $J(q)$ to zero can be explicitly derived[13]. In particular, this stationary point $(\mathbf{a}, \mathbf{b})$ depends on the parametrization of the complete conditional and the expectation of the latent variables regarding the variational distribution q, i.e.

$$\left.\begin{aligned}\mathbf{a}_{ik} &= \mathbb{E}_q[\boldsymbol{\eta}_{ik}(\text{—})]\\ \mathbf{b}_{jk} &= \mathbb{E}_q[\boldsymbol{\eta}_{jk}(\text{—})]\end{aligned}\right\}\ \text{verify}\ \nabla_{(\mathbf{a},\mathbf{b})}\,J(q) = 0\,.$$

Since the stationary point is known, an iterative optimization through a coordinate descent algorithm (Wright, 2015) may then be used to estimate the variational parameters. The complete formulation of the algorithm will be detailed in Chapter 6.

[12] We recall that the variational parameters are $(\mathbf{a}, \mathbf{b})$ with $\mathbf{a} \in (\mathbb{R}^+)^{n\times K\times 2}$ and $\mathbf{b} \in (\mathbb{R}^+)^{p\times K\times 2}$.

[13] c.f. Appendix Section D.2.2

Level	Distribution	Formulation	Independence
Model	Conditional	$X_{ij} \mid (U_{ik}, V_{jk})_k \sim \mathscr{P}(\sum_k U_{ik}\, V_{jk})$	Conditional
	Prior	$U_{ik} \sim \Gamma(\alpha_{k,1}, \alpha_{k,2})$ $V_{jk} \sim \Gamma(\beta_{k,1}, \beta_{k,2}$	✓
Optimization	Variational	$U_{ik} \overset{q}{\sim} \Gamma(a_{ik,1}, a_{ik,2})$ $V_{jk} \overset{q}{\sim} \Gamma(b_{jk,1}, b_{jk,2})$	✓
	Complete conditional	$U_{ik} \mid \text{—} \sim \Gamma\big(\eta_{ik,1}(\text{—}), \eta_{ik,2}(\text{—})\big)$ $V_{jk} \mid \text{—} \sim \Gamma\big(\eta_{jk,1}(\text{—}), \eta_{jk,2}(\text{—})\big)$	×

Table 5.1 – The different distributions that are defined in the model of for the need of the optimization procedures with their corresponding independence status. The notation $\overset{q}{\sim}$ refers to the variational distribution.

Variational inference in the Gamma-Poisson model

The variational inference inherently infers a distribution q that is an approximation of the true posterior of the latent factors. Therefore, the latent factors can be estimated by their expectation regarding the variational distribution, i.e. $\mathbb{E}_q[\mathbf{U}]$ and $\mathbb{E}_q[\mathbf{V}]$ respectively. These can be viewed as a proxy for the posterior expectation, i.e. $\mathbb{E}[\mathbf{U}\,|\,\mathbf{X}]$ and $\mathbb{E}[\mathbf{V}\,|\,\mathbf{X}]$ respectively. In this context, it has to be noted that $\mathbf{U}$ and $\mathbf{V}$ are not estimated as the mode of the variational distribution q that would correspond to an approximation of the MAP.

In the case of Gamma distribution[14], U_{ik} and V_{jk} are therefore respectively estimated by:

$$\widehat{U}_{ik} = \mathbb{E}_q[U_{ik}] = \frac{a_{ik,1}}{a_{ik,2}} \quad \text{and} \quad \widehat{V}_{jk} = \mathbb{E}_q[V_{jk}] = \frac{b_{jk,1}}{b_{jk,2}}\,.$$

The variational framework that we just introduced estimates the hyper-parameters $\mathbf{a}$ and $\mathbf{b}$ of the variational distribution q. As we will see in Chapter 6, the values of the stationary points $\mathbf{a}$ and $\mathbf{b}$ depend on the values of the hyper-parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ of the prior distributions set on $\mathbf{U}$ and $\mathbf{V}$ in the model. The problem here is that such hyper-parameters are not estimated in this framework and the values of the estimated $\mathbf{a}$ and $\mathbf{b}$ directly depend on the arbitrary initial values set to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, fixed or tuned by the user. Cemgil (2009), Hoffman et al. (2010), Gopalan et al. (2014) or Paisley et al. (2014) used this variational approach in their work regarding the GaP factor model. Zhou & Carin (2012) introduced a similar method in their unified framework of matrix factorization but without identifying it as variational inference.

Variational EM

In order to estimate the hyper-parameters $\boldsymbol{\Omega} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, the variational inference may also be used within the E-step of the EM algorithm. For instance, the expectation of the joint likelihood regarding the posterior is:

$$\mathbb{E}[p(\mathbf{X}, \mathbf{U}, \mathbf{V}\,;\,\boldsymbol{\Omega})|\mathbf{X}]\,.$$

It is not tractable in our model. However, it can be approximated by:

$$\mathbb{E}_q[p(\mathbf{X}, \mathbf{U}, \mathbf{V}\,;\,\boldsymbol{\Omega})]\,,$$

because the variational distribution q approximate the posterior. Such algorithm was introduced as the variational-EM algorithm (Beal & Ghahramani, 2003). The E-step

[14] In the parametrization of the Gamma distribution that we consider, the expectation of $U \sim \Gamma(\alpha_1, \alpha_2)$ is $\mathbb{E}[U] = \frac{\alpha_1}{\alpha_2}$.

consists in deriving the variational distribution approximating the posterior and integrating the joint log-likelihood regarding this variational distribution, while the M-step consists in maximizing $\mathbb{E}_q[p(\mathbf{X},\mathbf{U},\mathbf{V}\,;\,\boldsymbol{\Omega})]$ regarding the hyper-parameters $\boldsymbol{\Omega} = (\boldsymbol{\alpha},\boldsymbol{\beta})$. This approach produces an approximation of the posterior and a point estimation of the hyper-parameters.

It can be noted that the variational-EM is an alternative to the Monte Carlo EM (MCEM) algorithm (Wei & Tanner, 1990) or to the Stochastic Approximation of EM (SAEM) algorithm (Delyon et al., 1999). In particular, these two algorithms are also based on an approximate computation of the intractable integral $\mathbb{E}[p(\mathbf{X},\mathbf{U},\mathbf{V}\,;\,\boldsymbol{\Omega})|\mathbf{X}]$ at each iteration, either by a Monte Carlo method (MCEM) or by a stochastic averaging procedure (SAEM).

Dikmen & Févotte (2012) proposed to use the variational-EM algorithm in a modified GaP factor model. Their approach is based on the following assumptions. The factor $\mathbf{V}$ are supposed to be hyper-parameters of the model (without considering any Gamma prior). A Gamma prior is only set on the factor $\mathbf{U}$, i.e. $U_{ik} \sim \Gamma(\alpha_{k,1},\alpha_{k,2})$. Finally, the parameter $\alpha_{k,1}$ is fixed so that the shape of the Gamma distribution is set[15]. The posterior $p(\mathbf{U},|\,\mathbf{X}\,;\mathbf{V})$ is approximated by variational inference[16] in the E-step, and the objective $\mathbb{E}_q[p(\mathbf{X},\mathbf{U}\,;\,\mathbf{V},\boldsymbol{\alpha}_2)]$ is maximized regarding the factor $\mathbf{V}$ and the hyper-parameters $\boldsymbol{\alpha}_2 = (\alpha_{k,2})_k$ in the M-step. The interest of this formulation is to discard the inference of $\mathbf{U}$ because the number of parameters in $\mathbf{U}$ grows with the number of observations n.

5.3 Sparsity in matrix factorization

On the continuity of Part I, we will also consider variable selection in the context of matrix factorization. Following the definition of sparse PCA and sparse SVD, this setting relies on a hypothesis of parsimony. Only a portion of the variables are useful to explain the latent structure within the data. The objective is to simultaneously select these important variables and infer the underlying organization of the data. Based on this framework, the idea is to consider sparse columns in the matrix $\mathbf{V}$ that account for the variable contributions. Therefore, when the observations are decomposed as sparse linear combination of the factors, i.e. $x_{ij} \approx \sum_k u_{ik}\, v_{jk}$, the variables corresponding to null v_{jk} are discarded.

In the context of matrix factorization, there are two options to enforce sparsity on the columns $(\mathbf{v}_k)_{k=1:K}$ of the factor matrix $\mathbf{V}$. We may either add sparsity-inducing penalties

[15] c.f. Appendix Chapter A

[16] They also proposed a MCMC algorithm as previously stated.

on the vectors $\mathbf{v}_k$ in the optimization procedures, i.e. following the Lasso methodology Tibshirani (1996). We can either use sparsity-inducing prior and modify the model as in the Bayesian Lasso (Park & Casella, 2008), so that the selection is achieved directly by inferring the model. Both approach have been considered in the literature regarding matrix factorization. We will rather introduce sparsity-inducing priors in the GaP factor model.

5.3.1 Sparse NMF

We first recall the different methods for sparse NMF that are based on penalized criteria. Following the definition of the sparse PCA, it has been proposed to constrain the optimization problem behind the standard NMF to enforce sparsity in the columns $\mathbf{v}_k$ of $\mathbf{V}$. Such methodology follows the principle of the Lasso that consists in penalizing the ℓ_1-norm of the parameters. This penalty will shrink the coefficient of the non-pertinent variables to zero during the optimization.

The first penalized version of NMF was introduced by Hoyer (2002). His Non-Negative sparse Coding (NNSC) algorithm is based on the optimization of the least square criterion defining the ls-NMF with an ℓ_1 penalty on $\mathbf{v}_k$. Kim & Park (2007) then proposed a fast implementation of this framework. However, such approach is certainly suitable for non-negative data but not specifically designed to deal with count data, since the geometry induced by the Euclidean metric is not the most appropriate for count data. To overcome this issue, Liu et al. (2003) considered a penalized version of the Poisson-NMF formulated as a Bregman divergence minimization problem[17], i.e.

$$\text{argmin}_{\mathbf{U},\mathbf{V}}\, D(\mathbf{X} \,|\, \mathbf{U}\mathbf{V}^T) + \nu \textstyle\sum_k |\mathbf{v}_k|_1 \,,$$

with $\nu > 0$ and $D(\cdot\,|\,\cdot)$ defined in Equation (5.3). Their algorithm, called sparse NMF (SNMF), estimates sparse factor $\mathbf{v}_k$. The interest is that the Bregman divergence implies an underlying geometry suitable for count data.

The main sparse approaches for NMF are detailed in Table 5.2. They will be used for comparison when testing our algorithm. It can be noted that other approaches were also developed to impose sparsity in the ls-NMF. For instance, Eggert & Korner (2004) considered a ℓ_1-penalty on the factor $\mathbf{U}$. Hoyer (2004) proposed to use a penalty based on the ratio $|\mathbf{v}_k|_1/\|\mathbf{v}_k\|_2$. Eventually, Pascual-Montano et al. (2006) developed an approach that uses weighted matrix product between $\mathbf{U}$ and $\mathbf{V}$. However, we will not consider these procedures since they are based on the Euclidean distance.

[17] c.f. Section 5.1.2

Based on this remark, we will now introduce a framework for variable selection in hierarchical model, that we will extend to the GaP factor model.

Method	Loss	Sparse?	Optimization	Reference
ls-NMF	$\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2$	×	Gradient descent (multiplicative update rules)	Paatero (1997), Lee & Seung (2001), Brunet et al. (2004)
Poisson-NMF	$D(\mathbf{X} \,\|\, \mathbf{U}\mathbf{V}^T)$	×	Gradient descent (multiplicative update rules)	Lee & Seung (1999, 2001), Brunet et al. (2004)
NNSC	$\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \nu \sum_k \lvert\mathbf{v}_k\rvert_1$	✓	Projected gradient descent	Hoyer (2002)
SNMF	$D(\mathbf{X} \,\|\, \mathbf{U}\mathbf{V}^T) + \nu \sum_k \lvert\mathbf{v}_k\rvert_1$	✓	Projected gradient descent	Liu et al. (2003)
SNMF/R	$\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \nu_1 \sum_k \lvert\mathbf{v}_k\rvert_1$ $+ \nu_2 \|\mathbf{V}\|_F^2$	✓	Alternating least squares approach	Kim & Park (2007)

Table 5.2 – The different algorithms for NMF, including standard procedures and sparsity-inducing methods ($\nu, \nu_1, \nu_2 > 0$).

5.3.2 Spike-and-slab in Bayesian settings

The principle of Bayesian variable selection (Mitchell & Beauchamp, 1988; George & McCulloch, 1993) consists in setting specific prior that will enforce sparsity in the model. The reader may refer to O'Hara & Sillanpää (2009) for a complete review of the question. Such approaches have been widely studied in the case of linear regression and have been called spike-and-slab methods (Ishwaran & Rao, 2005).

To explain this framework, we extend it directly to our model. In the context of continuous probability distribution, a prior that generates true zeros with a non-null probability among the factors V_{jk} will be a two-group prior, defined as a mixture between a Dirac mass at zero δ_0 and the Gamma distribution $p(v_{jk}\,;\,\boldsymbol{\beta}_k)$, i.e.

$$V_{jk} \sim (1-\pi_k)\,\delta_0(v_{jk}) + \pi_k\, p(v_{jk}\,;\,\boldsymbol{\beta}_k)\,.$$

Parameter $\pi_k \in [0,1]$ regulates the balance between the mass at zero, i.e. the spike, and the other distribution, i.e. the slab (Malsiner-Walli & Wagner, 2011).

The issue with such formulation is that for a vector $\mathbf{v}_k$ there exist 2^p possible discrete configurations to distinguish between null and non-null coefficients. This point is a pitfall in the inference process. It induces tractability problems, especially with large p. To overcome this issue, it has been proposed to use continuous one-group priors with a huge mass around zero that play the role of the spike. In this context, the less relevant coefficient v_{jk} are not exactly set to zero but concentrated around zero. Malsiner-Walli & Wagner (2011) and Engelhardt & Adams (2014) reviewed the different spike-and-slab continuous prior (Gaussian-Inverse, Beta, Dirichlet, Laplace). Continuous spike-and-slab prior can be seen as a continuous relaxation of two-group sparsity inducing prior.

For instance, a famous application is the Bayesian Lasso (Park & Casella, 2008) that uses a Laplace prior (exponential distribution symmetric around zero) over the coefficients. In the case of Poisson model, Datta & Dunson (2015) proposed to use sparsity-inducing one-group prior for Poisson rate inference. We also cite Titsias & Lázaro-Gredilla (2011), Carbonetto et al. (2012) or Babacan et al. (2014) that use Bayesian variable selection and spike-and-slab two-group prior in variational inference. Our objective will be now to use a two-group prior to enforce sparsity over the columns of $\mathbf{V}$ in the GaP factor model.

In the context of hierarchical model for matrix factorization, the spike-and-slab approach have been used in different works, mainly in the Gaussian case, i.e. when the observation X_{ij} are supposed to follow a multivariate Gaussian distribution. Different studies were based on a one-group prior over the factors: Fevotte & Godsill (2006) and Lakshminarayanan et al. (2011) (both with $\mathcal{T}$-Student priors), Caron & Doucet (2008) (Gamma priors), Archambeau & Bach (2009) (Inverse-Gamma priors) and Gao & Engelhardt

(2012) (3-parameter Beta prior). On the contrary, other studies were based on two-group priors that exactly set the factors to zero. A series of works by Knowles & Ghahramani (2007), Knowles & Ghahramani (2011), Bhattacharya & Dunson (2011) and Shah et al. (2015) considered an infinite factor model[18] based on the least squares approximation (in the Gaussian framework). They considered Dirac-Gaussian mixture priors on the factor V_{jk} (or sometimes U_{ik}). In all these approaches, the inference was based on a MCMC procedure to infer the "sparsity" probability π_k. They also considered a Beta prior[19] on π_k.

Eventually, Gupta et al. (2012) introduced an infinite sparse GaP factor model, also based on the Beta-Bernoulli construction and inferred by MCMC. For our part, we will rather consider a Dirac-Gamma mixture prior on V_{jk} and infer the model in the variational framework.

5.4 Novelty for count matrix factorization in the Gamma-Poisson model

We will introduce a matrix factorization based on the GaP factor model in the next chapter. Our contributions may be summarized as follows:

- We consider a variational-EM algorithm to infer the latent factor $\mathbf{U}$ and $\mathbf{V}$ when considering Gamma prior both on U_{ik} and V_{jk}. The interest is to infer similarly the structure within the n observations and within the p variables without distinction. Such framework is moreover appropriate to handle over-dispersed count data.

- We will extend the GaP to handle zero-inflated data, by using Dirac-Poisson mixture distribution to account for the drop-out events (this point will be detailed in Chapter 6). Such framework has not been proposed yet in the context of data exploration with matrix factorization.

- The model will also be extended to impose sparsity among the columns of the factor $\mathbf{V}$ in order to select variables. We will use sparsity-inducing prior. To our knowledge, such an approach is new in the context of variational inference.

In order to highlight the historical evolution of some count matrix factorization methods, some of the different approaches based on Poisson or Gamma-Poisson factor models are

[18] with a theoretical infinite number of factors, in practice K is also estimated during the inference process.

[19] Such Beta-Bernoulli hierarchical model corresponds to an Indian Buffet Process, c.f. Griffiths & Ghahramani (2011).

summarized in Figure 5.1 as graphical models. It is possible to see the difference between the NMF, where the factors are considered as parameters, and the GaP factor model, which is based on a hierarchical construction with latent variables.

In our different GaP factor models, the inference will be based on the variational algorithms previously introduced. We choose the variational inference over MCMC approaches for the following reasons. Contrary to MCMC that potentially lead to an exact sampling in the posterior (if iterating far enough), variational inference is an approximation method. However, its main advantage over MCMC concerns the convergence speed. For instance, Nathoo et al. (2013) observed that the iterative optimization in variational methods are ~ 100 times faster to converge than MCMC methods. They considered linear regression problem and a data set with $n = 128$ observations and $p = 8196$ variables. This scale of data is consistent with what we may encounter in genomics. Dikmen & Févotte (2012) obtained the similar results in the matrix factorization problem on small data set ($n = p = 50$). Shen et al. (2010) also proposed a comparison of variational inference and MCMC with the same conclusion.

The point here is not to say that MCMC methods are useless and variational inference is the best option, but rather to motivate our choice for variational approaches. First, the standard methods in our context of matrix factorization, i.e. PCA, has a very low computational cost (Klema & Laub, 1980). It seems important that the alternative approaches that we propose and develop keep low computation time. Then, although high performance computing is on the rise, heavy computations still have a huge cost (in time and energy). The efficiency (regarding computation time) of variational inference is interesting regarding this point. Finally, developing an efficient MCMC procedure is a complete subject and we decided to focus on optimization-based approaches that suits more closely to the general topic of this PhD project.

In the next chapter, we will detail our matrix factorization method based on the GaP factor model that accounts for zero-inflation in the data and enforce sparsity among variables.

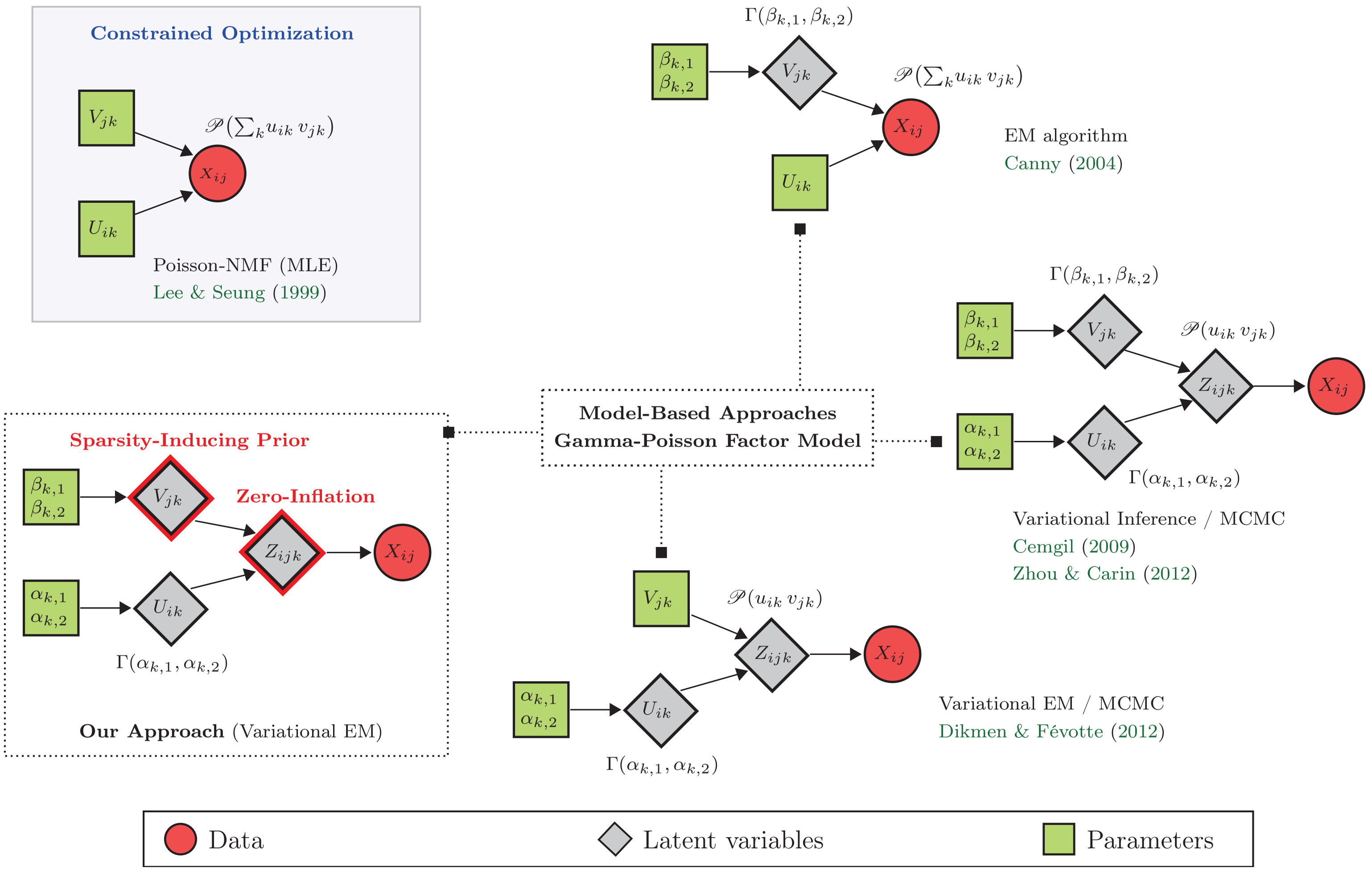

Constrained Optimization
V_{jk}
U_{ik}
$\mathscr{P}(\sum_k u_{ik}\, v_{jk})$
x_{ij}
Poisson-NMF (MLE)
Lee & Seung (1999)
$\Gamma(\beta_{k,1}, \beta_{k,2})$
$\beta_{k,1}$ $\beta_{k,2}$
V_{jk}
$\mathscr{P}(\sum_k u_{ik}\, v_{jk})$
X_{ij}
U_{ik}
EM algorithm
Canny (2004)
$\Gamma(\beta_{k,1}, \beta_{k,2})$
$\beta_{k,1}$ $\beta_{k,2}$
V_{jk}
$\mathscr{P}(u_{ik}\, v_{jk})$
Z_{ijk}
X_{ij}
$\alpha_{k,1}$ $\alpha_{k,2}$
U_{ik}
$\Gamma(\alpha_{k,1}, \alpha_{k,2})$
Variational Inference / MCMC
Cemgil (2009)
Zhou & Carin (2012)
Model-Based Approaches
Gamma-Poisson Factor Model
Sparsity-Inducing Prior
$\beta_{k,1}$ $\beta_{k,2}$
V_{jk}
Zero-Inflation
Z_{ijk}
X_{ij}
$\alpha_{k,1}$ $\alpha_{k,2}$
U_{ik}
$\Gamma(\alpha_{k,1}, \alpha_{k,2})$
Our Approach (Variational EM)
V_{jk}
$\mathscr{P}(u_{ik}\, v_{jk})$
Z_{ijk}
X_{ij}
$\alpha_{k,1}$ $\alpha_{k,2}$
U_{ik}
$\Gamma(\alpha_{k,1}, \alpha_{k,2})$
Variational EM / MCMC
Dikmen & Févotte (2012)
Data
Latent variables
Parameters

Figure 5.1 – Graphical models associated to different count matrix factorization approaches. The latent factors U_{ik} and V_{jk} are either considered as parameters or as latent variables. The latent variables Z_{ijk} will be defined in Chapter 6, they verify $Z_{ijk} \sim \mathscr{P}(u_{ik}\, v_{jk})$ so that $X_{ij} = \sum_k Z_{ijk}$ and are useful to derive the different variational algorithms.

Chapter 6

Count matrix factorization and single cell data analysis

In this chapter, we will specifically derive the inference algorithms suitable for different Gamma-Poisson (GaP) factor models that account for zero-inflation or that induce sparsity. We recall that our objective is to find a factorization $\mathbf{U}\mathbf{V}^T$ of the data matrix $\mathbf{X}$ when considering count data. We will first introduce our new variational-EM algorithm for the standard GaP factor model. Then, we will present two refinements of this model. On the one hand, the zero-inflated Gamma-Poisson factor model will account for zero-inflation in the data. On the other hand, the sparse Gamma-Poisson (sparse-GaP) factor model will impose sparsity among the factor $\mathbf{V}$, so that only the relevant variables will be selected to contribute to the latent directions.

We developed a new implementation of the variational-EM algorithm in both cases: zero-inflated (ZI) and sparse. It can be noted that Simchowitz (2013) proposed a zero-inflated Poisson matrix factorization method in a student project (that remains unpublished) where he considered a similar zero-inflated Gamma-Poisson factor model. However, his variational inference framework is slightly different from the one that we introduced. Concerning the sparse model, the use of variational inference in the case of spike-and-slab approach with two-group sparsity-inducing prior is also new since the previous approach by Gupta et al. (2012) was based on MCMC with an infinite number of factors.

The performance of our approach for dimension reduction and data exploration will be assessed on simulations. In particular, we will show the interest of such inference framework for data visualization and clustering, especially in the case of zero-inflated data. In this regard, we will discuss the robustness of variational inference regarding corrupted data. Then, we will illustrate the interest of the spike-and-slab approach regarding selection accuracy. Eventually, we will present an example of application on the single-cell data set introduced in Chapter 4.

Our variational framework for matrix factorization is implemented in a `R`-package, namely `CMF` for Count Matrix Factorization, that will be soon available on the `CRAN` (`https://cran.r-project.org/`). We dedicated time and efforts to implement our algorithms in `C++` to take advantage of the computational performance of this language. The package is based on `R` for input-output management and interface the `C++` code for computations. It is currently on a testing phase before the release of a stable version.

6.1 Implementation of the Gamma-Poisson factor model

In this section, we first briefly give more details about the variational inference algorithm for the GaP factor model. This framework will be useful in the following when considering more complex models (zero-inflated or sparse). Then, based on this formulation, we explicitly introduce the corresponding variational-EM algorithm that we developed.

6.1.1 Recap about the Gamma-Poisson model

We start by recalling the definition of the Gamma-Poisson factor model (Cemgil, 2009) that we will consider. In order to facilitate the computation, some third-party latent variables are introduced to quantify the decomposition of the count X_{ij} over the different latent directions. The set of latent variables $(Z_{ijk})_{k=1:K}$ for any fixed i and j is defined such that $X_{ij} = \sum_k Z_{ijk}$ where Z_{ijk} follows a conditional Poisson distribution, i.e. $Z_{ijk} \,|\, U_{ik}, V_{jk} \sim \mathscr{P}(U_{ik}\, V_{jk})$. Thus, the conditional distribution of X_{ij} remains $\mathscr{P}(\sum_k U_{ik}\, V_{jk})$ thanks to the additive property of the Poisson distribution. In addition, the variables Z_{ijk} are assumed to be conditionally independent. These quantities will be very useful when deriving the inference algorithm and are used in different approaches. To be complete, the factors U_{ik} and V_{jk} are assumed to be independent. Therefore, the complete GaP factor model can be summarized as:

$$
\begin{aligned}
&X_{ij} = \sum_k Z_{ijk}\,, \\
&Z_{ijk} \,|\, U_{ik}, V_{jk} \sim \mathscr{P}(U_{ik}\, V_{jk})\,, \\
&U_{ik} \sim \Gamma(\alpha_{k,1}, \alpha_{k,2})\,, \\
&V_{jk} \sim \Gamma(\beta_{k,1}, \beta_{k,2})\,.
\end{aligned}
\tag{6.1}
$$

The collection of latent variables in this model are therefore $\mathbf{U}_{n\times K}$, $\mathbf{V}_{p\times K}$ and the set of all Z_{ijk} gathered in the object $\mathbf{Z} = [Z_{ijk}] \in \mathbb{R}^{n\times p\times K}$.

The global framework for variational inference is recalled in Figure 6.1. As introduced in Chapter 5, from the model, we aim at approximating the intractable posterior. To do so, we define the variational distribution q. The inference of q is based on the optimization of the Evidence Lower Bound (ELBO) $J(q)$. This optimization requires to derive the complete conditional distributions of the latent variables.

Complete conditional and variational distributions

We consider the model with the latent variables Z_{ijk}, the principle[1] of variational inference is to find a distribution $q(\mathbf{U},\mathbf{V},\mathbf{Z})$ that approximates the posterior $p(\mathbf{Z},\mathbf{U},\mathbf{V}\,|\,\mathbf{X})$. Combined with the mean-field assumption, the variational distribution q is factorisable such as:

$$\begin{aligned} q(\mathbf{U},\mathbf{V},\mathbf{Z}) = \quad & \prod_{i=1}^{n}\prod_{k=1}^{K} q(u_{ik}\,|\,\mathbf{a}_{ik}) \times \prod_{j=1}^{p}\prod_{k=1}^{K} q(v_{jk}\,|\,\mathbf{b}_{jk}) \\ & \times \prod_{i=1}^{n}\prod_{j=1}^{p} q\big((z_{ijk})_k\,|\,(r_{ijk})_k\big)\,, \end{aligned} \tag{6.2}$$

where the variational parameters are

$$\begin{aligned} &\mathbf{a}_{ik} = (a_{ik,1}, a_{ik,2}) \in (\mathbb{R}^{+})^2\,, \\ &\mathbf{b}_{jk} = (b_{jk,1}, b_{jk,2}) \in (\mathbb{R}^{+})^2\,, \\ &(r_{ijk})_k \in [0,1]^K \quad \text{with } \textstyle\sum_k r_{ijk} = 1\,. \end{aligned}$$

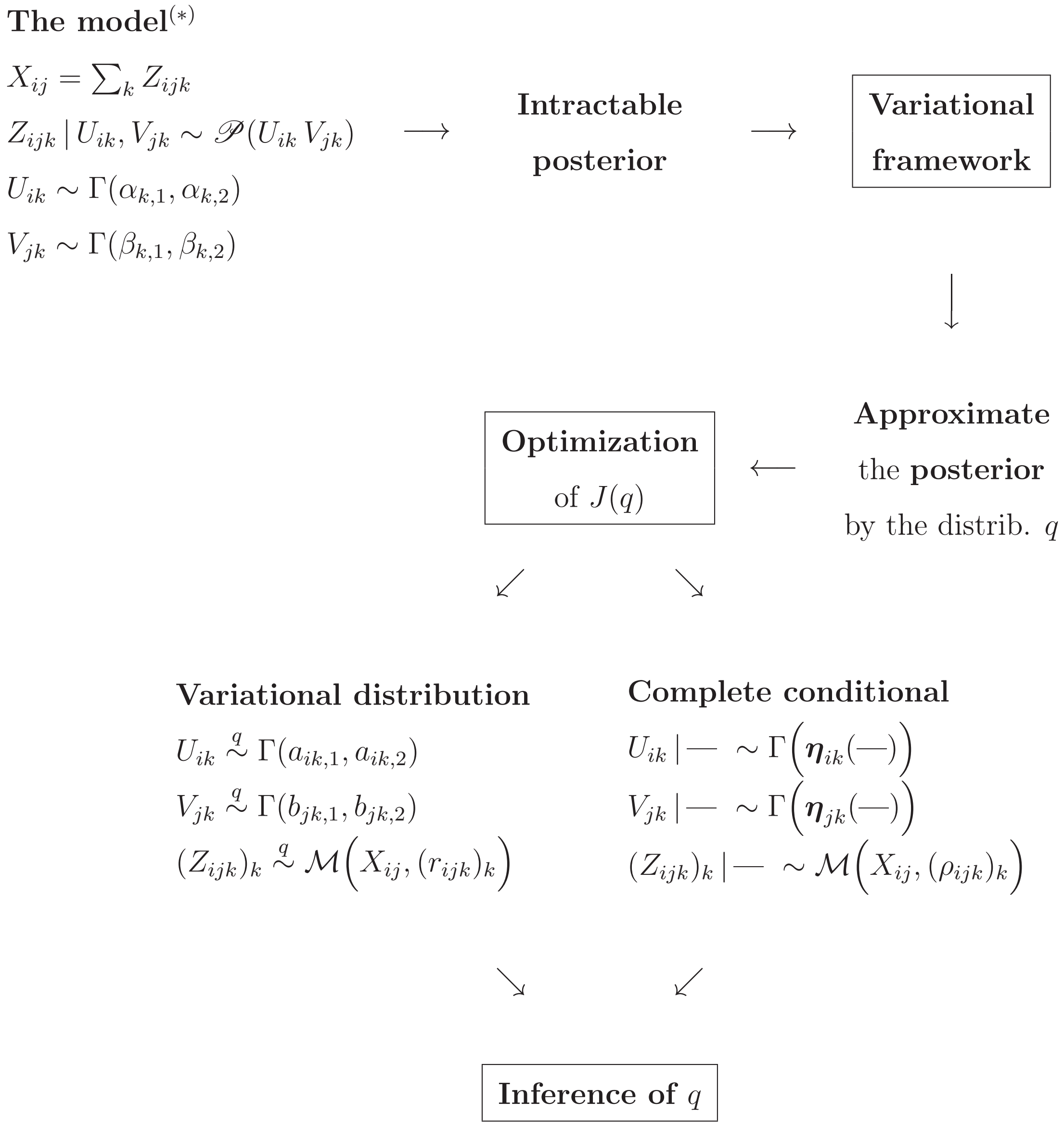
Moreover, as previously mentioned, each term in Equation (6.2) lies in the same exponential family as the complete conditional of the corresponding latent variables, i.e. each $p(u_{ik}\,|\,—)$ determines the type of each distribution $q(u_{ik}\,|\,\mathbf{a}_{ik})$, and similarly with $p(v_{jk}\,|\,—)$ for $q(v_{jk}\,|\,\mathbf{b}_{jk})$, and with $p\big((z_{ijk})_k\,|\,—\big)$ for $q\big((z_{ijk})_k\,|\,(r_{ijk})_k\big)$.

It has to be noted that we consider the joint distribution $q\big((z_{ijk})_k\,|\,(r_{ijk})_k\big)$ over the vector $(Z_{ijk})_{k=1:K}$ parametrized by $(r_{ijk})_k \in \mathbb{R}^K$, because it is only possible to derive the complete conditional $p\big((z_{ijk})_k\,|\,—\big)$, as we will see below.

The objective function that we aim at optimizing is the ELBO $J(q)$. It is defined as:

$$J(q) = \mathbb{E}_q[\log p(\mathbf{X},\mathbf{U},\mathbf{V},\mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{U},\mathbf{V},\mathbf{Z})]\,.$$

[1] When introducing the variational inference in the previous chapter, we did not consider the latent Poisson variables $\mathbf{Z}$ in order to avoid heavy notations. The principle remains unchanged.


The model(*)
$X_{ij} = \sum_k Z_{ijk}$
$Z_{ijk} \,|\, U_{ik}, V_{jk} \sim \mathscr{P}(U_{ik} V_{jk})$
$U_{ik} \sim \Gamma(\alpha_{k,1}, \alpha_{k,2})$
$V_{jk} \sim \Gamma(\beta_{k,1}, \beta_{k,2})$
Intractable posterior
Variational framework
Approximate the posterior by the distrib. q
Optimization of $J(q)$
Variational distribution
$U_{ik} \overset{q}{\sim} \Gamma(a_{ik,1}, a_{ik,2})$
$V_{jk} \overset{q}{\sim} \Gamma(b_{jk,1}, b_{jk,2})$
$(Z_{ijk})_k \overset{q}{\sim} \mathcal{M}\Big(X_{ij}, (r_{ijk})_k\Big)$
Complete conditional
$U_{ik} \,|\text{—} \sim \Gamma\Big(\boldsymbol{\eta}_{ik}(\text{—})\Big)$
$V_{jk} \,|\text{—} \sim \Gamma\Big(\boldsymbol{\eta}_{jk}(\text{—})\Big)$
$(Z_{ijk})_k \,|\text{—} \sim \mathcal{M}\Big(X_{ij}, (\rho_{ijk})_k\Big)$
Inference of q

(*) with conditional independence between the Z_{ijk}'s and independence between the U_{ik}'s and V_{jk}'s

Figure 6.1 – Variational inference to approximate the posterior of the model, based on the optimization of the ELBO that required to derive the complete conditional. The notation $\overset{q}{\sim}$ refers to the variational distribution.

In order to find a stationary point of the ELBO, $J(q)$ is differentiated regarding each variational parameter separately[2], here $\mathbf{a}_{ik}$, $\mathbf{b}_{jk}$ and $(r_{ijk})_k$. The formulation of the ELBO regarding each parameter separately is based on the corresponding complete conditional, i.e. $p(u_{ik}\,|\!-\!\!-)$, $p(v_{jk}\,|\!-\!\!-)$ and $p\big((z_{ijk})_k\,|\!-\!\!-\big)$ respectively. Therefore, the ELBO is explicit regarding each variational parameter and the gradient can be derived in order to find the coordinate of the stationary point, that corresponds to a local optimum.

As explained in Appendix Section D.2.1, the complete conditionals of U_{ik} and V_{jk} are Gamma distributions. The proof is based on the Bayes rule and the distribution of the latent variables $\mathbf{Z}$, that are actually necessary to derive $p(u_{ik}\,|\!-\!\!-)$ and $p(v_{jk}\,|\!-\!\!-)$. The complete conditional of the vector $(Z_{ijk})_{k=1:K}$ is also explicit, being especially a Multinomial distribution[3], i.e. $(Z_{ijk})_k\,|\!-\!\!- \;\sim \mathcal{M}\big(X_{ij},(\rho_{ijk})_k\big)$, with $\sum_k \rho_{ijk} = 1$. The Multinomial probabilities depend on $(U_{ik}, V_{jk})_k$ and are defined below. This point justifies why the variational distribution is based on the vector $(Z_{ijk})_{k=1:K}$ instead of taking each Z_{ijk} separately.

We summarize the complete conditionals in the GaP factor model, that are respectively defined as:

$$\begin{aligned}
U_{ik}\,|\!-\!\!- \;&\sim \Gamma(\alpha_{k,1} + \textstyle\sum_j z_{ijk},\ \alpha_{k,2} + \sum_j v_{jk})\,,\\
V_{jk}\,|\!-\!\!- \;&\sim \Gamma(\beta_{k,1} + \textstyle\sum_i z_{ijk},\ \beta_{k,2} + \sum_i u_{ik})\,,\\
(Z_{ijk})_k\,|\!-\!\!- \;&\sim \mathcal{M}\Big(X_{ij},(\rho_{ijk})_k\Big)\,,
\end{aligned} \tag{6.3}$$

where the Multinomial probabilities $(\rho_{ijk})_k$ quantify the contribution of the factor k to the observations X_{ij}, i.e. $\rho_{ijk} = \frac{u_{ik}v_{jk}}{\sum_\ell u_{i\ell}v_{j\ell}}$. As previously mentioned, the variational distribution is assumed to lie in the same exponential family as the corresponding complete conditional, thus:

$$\begin{aligned}
U_{ik} &\overset{q}{\sim} \Gamma(a_{ik,1}, a_{ik,2})\,,\\
V_{jk} &\overset{q}{\sim} \Gamma(b_{jk,1}, b_{jk,2})\,,\\
(Z_{ijk})_k &\overset{q}{\sim} \mathcal{M}\Big(X_{ij},(r_{ijk})_k\Big)\,.
\end{aligned}$$

We recall that $\overset{q}{\sim}$ denotes the variational distribution.

[2] c.f. Appendix Section D.2.2

[3] This results is explicitly proven in Zhou & Carin (2012)

An explicit coordinate descent algorithm

Thanks to the formulation in the exponential family, the stationary point of the ELBO is explicit regarding each variational parameter[4]. It depends on the parametrization of the complete conditional and the expectation with respect to the distribution q. For the factors U_{ik} and V_{jk}, it is respectively:

$$\begin{aligned} \mathbf{a}_{ik} &= \mathbb{E}_q\big[(\alpha_{k,1} + \textstyle\sum_j Z_{ijk},\ \alpha_{k,2} + \sum_j V_{jk})^T\big]\,, \\ \mathbf{b}_{jk} &= \mathbb{E}_q\big[(\beta_{k,1} + \textstyle\sum_i Z_{ijk},\ \beta_{k,2} + \sum_i U_{ik})^T\big]\,, \end{aligned} \tag{6.4}$$

i.e.

$$\begin{aligned} \mathbf{a}_{ik} &= \big(\alpha_{k,1} + \textstyle\sum_j \mathbb{E}_q[Z_{ijk}],\ \alpha_{k,2} + \sum_j \mathbb{E}_q[V_{jk}]\big)\,, \\ \mathbf{b}_{jk} &= \big(\beta_{k,1} + \textstyle\sum_i \mathbb{E}_q[Z_{ijk}],\ \beta_{k,2} + \sum_i \mathbb{E}_q[U_{ik}]\big)\,. \end{aligned}$$

Concerning Z_{ijk}, based on the derivation of the stationary point of the ELBO, the optimum values for the probability r_{ijk} verifies[5]:

$$\log(r_{ijk}) = \mathbb{E}_q[\log(\rho_{ijk})]\,,$$

which corresponds to the following formulation[6]:

$$r_{ijk} = \frac{\exp\Big(\mathbb{E}_q[\log(U_{ik})] + \mathbb{E}_q[\log(V_{jk})]\Big)}{\sum_\ell \exp\Big(\mathbb{E}_q[\log(U_{i\ell})] + \mathbb{E}_q[\log(V_{j\ell})]\Big)} \tag{6.5}$$

Since the coordinates of the point that set the gradient to zero are known, it is possible to optimize the ELBO through a coordinate descent (or coordinate ascent) algorithm[7] which computes alternate updates of the parameters following the relations in Equation (6.4) and Equation (6.5), c.f. Algorithm 6.1.

[4] c.f. Appendix Section D.2.2

[5] The log corresponds to the natural parametrization of the Multinomial distribution in the exponential family

[6] c.f. Appendix Section D.2.2

[7] See Wright (2015) for a review of this optimization process.

Moments of the variational distribution

The moments of the different latent variables Z_{ijk}, U_{ik} and V_{jk} regarding the variational distribution q are required in the algorithm. However, they are known since the variational distribution is explicit. Recalling the moments of the Multinomial distribution[8], the moments and log-moments of the Gamma distribution[9], we have:

$$\mathbb{E}_q[Z_{ijk}] = X_{ij}\, r_{ijk}$$

$$\begin{aligned} \mathbb{E}_q[U_{ik}] &= \frac{a_{ik,1}}{a_{ik,2}}\,, & \mathbb{E}_q[V_{jk}] &= \frac{b_{jk,1}}{b_{jk,2}}\,, \\ \mathbb{E}_q[\log(U_{ik})] &= \psi(a_{ik,1}) - \log(a_{ik,2})\,, & \mathbb{E}_q[\log(V_{jk})] &= \psi(b_{jk,1}) - \log(b_{jk,2})\,, \end{aligned}$$

where $\psi(\cdot)$ is the digamma function, i.e. $\psi(x) = \frac{\partial}{\partial x} \log\Gamma(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ for any $x > 0$.

As previously mentioned, at the end of the optimization process, the factors U_{ik} and V_{jk} are not estimated by the mode of q, which would correspond to an approximation of the Maximum a Posteriori (MAP). Instead, the factors U_{ik} and V_{jk} are respectively estimated by $\widehat{U}_{ik} = \mathbb{E}_q[U_{ik}]$ and $\widehat{V}_{jk} = \mathbb{E}_q[V_{jk}]$, which approximate the expectation of the posterior, i.e. $\mathbb{E}[U_{ik} \,|\, X_{ij}]$ and $\mathbb{E}[V_{jk} \,|\, X_{ij}]$ respectively.

```
Input: Hyper-parameter values $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$
Output: Estimates of $\mathbf{a}$, $\mathbf{b}$, $\mathbb{E}_q[\mathbf{U}]$, $\mathbb{E}_q[\mathbf{V}]$
Initialize $\mathbf{a} = [(a_{ik,1}, a_{ik,2})] \in \mathbb{R}^{n\times K\times 2}$, $\mathbf{b} = [(b_{jk,1}, b_{jk,2})] \in \mathbb{R}^{p\times K\times 2}$
repeat
    Multinomial parameters
        $r_{ijk} \leftarrow \frac{\exp\left(\mathbb{E}_q[\log(U_{ik})]+\mathbb{E}_q[\log(V_{jk})]\right)}{\sum_\ell \exp\left(\mathbb{E}_q[\log(u_{i\ell}]+\mathbb{E}_q[\log(v_{j\ell})]\right)}$
    Gamma parameters
        $\mathbf{a}_{ik} \leftarrow \big(\alpha_{k,1} + \sum_j \mathbb{E}_q[z_{ijk}],\ \alpha_{k,2} + \sum_j \mathbb{E}_q[V_{jk}]\big)$
        $\mathbf{b}_{jk} \leftarrow \big(\beta_{k,1} + \sum_i \mathbb{E}_q[z_{ijk}],\ \beta_{k,2} + \sum_i \mathbb{E}_q[U_{ik}]\big)$
until Convergence
return a, b
```

Algorithm 6.1: Variational algorithm to infer the GaP factor model

[8] If $(Z_k)_k \sim \mathcal{M}\big(X, (r_k)\big)$, then $\mathbb{E}[Z_k] = X\, r_k$

[9] If $U \sim \Gamma(\alpha_1, \alpha_2)$, then $\mathbb{E}[U] = \alpha_1/\alpha_2$ and $\mathbb{E}[\log(U)] = \psi(\alpha_1) - \log(\alpha_2)$.

Convergence

The convergence is assessed by controlling the normalized gap between two iterates. If we denote by $\mathbf{O}^{(t)}$ the vectorized set of the values of all variational parameters in the model at the iteration t, i.e. $\mathbf{O}^{(t)} = (\mathbf{a}, \mathbf{b})$, the normalized gap between two iterates is defined as:

$$\gamma^{(t)} = \frac{\|\mathbf{O}^{(t)} - \mathbf{O}^{(t-1)}\|_2}{\|\mathbf{O}^{(t-1)}\|_2} \,.$$

Checking the evolution of the objective function is not sufficient when assessing the convergence because if the optima lie in a part of the space where the objective is near flat, the variation of the coordinates through the argmax may be huge compared to the variation of the objective (as for instance high-order monomial). On the contrary, checking the variations of the objective may lead to wrong estimates of the optimum point.

6.1.2 Formulation of the variational EM algorithm

E-step

In the Expectation-Maximization (EM) algorithm, the E-step consists in computing the expectation of the joint likelihood $\mathbb{E}_{\mathbf{Z},\mathbf{U},\mathbf{V}\,|\,\mathbf{X}}[\log p(\mathbf{X},\mathbf{Z},\mathbf{U},\mathbf{V}\,;\,\boldsymbol{\alpha},\boldsymbol{\beta})]$. As we saw, this integral is intractable. However, the posterior is approximated by the variational distribution q, hence it is possible to approximate the objective function of the EM algorithm by the expectation of the joint likelihood regarding the variational distribution:

$$\begin{aligned}\mathbb{E}_{\mathbf{Z},\mathbf{U},\mathbf{V}\,|\,\mathbf{X}}[\log p(\mathbf{X},\mathbf{Z},\mathbf{U},\mathbf{V}\,;\,\boldsymbol{\alpha},\boldsymbol{\beta})] \approx\; & \mathbb{E}_q[\log p(\mathbf{X}\,|\,\mathbf{Z})] \\ & + \mathbb{E}_q[\log p(\mathbf{Z}\,|\,\mathbf{U},\mathbf{V})] \\ & + \mathbb{E}_q[\log p(\mathbf{U},\mathbf{V}\,;\,\boldsymbol{\alpha},\boldsymbol{\beta})]\,.\end{aligned} \tag{6.6}$$

The conditional distribution of $\mathbf{X}$ knowing $\mathbf{Z}$ is deterministic as $X_{ij} = \sum_k Z_{ijk}$. The term $\mathbb{E}_q[\log p(\mathbf{Z}\,|\,\mathbf{U},\mathbf{V})]$ is developed as:

$$\begin{aligned}\mathbb{E}_q[\log p(\mathbf{Z}\,|\,\mathbf{U},\mathbf{V})] = \sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{k=1}^{K} \Big\{ & - \mathbb{E}_q[U_{ik}]\,\mathbb{E}_q[V_{jk}] + \mathbb{E}_q[z_{ijk}]\,\mathbb{E}_q[\log(U_{ik})] \\ & + \mathbb{E}_q[z_{ijk}]\,\mathbb{E}_q[\log(V_{jk})] - \mathbb{E}_q[\log\Gamma(z_{ijk}+1)]\Big\}\end{aligned}$$

which is constant regarding the hyper-parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and will disappear when differentiating. The last term in Equation (6.6) is the only one depending on the hyper-parameters. The expectation regarding q of the Gamma log-priors are especially:

$$\begin{aligned}
\mathbb{E}_q[\log p(\mathbf{U},\mathbf{V}\,;\,\boldsymbol{\alpha},\boldsymbol{\beta})] = \quad & \sum_{i=1}^{n}\sum_{k=1}^{K}\Big\{(\alpha_{k,1}-1)\,\mathbb{E}_q[\log(U_{ik})] + \alpha_{k,1}\,\log\alpha_{k,2} \\
& \qquad - \alpha_{k,2}\,\mathbb{E}_q[U_{ik}] - \log\Gamma(\alpha_{k,1})\Big\} \\
& + \sum_{j=1}^{p}\sum_{k=1}^{K}\Big\{(\beta_{k,1}-1)\,\mathbb{E}_q[\log(V_{jk})] + \beta_{k,1}\,\log\beta_{k,2} \\
& \qquad - \beta_{k,2}\,\mathbb{E}_q[V_{jk}] - \log\Gamma(\beta_{k,1})\Big\}\,.
\end{aligned}$$

M-step

The objective function in the M-step of the variational-EM algorithm is formulated as:

$$\widetilde{Q}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \mathbb{E}_q[\log p(\mathbf{U},\mathbf{V}\,;\,\boldsymbol{\alpha},\boldsymbol{\beta})] + \text{const} \tag{6.7}$$

As detailled in Appendix Section D.2.3, the stationary point that sets the gradient of $\widetilde{Q}(\boldsymbol{\alpha},\boldsymbol{\beta})$ to zero verifies:

$$\begin{cases}
\psi(\alpha_{k,1}) = \log(\alpha_{k,2}) + \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\mathbb{E}_q[\log(U_{ik})]\,, \\
\alpha_{k,2} = n\,\dfrac{\alpha_{k,1}}{\sum_{i=1}^{n}\mathbb{E}_q[U_{ik}]}\,, \\
\psi(\beta_{k,1}) = \log(\beta_{k,2}) + \dfrac{1}{p}\displaystyle\sum_{j=1}^{p}\mathbb{E}_q[\log(V_{jk})]\,, \\
\beta_{k,2} = p\,\dfrac{\beta_{k,1}}{\sum_{j=1}^{p}\mathbb{E}_q[V_{jk}]}\,,
\end{cases} \tag{6.8}$$

where the digamma function ψ is defined as $\psi(x) = \frac{\partial}{\partial x}\log\Gamma(x)$ for any $x > 0$.

It can be noted that the prior hyper-parameters are computed so that the prior moment and log-moment correspond to their respective empirical counterparts regarding q. For instance, since $U_{ik} \sim \Gamma(\alpha_{k,1},\alpha_{k,2})$ in the model, we have $\mathbb{E}[U_{ik}] = \frac{\alpha_{k,1}}{\alpha_{k,2}}$ and $\mathbb{E}[\log(U_{ik})] = \psi(\alpha_{k,1}) - \log(\alpha_{k,2})$. Following Equation (6.8), the updates of $(\alpha_{k,1},\alpha_{k,2})$ exactly verify:

$$\begin{aligned}
&\psi(\alpha_{k,1}) - \log(\alpha_{k,2}) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_q[\log(U_{ik})]\,, \\
&\frac{\alpha_{k,1}}{\alpha_{k,2}} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_q[U_{ik}]\,,
\end{aligned}$$

and similarly for V_{jk}.

Algorithm

The variational EM algorithm is defined in Algorithm 6.2. To update $\alpha_{k,1}$ and $\beta_{k,1}$, we need to invert the digamma function ψ, or at least to resolve the problem $y = \psi(x)$ regarding x when y is known, fast with high accuracy. We refer to the method by Minka (2000). It is based on a Newton-Raphson algorithm that finds the root of the equation $y - \psi(x) = 0$, which is unique as the digamma function is strictly increasing on $\mathbb{R}^+$. The convergence is assessed similarly as in Algorithm 6.1 by controlling the normalized gap between two iterates $\widetilde{\mathbf{O}}^{(t)}$ where $\widetilde{\mathbf{O}}^{(t)} = (\mathbf{a}, \mathbf{b}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ stores the vectorized values of all variational parameters and hyper-parameters in the model at the iteration t.

```
Input: /
Output: Estimates of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathbf{a}$, $\mathbf{b}$, $\mathbb{E}_q[\mathbf{U}]$, $\mathbb{E}_q[\mathbf{V}]$
Initialize $\boldsymbol{\alpha} = [(\alpha_{k,1}, \alpha_{k,2})] \in \mathbb{R}^{K \times 2}$, $\boldsymbol{\beta} = [(\beta_{k,1}, \beta_{k,2})] \in \mathbb{R}^{K \times 2}$
Initialize $\mathbf{a} = [(a_{ik,1}, a_{ik,2})] \in \mathbb{R}^{n \times K \times 2}$, $\mathbf{b} = [(b_{jk,1}, b_{jk,2})] \in \mathbb{R}^{p \times K \times 2}$
repeat
    E-STEP
        Multinomial variational parameters
            $r_{ijk} \leftarrow \frac{\exp\left(\mathbb{E}_q[\log(U_{ik})]+\mathbb{E}_q[\log(V_{jk})]\right)}{\sum_\ell \exp\left(\mathbb{E}_q[\log(u_{i\ell}]+\mathbb{E}_q[\log(v_{j\ell})]\right)}$
        Gamma variational parameters
            $\mathbf{a}_{ik} \leftarrow \big(\alpha_{k,1} + \sum_j \mathbb{E}_q[z_{ijk}],\ \alpha_{k,2} + \sum_j \mathbb{E}_q[V_{jk}]\big)$
            $\mathbf{b}_{jk} \leftarrow \big(\beta_{k,1} + \sum_i \mathbb{E}_q[z_{ijk}],\ \beta_{k,2} + \sum_i \mathbb{E}_q[U_{ik}]\big)$
    M-STEP
        Gamma hyper-parameters
            $\alpha_{k,1} \leftarrow \psi^{-1}\big(\log(\alpha_{k,2}) + \frac{1}{n}\sum_{i=1}^n \mathbb{E}_q[\log(U_{ik})]\big)$
            $\alpha_{k,2} \leftarrow n\,\frac{\alpha_{k,1}}{\sum_{i=1}^n \mathbb{E}_q[U_{ik}]}$
            $\beta_{k,1} \leftarrow \psi^{-1}\big(\log(\beta_{k,2}) + \frac{1}{p}\sum_{j=1}^p \mathbb{E}_q[\log(V_{jk})]\big)$
            $\beta_{k,2} \leftarrow p\,\frac{\beta_{k,1}}{\sum_{j=1}^p \mathbb{E}_q[V_{jk}]}$
until Convergence
return $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathbf{a}$, $\mathbf{b}$
```

Algorithm 6.2: Variational-EM algorithm to infer the GaP factor model

Standard variational algorithm versus variational EM algorithm

In order to assess the interest of the variational-EM algorithm compared to the standard variational algorithm (i.e. Algorithm 6.2 versus Algorithm 6.1 respectively), we generate synthetic data according to the GaP factor model. We would expect the variational-EM (var-EM) algorithm to converge faster since it narrows the exploration of the variational parameter space by updating the hyper-parameters of the model. We consider different configurations regarding the values of n, p and K^*. The number K^* will refer to the true number of factors in the data, whereas K will correspond to the number of factors in the considered model. For each configuration, each algorithm is run 50 times. Figure 6.2 shows the evolution of the ELBO through iterations for different runs[10] of each algorithm on the same data set for different configurations (n, p, K^*). In all configurations, the variational-EM algorithm reaches a better local optimum, characterized by higher values of the ELBO. The interesting point is that the improvements of the optimization does not seem to have a cost regarding the speed of convergence.

Indeed, Figure 6.2 also shows the evolution of the convergence criterion through iterations for the same runs. As expected, the empirical convergence speed appears to be a bit better in the case of the variational-EM algorithm although it updates more parameters than the standard variational algorithm. It can be noted that even in high dimension $(p > n)$, the number of iterations to reach the convergence state does not increase.

6.1.3 Initialization of the parameters

The optimization procedures in Algorithm 6.1 and Algorithm 6.2 are based on a coordinate ascent algorithm. Therefore, the optimal values that are returned correspond to a local optimum. Such procedures are very sensitive to the initialization of the parameters. For instance, Figure 6.2 shows the different trajectories of the ELBO for 10 different runs of our two algorithms (variational and variational-EM) with different random initialization on the same simulated data set. To avoid the potential issues related to the dependence of the solution on the initial values, which may lead to a "bad" local optimum, we decided to run the algorithm with multiple initializations and choose the solution as the one associated with the highest value of the objective function. To reduce the computational cost of multiple initializations, our algorithm iterates each run a hundred times, and then keeps the best seed regarding the value of the ELBO to be iterated until convergence. The question of initialization will be discussed in Chapter 7 regarding possible improvements of the algorithm in the future.

[10] For each run, the parameters are randomly initialized, c.f. next section.

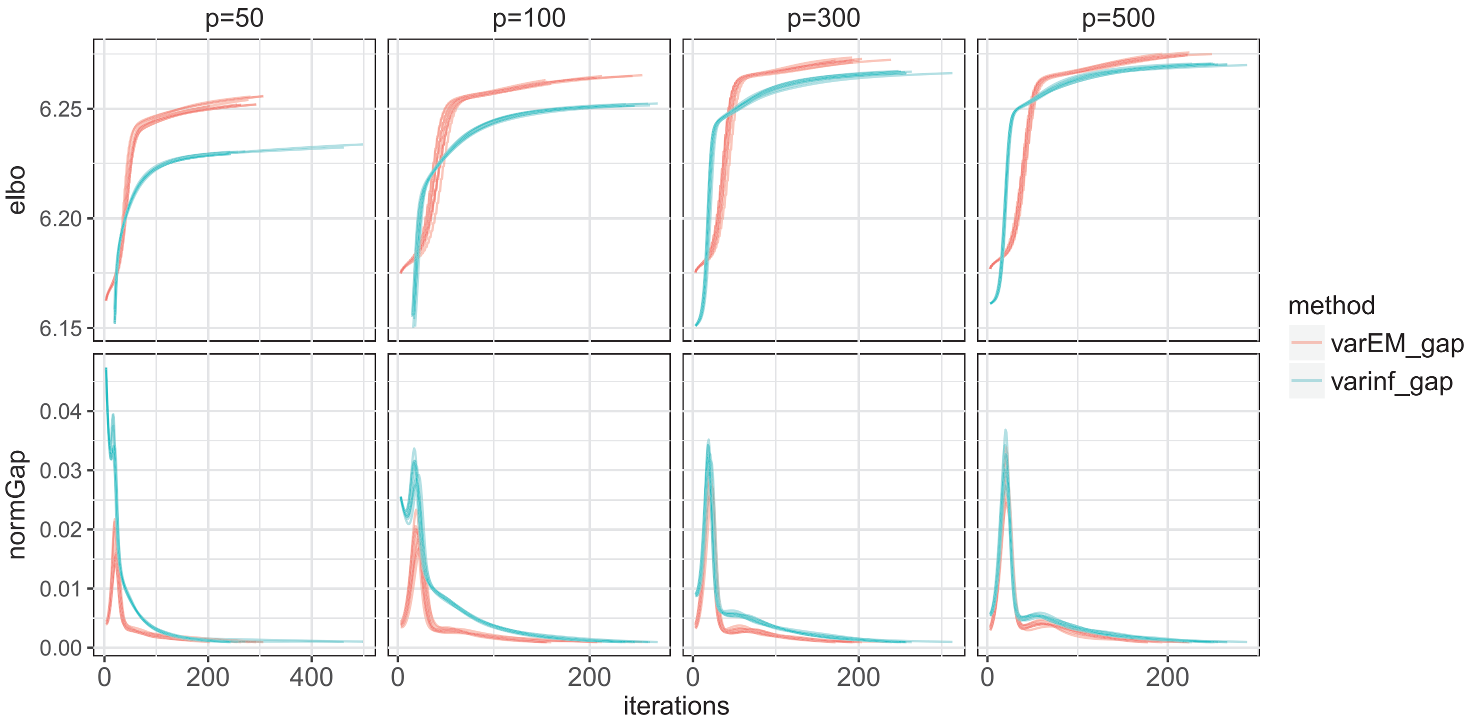


Figure 6.2 – (Top) Evolution of the ELBO through iterations for 10 different trajectories (i.e. different initializations), comparison of Algorithm 6.2 (`varEM_gap`) in red and Algorithm 6.1 (`varinf_gap`) in blue. Data are generated with $n = 100$, $K^* = 10$ and $p = 50, 100, 300, 500$. Model fitted with $K = 10$ factors. The ELBO values are normalized depending on the value of p to fit on the same scale. The interest here is the tendency. The values between different p are not comparable on this graph. (Bottom) Evolution of the convergence criterion through iterations for the same runs (the values here are not scaled).

The variational algorithm requires to supply fixed values for the hyper-parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The initial values of the variational parameter $\mathbf{a}$ and $\mathbf{b}$ are set randomly. In the variational-EM algorithm, all parameters, i.e. $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathbf{a}$ and $\mathbf{b}$, are randomly initialized.

6.1.4 Improvements of the convergence speed

The efficiency of the iterative procedure in variational inference may be improved regarding the number of iterations to reach the convergence. Hoffman et al. (2013) proposed to use stochastic gradient methods based on non-Euclidean geometry. Their purpose was to process a gradient descent by estimating the gradient at each iteration on a subsample of the observations (reducing the cost of the gradient computation) and explore more efficiently the parameter space to find the optimum (using a metric suitable for probability distributions). We tried to use stochastic variational inference in our model. However, we did not observe any clear improvement regarding speed convergence. It may be explained by the fact that the sample size that we consider ($n = O(10^2)$ or $n \sim 10^3$ at most) does not require to use convergence improvement methods, since the standard variational inference is relatively efficient regarding convergence. For instance, Hoffman et al. (2013) consider data sets where the number n of individuals reaches orders of 10^5 or 10^6.

We also tried to adapt the "epsilon" algorithm to the variational framework. Such method was developed to accelerate the convergence of slowly converging sequence (Graves-Morris et al., 2000), especially in the case of the EM algorithm (Wang et al., 2008). However, the results were not as good as expected, the speed of convergence was not increased. Sometimes, this procedure even disturbed and delayed the convergence of the variational algorithm when considering inference in the GaP factor model. Following these tests, we did choose to use the standard variational-EM algorithm.

6.2 Sparse and zero-inflated matrix factorization

We now define the zero-inflated Gamma-Poisson (ZI-GaP) factor model, based on zero-inflated Poisson conditional distributions that account for potential drop-out events[11]. We derive the associated variational-EM algorithm. Then, we define a sparse Gamma-Poisson factor model that induces sparsity among the columns of the factor $\mathbf{V}$ in order to select variables. This approach will be based on a spike-and-slab formulation and the model will also be inferred by a variational-EM algorithm.

[11] as introduced in Chapter 4

6.2.1 Zero-inflated model

As previously stated, in zero-inflated data, an unknown proportion of zeros correspond to drop-out events, i.e. unobserved values. Therefore, the null values in $\mathbf{X}$ originate from the Poisson distribution or from a loss of the signal. To model such patterns, we consider a Poisson-Dirac mixture Lambert (1992). The conditional distribution of the X_{ij}'s is the following:

$$X_{ij} \,|\, (U_{ik}, V_{jk})_{k=1,\dots,K} \sim (1 - \pi_j^{\text{D}}) \times \delta_0 + \pi_j^{\text{D}} \times \mathscr{P}(\textstyle\sum_k U_{ik}\, V_{jk})\,, \tag{6.9}$$

where δ_0 is the Dirac mass function on $\{0\}$. The probability π_j^{D} regulates the balance between drop-out events and the true signal. Thus, $1 - \pi_j^{\text{D}}$ is the probability that X_{ij} is a drop-out event (the D stands for drop-out). For instance, the conditional probability that X_{ij} is null becomes:

$$\mathbb{P}\big(X_{ij} = 0 \,|\, (U_{ik}, V_{jk})_k\big) = (1 - \pi_j^{\text{D}}) + \pi_j^{\text{D}} \,\exp\big(\textstyle\sum_k u_{ik}\, v_{jk}\big)\,,$$

accounting for the two sources of zeros.

The parameters $(\pi_j^{\text{D}})_{j=1,\dots,p}$ are considered variable-specific because in our application regarding gene expression profiles the drop-out rate tends to depend on the genes (Pierson & Yau, 2015). In order to integrate the zero-inflation in the variational framework, we consider hidden variables $D_{ij} \in \{0, 1\}$ for $i = 1, \dots, n$ and $j = 1, \dots, p$. Each D_{ij} indicates the status of the observation X_{ij} for the individual i and gene j. When, D_{ij} is null, the observation X_{ij} is a drop-out events and then null. On the contrary, when D_{ij} is equal to 1, the observation X_{ij} (null or not) is the true one (drawn from the Poisson distribution). The drop-out indicators are binary latent variables following a Bernoulli distribution, i.e. $D_{ij} \sim \mathcal{B}(\pi_j^{\text{D}})$.

To define the model, the drop-out indicators are incorporated in the conditional Poisson distribution as:

$$Z_{ijk} \,|\, U_{ik}, V_{jk}, D_{ij} \sim \mathscr{P}(D_{ij}\, U_{ik}\, V_{jk})\,.$$

Indeed, the Poisson distribution degenerates in the Dirac mass δ_0 when the rate is null, i.e. when $D_{ij} = 0$. Thus, this formulation is a rewriting of the following conditional distribution:

$$Z_{ijk} \,|\, U_{ik}, V_{jk}, D_{ij} \sim (1 - D_{ij}) \times \delta_0 + D_{ij} \times \mathscr{P}(U_{ik}\, V_{jk})\,.$$

Finally, the ZI-GaP model is summarized as follows:

$$\begin{aligned}
&X_{ij} = \sum_k Z_{ijk}\,,\\
&Z_{ijk}\,|\,U_{ik}, V_{jk}, D_{ij} \sim \mathscr{P}(D_{ij}\,U_{ik}\,V_{jk})\,,\\
&U_{ik} \sim \Gamma(\alpha_{k,1}, \alpha_{k,2})\,,\\
&V_{jk} \sim \Gamma(\beta_{k,1}, \beta_{k,2})\,,\\
&D_{ij} \sim \mathcal{B}(\pi_j^{\mathrm{D}})\,,
\end{aligned} \tag{6.10}$$

with conditional independence between the variables Z_{ijk} and independence between the factors U_{ik} and V_{jk}. The drop-out indicators D_{ij} are assumed to be independent from the factors. Following this definition, when integrating D_{ij} out, the conditional distribution of X_{ij} knowing the latent factors indeed correspond to the Poisson-Dirac mixture (6.9).

The posterior is again approximated thanks to the variational framework. We recall that we follow the following steps. The variational distribution q is an approximation of the posterior. The optimization process requires to consider some constraints of independence on q and to assume the distribution of the different latent variables regarding q. We derive the complete conditional distributions and finally optimize the ELBO.

We assign a variational distribution $q(D_{ij}\,|\,p_{ij}^{\mathrm{D}})$ to each variable D_{ij}. These are incorporated into the variational framework and the variational distribution q becomes:

$$\begin{aligned}
q(\mathbf{U},\mathbf{V},\mathbf{Z},\mathbf{D}) = \;& \prod_{i=1}^{n}\prod_{k=1}^{K} q(u_{ik}\,|\,\mathbf{a}_{ik}) \times \prod_{j=1}^{p}\prod_{k=1}^{K} q(v_{jk}\,|\,\mathbf{b}_{jk})\\
&\times \prod_{i=1}^{n}\prod_{j=1}^{p} q\big((z_{ijk})_k\,|\,(r_{ijk})_k\big) \times \prod_{i=1}^{n}\prod_{j=1}^{p} q(D_{ij}\,|\,p_{ij}^{\mathrm{D}})\,.
\end{aligned} \tag{6.11}$$

The variational parameter p_{ij}^{D} for the variable D_{ij} depends on both i and j, on the contrary to the prior parameter π_j^{D}. Indeed, in the variational framework, the parameter p_{ij}^{D} of $q(D_{ij}\,;\,p_{ij}^{\mathrm{D}})$ depends on both $(U_{ik})_k$ and $(V_{jk})_k$ that are individual and gene specific.

Complete conditional distributions

In order to derive the variational algorithm, we need to compute the complete conditionals of the different latent variables. Despite the zero-inflated model, the different complete conditional distributions remain explicit. We can show that the complete conditional regarding $(Z_{ijk})_k$ remains a Multinomial distribution, when $D_{ij} = 1$. When $D_{ij} = 0$, it implies that $X_{ij} = 0$ so that Z_{ijk} are set to zero. In particular, the complete conditional of $(Z_{ijk})_k$ is defined as:

$$(Z_{ijk})_k\,|\,\text{—} \sim \mathcal{M}\Big(X_{ij}, (\rho_{ijk})_k\Big)\,, \tag{6.12}$$

where $\rho_{ijk} = \frac{u_{ik} v_{jk}}{\sum_\ell u_{i\ell} v_{j\ell}}$, for all pairs (i, j) such that $X_{ij} \neq 0$.

The complete conditional regarding the factors U_{ik} and V_{jk} are also explicit, especially the contribution of the individual i and gene j is balanced by the drop-out indicator D_{ij}. They are Gamma distribution, respectively parametrized as:

$$\begin{aligned} U_{ik} \,|\, — &\sim \Gamma(\alpha_{k,1} + \textstyle\sum_j D_{ij}\, z_{ijk},\ \alpha_{k,2} + \sum_j D_{ij}\, v_{jk})\,, \\ V_{jk} \,|\, — &\sim \Gamma(\beta_{k,1} + \textstyle\sum_i D_{ij}\, z_{ijk},\ \beta_{k,2} + \sum_i D_{ij}\, u_{ik})\,. \end{aligned} \tag{6.13}$$

We prove this point regarding the factor U_{ik} for example. Because of the independence between latent factors and thanks to the Bayes rule, the complete conditional of U_{ik} can be reduced to:

$$p\big(u_{ik} \,|\, (z_{ijk})_j, (v_{jk})_j\big) \propto p\big((z_{ijk})_j \,|\, u_{ik}, (v_{jk})_j\big)\ p(u_{ik})\,.$$

The $(Z_{ijk})_j$ are conditionally independent and each Z_{ijk} conditionally follows a Poisson distribution when $D_{ij} \neq 0$. Thus, the complete conditional of U_{ik} can be formulated as a product of Poisson and Gamma densities[12]:

$$\begin{aligned} p(u_{ik} \,|\, —) \propto \quad & \prod_{j, D_{ij} \neq 0} \Big\{ \exp\Big(- u_{ik} v_{jk} + \log(u_{ik} v_{jk}) z_{ijk} \Big)\ \frac{1}{z_{ijk}\,!} \Big\} \\ & \times \frac{(\alpha_{k,2})^{\alpha_{k,1}}}{\Gamma(\alpha_{k,1})}\ \exp\Big(- \alpha_{k,2}\, u_{ik} + (\alpha_{k,1} - 1)\, \log(u_{ik}) \Big)\,. \end{aligned}$$

When reordering all the terms, it becomes:

$$\begin{aligned} p(u_{ik} \,|\, —) \propto \quad & \exp\Big(\big(\alpha_{k,1} - 1 + \textstyle\sum_{j, D_{ij} \neq 0} z_{ijk}\big) \log(u_{ik}) \Big) \\ & \times \exp\Big(- u_{ik}\, \big(\alpha_{k,2} + \textstyle\sum_{j, D_{ij} \neq 0} v_{jk}\big) \Big)\,. \end{aligned}$$

This corresponds to the density of a Gamma distribution. Moreover, as D_{ij} takes its values in $\{0, 1\}$, the sums $\sum_{j, D_{ij} \neq 0} z_{ijk}$ and $\sum_{j, D_{ij} \neq 0} v_{jk}$ can be rewritten as $\sum_j D_{ij}\, z_{ijk}$ and $\sum_j D_{ij}\, v_{jk}$ respectively, so that $p(u_{ik} \,|\, —)$ is defined as in Equation (6.13). By the same reasoning, we can derive the complete conditional of V_{jk}.

Concerning the complete conditional of the drop-out indicator D_{ij}, the distribution of a binary variable is either deterministic or Bernoulli. Hence we have to compute the probability of D_{ij} being 0 or 1 knowing all the other variables in the model. A first remark is that, when we observe $X_{ij} \neq 0$, the posterior of D_{ij} is explicit since we know for sure that $D_{ij} = 1$. Therefore, in such case, the complete conditional of D_{ij} is

[12] In the case of discrete random variables as the Z_{ijk}, the term density refers to the probability mass function.

deterministic and the variational distribution corresponds exactly to the posterior with $p^{\text{D}}_{ij} = 1$.

When we observe $X_{ij} = 0$ in the data, the distribution of D_{ij} is not deterministic because the zero may originates from the Poisson draw or from a drop-out event. In this case, thanks to the Bayes rule, we show that:

$$p(D_{ij} = 1 \,|\, —) \propto p(D_{ij} = 1) \times \textstyle\prod_k p(Z_{ijk} = 0 \,|\, U_{ik}, V_{jk}, D_{ij} = 1)\,,$$

which corresponds to:

$$p(D_{ij} = 1 \,|\, —) \propto \pi^{\text{d}}_j \times \textstyle\prod_k \exp(-U_{ik}\, V_{jk})\,. \tag{6.14}$$

We can also show that:

$$p(D_{ij} = 0 \,|\, —) \propto (1 - \pi^{\text{d}}_j)\,. \tag{6.15}$$

Indeed, the conditional distribution of the variables $(Z_{ijk})_k$ knowing $D_{ij} = 0$ and $X_{ij} = 0$ is deterministic.

Derivation of the algorithm

The variational-EM algorithm for the ZI-GaP factor model is derived as follows (c.f. Algorithm 6.3). As previously, in the E-step, we compute the stationary point of the ELBO, i.e. the point that sets the gradient to zero regarding the parameters of the variational distribution $\mathbf{a}_{ik}$, $\mathbf{b}_{jk}$, $(r_{ijk})_k$ and p^{D}_{ij}. We recall that the stationary condition is that each variational parameter is equal to the expectation regarding q of the parameters of the corresponding complete conditional when considering the natural parametrization in the exponential family. In particular, based on Equations (6.12) and (6.13), for U_{ik}, V_{jk} and Z_{ijk}, we have respectively:

$$\begin{aligned}
\mathbf{a}_{ik} &= \mathbb{E}_q\big[(\alpha_{k,1} + \textstyle\sum_j D_{ij}\, Z_{ijk},\ \alpha_{k,2} + \textstyle\sum_j D_{ij}\, V_{jk})^T\big]\,,\\
\mathbf{b}_{jk} &= \mathbb{E}_q\big[(\beta_{k,1} + \textstyle\sum_i D_{ij}\, Z_{ijk},\ \beta_{k,2} + \textstyle\sum_i D_{ij}\, U_{ik})^T\big]\,,\\
\log(r_{ijk}) &= \mathbb{E}_q[\log(\rho_{ijk})]\,,
\end{aligned} \tag{6.16}$$

where $\mathbb{E}_q[D_{ij}] = p^{\text{D}}_{ij}$. The other expectations are derived as in the standard algorithm. Concerning D_{ij}, the natural parametrization of the Bernoulli distribution is based on the logit[13] of the probability of success. Therefore, depending on the values of X_{ij} and based on Equations (6.14) and (6.15), the stationary point is defined as:

$$\begin{cases}
p^{\text{D}}_{ij} = 1 & \text{if } X_{ij} \neq 0\,,\\
\text{logit}(p^{\text{D}}_{ij}) = \log\left(\dfrac{p(D_{ij} = 1 \,|\, —)}{p(D_{ij} = 0 \,|\, —)}\right) & \text{if } X_{ij} = 0\,.
\end{cases}$$

[13] The function logit is defined as $\text{logit}(\pi) = \frac{\log(\pi)}{\log(1-\pi)}$ for any $\pi \in (0, 1)$.

Finally, when $X_{ij} = 0$, the update formulation for p_{ij}^{D} verifies:

$$\log \frac{p_{ij}^{\text{D}}}{1 - p_{ij}^{\text{D}}} = \log \frac{\pi_j^{\text{D}}}{(1 - \pi_j^{\text{D}})} - \textstyle\sum_k \mathbb{E}_q[U_{ik}]\, \mathbb{E}_q[V_{jk}]\,. \tag{6.17}$$

This can be interpreted as correcting the prior probability of X_{ij} being a drop-out event by the estimated probability that X_{ij} is null if it is drawn from the Poisson distribution. At a first sight, we may think that the zero-inflated variational formulation is over-parametrized as we estimate $n \times p$ drop-out probabilities. However, such formulation of p_{ij}^{D} only depends on the expectation of U_{ik} and V_{jk} regarding q. We estimate $2 \times (n \times K + p \times K)$ parameters to infer the model, which is lower than the dimension of the data $(n \times p)$ when $K \ll \min(n, p)$, hence avoiding over-parametrization.

In the M-step, the update of the Gamma prior hyper-parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ remain unchanged. Whereas, the hyper-parameter π_j^{D} is updated as the mean of the corresponding variational parameters p_{ij}^{D}:

$$\pi_j^{\text{D}} = \frac{1}{n} \sum_{i=1}^{n} p_{ij}^{\text{D}}\,.$$

To prove this, we just have to write the objective function of the M-step and derive it regarding π_j^{D}.

6.2.2 Sparsity-inducing prior

We now consider that among all the recorded variables $j = 1, \ldots, p$, only a small proportion carries the signal whereas the others constitute some noise (hypothesis of parsimony). We modify the prior on the factor V_{jk} to consider a sparse model with a two-group sparsity-inducing prior:

$$V_{jk} \sim (1 - \pi_j^{\text{S}})\, \delta_0 + \pi_j^{\text{S}}\, \Gamma(\beta_{k,1}, \beta_{k,2})\,.$$

This spike-and-slab formulation ensures that V_{jk} is either null, i.e. the variable j does not contribute to the factor k, or drawn from the Gamma distribution, i.e. the contribution of the variable j to the component k is pertinent. The balance between discarding the variable and keeping it is regulated by the probability π_j^{S}. This spike-and-slab parameter (S stands for sparse) depends on j so that π_j^{S} can be seen as the probability that the variable j is in the model.

In order to properly define the model, we introduce a Bernoulli variable S_{jk} that indicates if V_{jk} is ruled by the Gamma or by the Dirac, i.e. $S_{jk} \sim \mathcal{B}(\pi_j^{\text{S}})$. To define the sparse GaP factor model, we use a latent factor $\widetilde{V}_{jk}$ that is independent from S_{jk} and

```
Input: /
Output: Estimates of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathbf{a}$, $\mathbf{b}$, $\mathbb{E}_q[\mathbf{U}]$, $\mathbb{E}_q[\mathbf{V}]$, $(\pi_j^{\text{D}})_j$ and $(p_{ij}^{\text{D}})_{ij}$
Initialize $\boldsymbol{\alpha} = [(\alpha_{k,1}, \alpha_{k,2})] \in \mathbb{R}^{K\times 2}$, $\boldsymbol{\beta} = [(\beta_{k,1}, \beta_{k,2})] \in \mathbb{R}^{K\times 2}$
Initialize $\mathbf{a} = [(a_{ik,1}, a_{ik,2})] \in \mathbb{R}^{n\times K\times 2}$, $\mathbf{b} = [(b_{jk,1}, b_{jk,2})] \in \mathbb{R}^{p\times K\times 2}$
repeat
    E-STEP
        Drop-out variational parameters
            $p_{ij}^{\text{D}} = \text{logit}^{-1}\big(\text{logit}(\pi_j^{\text{D}}) - \sum_k \mathbb{E}_q[U_{ik}]\,\mathbb{E}_q[V_{jk}]\big)$
        Multinomial variational parameters
            $r_{ijk} \leftarrow \frac{\exp\left(\mathbb{E}_q[\log(U_{ik})]+\mathbb{E}_q[\log(V_{jk})]\right)}{\sum_\ell \exp\left(\mathbb{E}_q[\log(u_{i\ell}]+\mathbb{E}_q[\log(v_{j\ell})]\right)}$
        Gamma variational parameters
            $\mathbf{a}_{ik} \leftarrow \big(\alpha_{k,1} + \sum_j p_{ij}^{\text{D}}\,\mathbb{E}_q[z_{ijk}],\ \alpha_{k,2} + \sum_j p_{ij}^{\text{D}}\,\mathbb{E}_q[V_{jk}]\big)$
            $\mathbf{b}_{jk} \leftarrow \big(\beta_{k,1} + \sum_i p_{ij}^{\text{D}}\,\mathbb{E}_q[Z_{ijk}],\ \beta_{k,2} + \sum_i p_{ij}^{\text{D}}\,\mathbb{E}_q[U_{ik}]\big)$
    M-STEP
        Drop-out hyper-parameters
            $\pi_j^{\text{D}} = \frac{1}{n}\sum_{i=1}^n p_{ij}^{\text{D}}$
        Gamma hyper-parameters
            $\alpha_{k,1} \leftarrow \psi^{-1}\big(\log(\alpha_{k,2}) + \frac{1}{n}\sum_{i=1}^n \mathbb{E}_q[\log(U_{ik})]\big)$
            $\alpha_{k,2} \leftarrow n\,\frac{\alpha_{k,1}}{\sum_{i=1}^n \mathbb{E}_q[U_{ik}]}$
            $\beta_{k,1} \leftarrow \psi^{-1}\big(\log(\beta_{k,2}) + \frac{1}{p}\sum_{j=1}^p \mathbb{E}_q[\log(V_{jk})]\big)$
            $\beta_{k,2} \leftarrow p\,\frac{\beta_{k,1}}{\sum_{j=1}^p \mathbb{E}_q[V_{jk}]}$
until Convergence
return $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathbf{a}$, $\mathbf{b}$, $(\pi_j^{\text{D}})_j$ and $(p_{ij}^{\text{D}})_{ij}$
```

Algorithm 6.3: Variational-EM algorithm to infer the ZI-GaP factor model

follows the Gamma distribution whatever the value of the spike-and-slab indicator S_{jk}. The contribution or absence of contribution from the variable j in the component k is accounted for in the conditional Poisson distribution of X_{ij}, i.e. especially:

$$X_{ij}\,|(U_{ik},\widetilde{V}_{jk},S_{jk})_k \sim \mathscr{P}(\textstyle\sum_k S_{jk}\,U_{ik}\,\widetilde{V}_{jk})$$

Following this definition, the conditional distribution of each Z_{ijk} is:

$$Z_{ijk}\,|U_{ik},\widetilde{V}_{jk},S_{jk} \sim \mathscr{P}(S_{jk}\,U_{ik}\,\widetilde{V}_{jk})$$

Thus, the sparse-GaP model is defined as follows:

$$\begin{aligned}
&X_{ij} = \sum_k Z_{ijk}\,,\\
&Z_{ijk}\,|U_{ik},\widetilde{V}_{jk},S_{jk} \sim \mathscr{P}(S_{jk}\,U_{ik}\,\widetilde{V}_{jk})\,,\\
&U_{ik} \sim \Gamma(\alpha_{k,1},\alpha_{k,2})\,,\\
&\widetilde{V}_{jk} \sim \Gamma(\beta_{k,1},\beta_{k,2})\,,\\
&S_{jk} \sim \mathcal{B}(\pi^{\mathrm{s}}_k)\,,
\end{aligned} \tag{6.18}$$

with conditional independence between the Z_{ijk} and independence between the factors U_{ik} and $\widetilde{V}_{jk}$. The spike-and-slab indicators S_{jk} are assumed to be independent from U_{ik} and $\widetilde{V}_{jk}$.

We follow the same path as previously to infer the model, we define the variational distribution q:

$$\begin{aligned}
q(\mathbf{U},\mathbf{V},\mathbf{Z},\mathbf{S}) = \;& \prod_{i=1}^{n}\prod_{k=1}^{K} q(u_{ik}\,|\,\mathbf{a}_{ik}) \times \prod_{j=1}^{p}\prod_{k=1}^{K} q(\widetilde{v}_{jk}\,|\,\mathbf{b}_{jk})\\
&\times \prod_{i=1}^{n}\prod_{j=1}^{p} q\big((z_{ijk})_k\,|\,(r_{ijk})_k\big) \times \prod_{j=1}^{p}\prod_{k=1}^{K} q(S_{jk}\,|\,p^{\mathrm{s}}_{jk})\,,
\end{aligned} \tag{6.19}$$

under assumptions of independence and based on the respective complete conditional distributions.

Complete conditional distributions

In order to derive the inference algorithm, we consider the complete conditional of each latent variable in the model. As previously, we use variational inference to derive the E-step of the EM algorithm. The complete conditional regarding $(Z_{ijk})_k$ is again a

Multinomial distribution, that depends on $\sum_k S_{jk}\, u_{ik} v_{jk}$. The contribution of the factor $\widetilde{V}_{jk}$ is regulated by the indicator S_{jk}, hence:

$$(Z_{ijk})_k \,|\text{—} \sim \mathcal{M}\Big(X_{ij}, (\rho_{ijk})_k\Big)\,, \tag{6.20}$$

where $\rho_{ijk} = \frac{S_{jk}\, u_{ik}\, \widetilde{v}_{jk}}{\sum_\ell S_{j\ell}\, u_{i\ell}\, \widetilde{v}_{j\ell}}$. For a determined variable j, if the S_{jk} are null for all k, the vector $(Z_{ijk})_k$ is deterministic and takes null values.

Thanks to the conjugacy in the model, the complete conditionals of the factors U_{ik} and $\widetilde{V}_{jk}$ remain Gamma distributions. They are respectively parametrized as:

$$\begin{aligned} U_{ik} \,|\text{—} &\sim \Gamma(\alpha_{k,1} + \textstyle\sum_j S_{jk}\, z_{ijk},\ \alpha_{k,2} + \textstyle\sum_j S_{jk}\, \widetilde{v}_{jk})\,, \\ \widetilde{V}_{jk} \,|\text{—} &\sim \Gamma(\beta_{k,1} + \textstyle\sum_i S_{jk}\, z_{ijk},\ \beta_{k,2} + \textstyle\sum_i S_{jk}\, u_{ik})\,. \end{aligned} \tag{6.21}$$

Indeed, as in the zero-inflated, the proof uses the Bayes rule and the conditional distribution $Z_{ijk}\,|U_{ik}, \widetilde{V}_{jk}, S_{jk} \sim \mathscr{P}(S_{jk}\, U_{ik}\, \widetilde{V}_{jk})$, so that the complete conditional of V_{jk} for example verifies:

$$p(\widetilde{v}_{jk} \,|\text{—}) \propto (\widetilde{v}_{jk})^{\beta_{k,1} - 1 + S_{jk} \sum_i z_{ijk}}\, e^{-\widetilde{v}_{jk}\, (\beta_{k,2} + S_{jk}\, \sum_i u_{ik})}\,.$$

Concerning the complete conditional of the spike-and-slab indicator S_{jk}, thanks to the Bayes rules, we can show that

$$p(S_{jk} \,|\text{—}) \propto p(S_{jk}) \times \textstyle\prod_i p(Z_{ijk} \,|\, U_{ik}, \widetilde{V}_{jk}, S_{jk})\,,$$

Indeed, only the Poisson variables $(Z_{ijk})_i$ depend on S_{jk}. Thus, the formulation of the complete conditional is explicit:

$$\begin{aligned} p(S_{jk} \,|\text{—}) \propto \quad & (\pi^{\mathrm{s}}_j)^{S_{jk}}\, (1 - \pi^{\mathrm{s}}_j)^{1 - S_{jk}} \\ & \times \textstyle\prod_i \exp(-S_{jk}\, U_{ik}\, \widetilde{V}_{jk})\ (S_{jk}\, U_{ik}\, \widetilde{V}_{jk})^{Z_{ijk}} \times \dfrac{1}{Z_{ijk}\,!}\,. \end{aligned} \tag{6.22}$$

Derivation of the algorithm

The variational-EM algorithm for the sparse-GaP factor model is derived as follows (c.f. Algorithm 6.4). As previously, in the E-step, we compute the stationary point of the ELBO. In particular, based on Equation (6.21), for U_{ik}, V_{jk}, we have respectively:

$$\begin{aligned} \mathbf{a}_{ik} &= \mathbb{E}_q\big[(\alpha_{k,1} + \textstyle\sum_j S_{jk}\, Z_{ijk},\ \alpha_{k,2} + \textstyle\sum_j S_{jk}\, \widetilde{V}_{jk})^T\big]\,, \\ \mathbf{b}_{jk} &= \mathbb{E}_q\big[(\beta_{k,1} + S_{jk} \textstyle\sum_i Z_{ijk},\ \beta_{k,2} + S_{jk} \textstyle\sum_i U_{ik})^T\big]\,. \end{aligned} \tag{6.23}$$

Regarding Z_{ijk}, based on Equation (6.20), we should compute the variational parameter as $\log(r_{ijk}) = \mathbb{E}_q[\log(\rho_{ijk})]$. The issue here is that ρ_{ijk} is computed as:

$$\rho_{ijk} = \frac{S_{jk}\, u_{ik}\, \widetilde{v}_{jk}}{\sum_\ell S_{j\ell}\, u_{i\ell}\, \widetilde{v}_{j\ell}}$$

Therefore, $\mathbb{E}_q[\log(\rho_{ijk})]$ involves the expectation $\mathbb{E}_q[\log(S_{jk}\, U_{ik} \widetilde{V}_{jk})]$ which is not tractable because of the indicator S_{jk}. To overcome this issue, we update r_{ijk} as follows:

$$r_{ijk} = \frac{\widehat{S}_{jk} \exp\Big(\mathbb{E}_q[\log(U_{ik})] + \mathbb{E}_q[\log(\widetilde{V}_{jk})]\Big)}{\sum_\ell \widehat{S}_{jk} \exp\Big(\mathbb{E}_q[\log(U_{i\ell})] + \mathbb{E}_q[\log(\widetilde{V}_{j\ell})]\Big)}\,, \tag{6.24}$$

where $\widehat{S}_{jk}$ indicates at the current iteration if the variable j contributes to factor k, i.e. if the current value of variational parameter p^{S}_{jk} is closer to 0 or to 1. In practice, we set $\widehat{S}_{jk}$ as:

$$\widehat{S}_{jk} = \mathbf{1}_{\{p^{\text{S}}_{jk} > \pi_{\text{thr}}\}}\,,$$

where $\pi_{\text{thr}} \in (0,1)$. This corresponds to thresholding the contribution of V_{jk} in the Poisson distribution. On this matter, sparsity-inducing approaches based on a penalized optimization are generally based on a tuning parameter that regulates the sparsity of the estimates (c.f. Chapter 1). Although the threshold π_{thr} also has to be chosen by the user, the interest here is that π_{thr} has a direct interpretation in the model, as the minimal frequency at which a recorded variable j contributes to the latent factors. In the algorithm, we choose $\pi_{\text{thr}} = 0.5$.

Concerning S_{jk}, the natural parametrization of the Bernoulli distribution is based on the logit of the probability of success. Therefore, based on Equation (6.22), the stationary point is defines as:

$$\begin{aligned} \text{logit}(p^{\text{S}}_{jk}) = \;& \text{logit}(\pi^{\text{S}}_j) - \textstyle\sum_i \big\{\mathbb{E}_q[U_{ik}]\, \mathbb{E}_q[\widetilde{V}_{jk}] \\ & + \mathbb{E}_q[Z_{ijk}] \left(\mathbb{E}_q[\log(U_{ik})] + \mathbb{E}_q[\log(\widetilde{V}_{jk})]\right)\big\}\,. \end{aligned} \tag{6.25}$$

In the M-step of the variational algorithm, the hyper-parameter π^{S}_j is again updated with the mean of the corresponding variational parameters:

$$\pi^{\text{D}}_j = \frac{1}{p} \sum_{j=1}^{p} p^{\text{S}}_{jk}$$

It can be noted that this update is similar to what would be obtained when considering π^{S}_j as a latent variable with a Beta prior and by inferring the parameter of this prior in the variational E-step.

Eventually, the updates of the prior hyper-parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ over $\mathbf{U}$ and $\mathbf{V}$ are not affected, thanks to the independence between the spike-and-slab indicator S_{jk} and U_{ik} or $\widetilde{V}_{jk}$.

```
Input: /
Output: Estimates of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathbf{a}$, $\mathbf{b}$, $\mathbb{E}_q[\mathbf{U}]$, $\mathbb{E}_q[\mathbf{V}]$, $(\pi_j^{\text{D}})_j$ and $(p_{ij}^{\text{D}})_{ij}$
Initialize $\boldsymbol{\alpha} = [(\alpha_{k,1}, \alpha_{k,2})] \in \mathbb{R}^{K \times 2}$, $\boldsymbol{\beta} = [(\beta_{k,1}, \beta_{k,2})] \in \mathbb{R}^{K \times 2}$
Initialize $\mathbf{a} = [(a_{ik,1}, a_{ik,2})] \in \mathbb{R}^{n \times K \times 2}$, $\mathbf{b} = [(b_{jk,1}, b_{jk,2})] \in \mathbb{R}^{p \times K \times 2}$
Initialize $(\pi_j^{\text{D}})_j$ and $(p_{ij}^{\text{D}})_{ij}$
repeat
    E-STEP
        Spike-and-slab indicators
            $\widehat{S}_{jk} = \mathbf{1}_{\{p_{jk}^{\text{S}} > \pi_{\text{thr}}\}}$
        Multinomial variational parameters
            $r_{ijk} = \frac{\widehat{S}_{jk} \exp\left(\mathbb{E}_q[\log(U_{ik})] + \mathbb{E}_q[\log(\widetilde{V}_{jk})]\right)}{\sum_\ell \widehat{S}_{jk} \exp\left(\mathbb{E}_q[\log(U_{i\ell})] + \mathbb{E}_q[\log(\widetilde{V}_{j\ell})]\right)}$
        Gamma variational parameters
            $\mathbf{a}_{ik} \leftarrow \big(\alpha_{k,1} + \sum_j p_{jk}^{\text{S}}\, \mathbb{E}_q[z_{ijk}],\ \alpha_{k,2} + \sum_j p_{jk}^{\text{S}}\, \mathbb{E}_q[\widetilde{V}_{jk}]\big)$
            $\mathbf{b}_{jk} \leftarrow \big(\beta_{k,1} + p_{jk}^{\text{S}} \sum_i \mathbb{E}_q[z_{ijk}],\ \beta_{k,2} + p_{jk}^{\text{S}} \sum_i \mathbb{E}_q[U_{ik}]\big)$
        Spike-and-slab variational parameters
            $p_{jk}^{\text{S}} = \text{logit}^{-1}\Big(\text{logit}(\pi_j^{\text{S}}) - \sum_i \big\{\mathbb{E}_q[U_{ik}]\, \mathbb{E}_q[\widetilde{V}_{jk}]$
                    $+ \mathbb{E}_q[Z_{ijk}]\, \big(\mathbb{E}_q[\log(U_{ik})] + \mathbb{E}_q[\log(\widetilde{V}_{jk})]\big)\big\}\Big)$
    M-STEP
        Spike-and-slab hyper-parameters
            $\pi_j^{\text{S}} = \frac{1}{p} \sum_{i=1}^n p_{jk}^{\text{S}}$
        Gamma hyper-parameters
            $\alpha_{k,1} \leftarrow \psi^{-1}\big(\log(\alpha_{k,2}) + \frac{1}{n} \sum_{i=1}^n \mathbb{E}_q[\log(U_{ik})]\big)$
            $\alpha_{k,2} \leftarrow n\, \frac{\alpha_{k,1}}{\sum_{i=1}^n \mathbb{E}_q[U_{ik}]}$
            $\beta_{k,1} \leftarrow \psi^{-1}\big(\log(\beta_{k,2}) + \frac{1}{p} \sum_{j=1}^p \mathbb{E}_q[\log(\widetilde{V}_{jk})]\big)$
            $\beta_{k,2} \leftarrow p\, \frac{\beta_{k,1}}{\sum_{j=1}^p \mathbb{E}_q[\widetilde{V}_{jk}]}$
until Convergence
return $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathbf{a}$, $\mathbf{b}$, $(\pi_j^{\text{S}})_j$ and $(p_{jk}^{\text{S}})_{jk}$
```

Algorithm 6.4: Variational-EM algorithm to infer the sparse-GaP factor model

6.3 Empirical study of the Gamma-Poisson factor model

We proposed different Gamma-Poisson factor models that correspond to different count matrix factorization problems, with different constraints depending on the type of the data (zero-inflated or sparse). We will now assess the performance of our methods for dimension reduction and data exploration. We will compare our results with some standard approaches that we introduced in the previous chapters.

All our computational experiments were processed within the `R` programming environment. For performance considerations, our own algorithms are implemented in the `C++` language and interfaced with `R`. The SVD and PCA are natively implemented in `R`. The different approaches of Non-negative Matrix Factorization (NMF) that we will consider are implemented in the `NMF` `R`-package. We will especially focus on the Poisson-NMF and the ls-NMF methods. To avoid issues linked to bad local optima during the optimization, the implementation in the `NMF` package is based on multiple runs of the considered algorithm on the data matrix $\mathbf{X}$ with different random initializations, so that the estimated factors $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$ are the averaged estimations over the multiple runs. The questions about computational efficiency are discussed in Appendix Section D.4.1.

We first focus on standard Gamma-Poisson factor models. Then, we will study the behavior of our inference algorithm in the case of zero-inflated data.

6.3.1 Generation of the data

In the different experiments, the data are simulated according to the generative process associated with the GaP factor model. However, we set the hyper-parameters to artificially create some groups of observations and variables. Thus, we will be able to check if a method succeeds to reconstruct this latent structure. Precisely, the data are generated thanks to the following scheme:

- The n individuals are divided into N groups denoted by $\mathcal{U}_1, \dots, \mathcal{U}_N$. The p variables are divided into P groups, denoted by $\mathcal{V}_1, \dots, \mathcal{V}_P$.
- The factors U_{ik} are generated following the Gamma distribution $\Gamma(\alpha_{k,1}, \alpha_{k,2})$. The values of $\alpha_{k,1}$ and $\alpha_{k,2}$ are fixed in each group of individuals $\mathcal{U}_g$, i.e. for any i in $\mathcal{U}_g$, the factors U_{ik} are drawn following the same Gamma prior (for $g = 1, \dots, N$).
- Similarly, the factors V_{jk} are drawn from the Gamma distribution $\Gamma(\beta_{k,1}, \beta_{k,2})$.

The values of $\beta_{k,1}$ and $\beta_{k,2}$ are fixed in each group of variables $\mathcal{V}_g$, i.e. for any j in $\mathcal{V}_g$, the factors V_{jk} are drawn following the same Gamma prior (for $g = 1, \ldots, P$).

- The observations X_{ij} are generated following the corresponding Poisson distribution $\mathscr{P}(\sum_k u_{ik}\, v_{jk})$.

When considering unsupervised statistical approaches, especially for data exploration, the question about assessing the efficiency of a method to represent the data is tricky, since there is no quantity as the error of prediction in supervised problem to check how an algorithm behaves depending on the experimental conditions. The partitioning of the individuals and variables into different groups in our simulation process is an artificial framework to enforce the underlying structure of the data, so that it becomes possible to verify if an algorithm is able to catch this specific structure, i.e. if it represents the data in a subspace that discriminates between the groups of individuals and variables. This behavior would be expected when analyzing an experimental data set where the underlying structure is unknown.

6.3.2 A criterion for comparison?

Comparing different approaches for matrix factorization raises questions about the choice of a criterion that quantifies the quality of a method. Since we are trying to approximate the data matrix $\mathbf{X}$ by the factor product $\mathbf{U}\mathbf{V}^T$, measuring the distance between $\mathbf{X}$ and $\widehat{\mathbf{U}}\widehat{\mathbf{V}}^T$ would be a good indication of the efficiency of the considered approach regarding dimension reduction. As introduced in Chapter 5, the choice of a metric implies to choose an underlying geometry that would be appropriate for the data.

Bregman divergence

We decided to not focus on the Euclidean geometry (i.e. ℓ_2-metric) since it is not suitable for count data. Instead, we focus on the Bregman divergence[14] $D(\mathbf{X} \,|\, \widehat{\mathbf{U}}\widehat{\mathbf{V}}^T)$ defined in Chapter 5 that is a generalization of the ℓ_2 distance to the Poisson distribution.

The Bregman divergence can also be interpreted as the deviance between the estimated Poisson model and the saturated Poisson model. The saturated model is the model where each λ_{ij} is directly estimated by the corresponding observations x_{ij}, so the deviance is defined as

$$\mathrm{Dev}(\mathbf{X}, \widehat{\mathbf{U}}\widehat{\mathbf{V}}^T) = -2 \times \left(\log p(\mathbf{X} \,|\, \mathbf{\Lambda} = \widehat{\mathbf{U}}\widehat{\mathbf{V}}^T) - \log p(\mathbf{X} \,|\, \mathbf{\Lambda} = \mathbf{X}) \right),$$

[14] $D(\mathbf{X} \,|\, \boldsymbol{\lambda}) = \sum_{i=1}^{n} \sum_{j=1}^{p} x_{ij} \, \log\left(\frac{x_{ij}}{\lambda_{ij}}\right) - x_{ij} + \lambda_{ij}$

with $\log p(\mathbf{X} \,|\, \mathbf{\Lambda})$ the Poisson log-likelihood, thus $\text{Dev}(\mathbf{X}, \widehat{\mathbf{U}}\widehat{\mathbf{V}}^T) \propto D(\mathbf{X} \,|\, \widehat{\mathbf{U}}\widehat{\mathbf{V}}^T)$. Regarding this matter, Landgraf & Lee (2015) proposed a generalization of the PCA to the exponential family that is based on this deviance criterion.

Ordering the factors

In a specific model with K factors, we question how the different matrices $\widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T$ approximate $\mathbf{X}$ depending on $k = 1, \dots, K$. We recall that the matrix $\widehat{\mathbf{U}}_{1:k}$ is the matrix with the first k columns of $\widehat{\mathbf{U}}$, similarly for $\widehat{\mathbf{V}}_{1:k}$ with the first k columns of $\widehat{\mathbf{V}}$. Indeed, in the context of the PCA, the algorithm gives all the models with $K = 1, \dots, \text{rank}(\mathbf{X})$ in a single run because the models with an increasing K are nested and there exists a natural order of the factors (by decreasing explained variability). However, it is not the case in the GaP factor model, or even in the Poisson factor model associated with the Poisson-NMF. The factors are not ordered and the model are not nested.

Such remark raises two issues. On the one hand, the number of factors K has to be carefully chosen when fitting the model, because the model with K factors is not included in the model with $K+1$ factors. This question will be treated in the next section. On the other hand, because of the absence of order between factors, it implies that the model is identifiable up to a permutation of the K factors[15]. In order to avoid any problem when comparing the different models, we choose to order the factor according to the cumulative Bregman divergence, defined as:

$$k \mapsto D\big(\mathbf{X} \,|\, \widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T\big) \,. \tag{6.26}$$

We precise that we also reorder the factors computed by the Poisson-NMF following the same criterion, because the order is not set in the ouput of the functions from the `NMF` package. Similarly, the factors computed by the ls-NMF are reordered following the cumulative Euclidean metric $k \mapsto \|\mathbf{X} - \widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T\|_F^2$ since it is based on this least squares criterion. As previously mentioned, the factors from the SVD or the PCA do not require to be reordered.

[15] i.e. by permuting the columns in $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$ according to the same reordering.

Reconstruction of the signal in the standard Gamma-Poisson factor model

We compare both our variational algorithms (EM and standard) to the Poisson-NMF (based on $D(\mathbf{X} \mid \widehat{\mathbf{U}}\widehat{\mathbf{V}}^T)$) and the ls-NMF (based on the Euclidean distance). We study the evolution of the Bregman divergence between $\mathbf{X}$ and $\widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T$ depending on $k = 1 \ldots, K$ as defined in Equation (6.26), for models with different numbers of factors, i.e. $K = 10, 15, 20$. We investigate different configurations of data with $n = 100$, $K^* = 10$ and $p = 50, 100, 300, 500$. In particular, we generate 50 different data sets for each configuration and run the different methods on each one with different numbers of factors $K = 10, 15, 20$.

We precise that we only display the results from the Poisson-NMF and not the results from the ls-NMF because they were very similar. Indeed, whereas the ls-NMF is based on a least squares approximation, it produces a reconstructed matrix $\widehat{\mathbf{U}}\widehat{\mathbf{V}}^T$ that fits the data regarding the Bregman divergence as well as the Poisson-NMF.

Figure 6.3 shows the results. A first comment is that the factors estimated by the variational-EM algorithm better approximate the data than the factors estimated by the standard variational algorithm. This point again highlights the interest to estimate the hyper-parameters in the variational framework (c.f. previous section). Then, it appears clearly that the inference of the GaP factor model gives better results regarding the reconstruction of the data than the Poisson-NMF in all data configurations. This empirical result is expected since the data are generated under the GaP factor model. Nonetheless, it remains necessary to check that the algorithm behaves as expected.

A second comment concerns the evolution of $D\big(\mathbf{X} \mid \widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T\big)$ depending on k. For any total number of K factors in the model, the Bregman divergence reaches a plateau when k is near the true K^*. This point highlights the particular abilities of our variational algorithm based on the GaP factor model for compression. As discussed in the next section, this criterion may be an option to choose the number K of factors.

6.3.3 Choice of the number of factors

As noticed in the previous section, the GaP factor model with an increasing number K of factors are not nested[16]. Thus, the choice of K is very sensitive. If K is chosen too small, the model looses some information with respect to the data. If K is too large, the number of parameters increases, potentially leading to over-parametrization issue, and the effect of the different factors are softened in the masses.

[16] The model associated to the NMF presents the same properties.

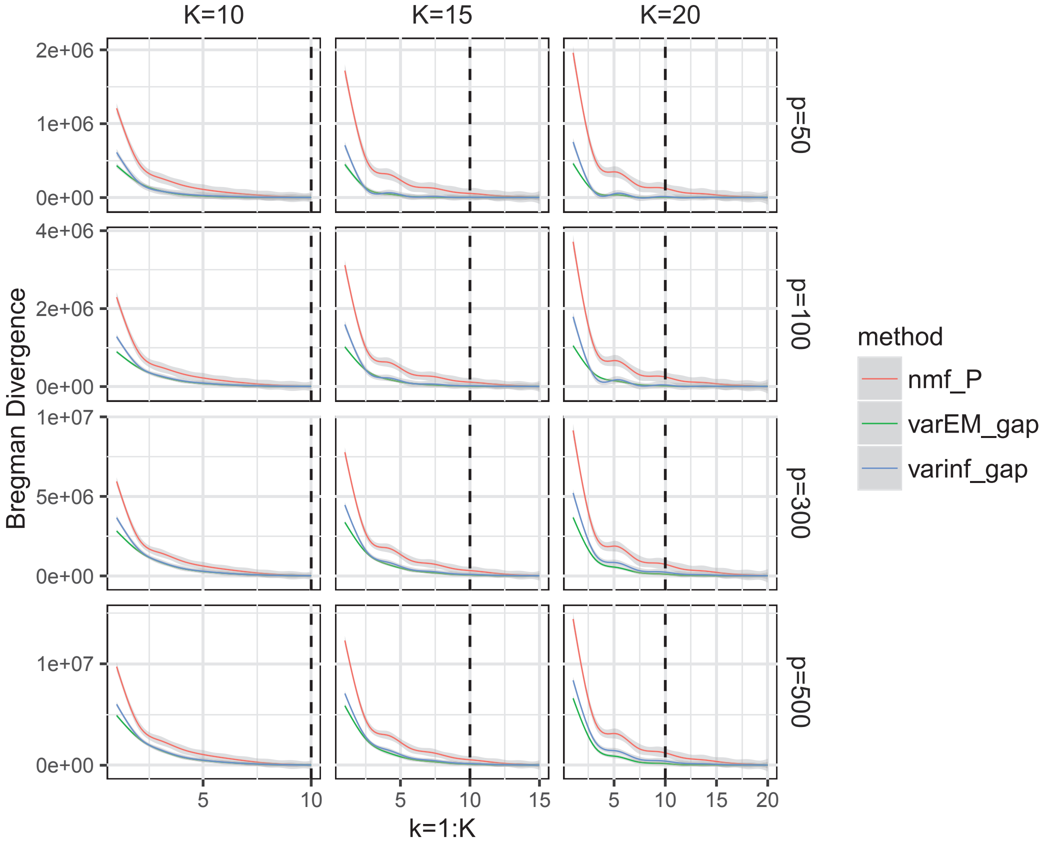


Figure 6.3 – Bregman divergence between $\mathbf{X}$ and $\widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T$ depending on $k = 1 \ldots, K$. The data are generated with $n = 100$, $K^* = 10$ (represented by the vertical dashed line) and different values $p = 50, 100, 300, 500$. The different methods are the Poisson-NMF (`nmf_P`) and the two variational algorithms (`varEm_gap` and `varinf_gap`). For each configuration, 50 data sets are generated and fitted. The line corresponds to the average Bregman divergence over the 50 repetitions with the confidence bandwidth in shaded grey.

Closest fit to the data

Moreover, in contrast to PCA, testing different values of K requires to fit different models. For instance, Knowles & Ghahramani (2007) or Bhattacharya & Dunson (2011) considered models with an infinite number of factors[17]. Their algorithms both estimate at which finite level the number K of factors should be reduced (as any other parameters). However, such approaches require more computation time as they explore a large range of values for K.

We decided to remain in the paradigm of finite models to reduce the computation cost. A first approach to choose the number of factors is to fit a model with a large K and verify how the matrix $\widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T$ reconstructs $\mathbf{X}$ depending on k. This approach is for instance widely used in PCA by checking the proportion of variability explained by each component, which is inherently linked to the ordered singular values of the matrix $\mathbf{X}_c$[18]. In our context, the explained variance is not the criteria that we will consider as it inherently linked to the Gaussian distribution and the Euclidean geometry. In particular, as mentioned in the previous section, the fit of the model to the data can be measured by the Bregman divergence $k \mapsto D\big(\mathbf{X} \,|\, \widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T\big)$ for a model with $k = 1, \dots, K$ factors. Furthermore, this measure can be used to empirically find the best K to fit the model, based on an elbow criterion, i.e. to find the values of K from where adding new factors does not improve $D\big(\mathbf{X} \,|\, \widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T\big)$. This determination is however not always unambiguous and may sometimes lead to some over-fitting, i.e. when considering too many factors.

To overcome these potential issues, we are currently working on a model selection procedure to choose K, as we will discuss in Chapter 7.

[17] They both worked in the Gaussian framework.

[18] The reader may refer to the PhD manuscript of Chloé Friguet (2010) p. 96 for a review of the different criteria to choose K in this context.

An automatic choice of K?

In practice the question of the choice of K may be directly resolved by the variational-EM algorithm. Indeed, we observe on our simulations that there exists a permutation of the columns $\widehat{\mathbf{u}}_k$ of $\widehat{\mathbf{U}}$ such that the norm of the vector $\widehat{\mathbf{u}}_k$ is set to zero as soon as k becomes higher than a certain threshold. In other words, the procedure of inference seems to recognize the unnecessary components and ensures that the observations i do not take part into this supplementary factors.

We checked the evolution of $\|\widehat{\mathbf{u}}_k\|_2$, depending on $k = 1, \dots, K$ for different models (with different K). Figure 6.4 shows the results in the case $n = 100$, $p = 100$ and $K^* = 10$. We observe that, when $\mathbf{U}$ is estimated by our variational-EM algorithm with $K > K^*$ factors, the value[19] of $\|\widehat{\mathbf{u}}_k\|_2$ tends toward 0 when $k > K^*$. On the contrary, if the number K of factors in the model is chosen to be smaller than the true K^*, the norm of $\|\widehat{\mathbf{u}}_k\|_2$ is not shrunk toward zero for any $k = 1, \dots, K$. This behavior is not observed for the other approaches: Poisson-NMF, ls-NMF and the standard variational algorithm in the GaP factor model.

The variational-EM algorithm seems to also learn the number of factors that the model should consider to be appropriate for the data. Dikmen & Févotte (2012) observed a similar behavior in their variational-EM algorithm, although their model is slightly different from our GaP factor model.

6.3.4 Behavior in the presence of zero-inflation

Our interest is mainly to explore zero-inflated single-cell data. Thus, we question the performance of our algorithm to infer a zero-inflated Gamma-Poisson model. We focus on different points that are crucial in data exploration: reconstruction of the true signal, sensitivity to the zero-inflation and reconstruction of the underlying structure. We will especially consider the questions of data visualization and data clustering.

To do so, we simulate the data as follows. We generate a count matrix $\mathcal{X}_{n\times p}$ following the generative process of the standard GaP model, i.e. without zero-inflation. Then, the matrix of drop-out events $\mathbf{D}_{n\times p}$ is generated with $D_{ij} \sim \mathcal{B}(\pi_j^{\mathrm{D}})$. In order to consider realistic zero-inflated data, the probability π_j^{D} depends on the average of the count j in $\mathcal{X}$. In particular, when denoting by $\bar{\mathcal{X}}_j$ the count average of the column j in $\mathcal{X}$, the probability π_j^{D} is defined as:

$$\pi_j^{\mathrm{D}} = 1 - \exp(-\mu\,\bar{\mathcal{X}}_j)\,,$$

[19] The columns of $\mathbf{U}$ are sorted by decreasing values of their ℓ_2 norm, for any K set in the model

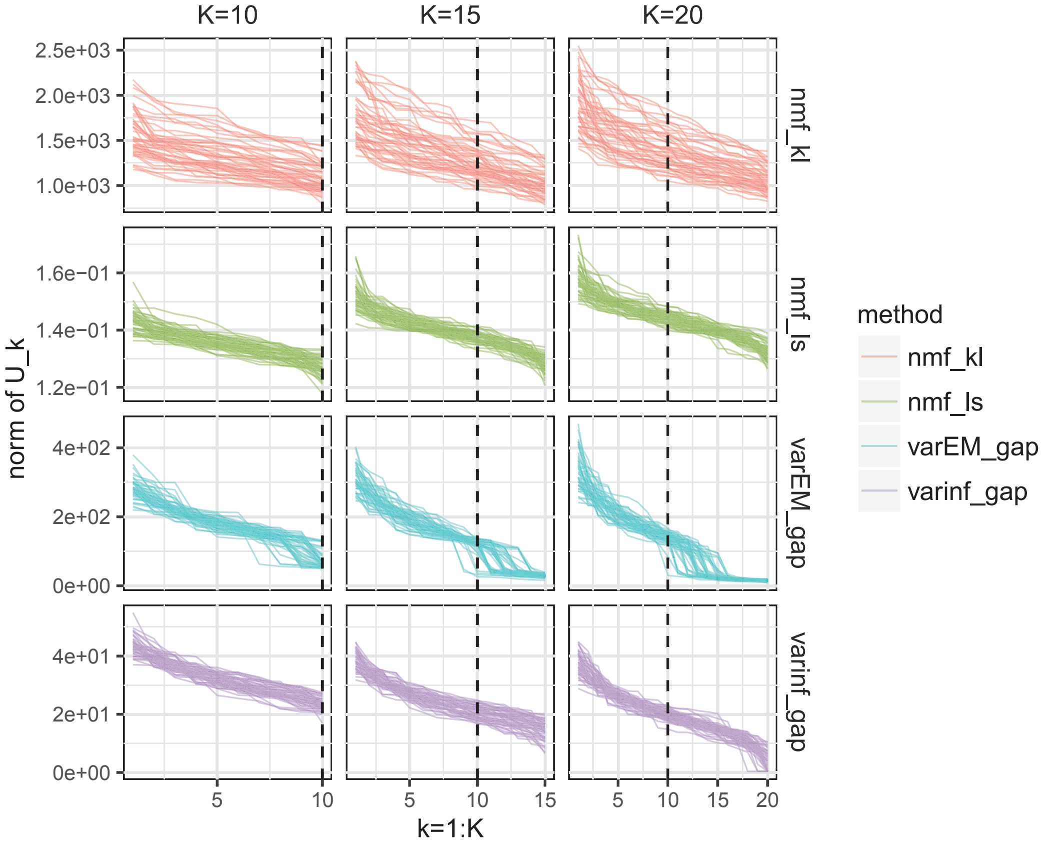


Figure 6.4 – Evolution of $\|\widehat{\mathbf{u}}_k\|_2$ depending on $k = 1, \ldots, K$ for the different methods Poisson-NMF (`nmf_P`), ls-NMF (`nmf_ls`) and the two variational algorithms (`varEm_gap` and `varinf_gap`) when considering $K = 10, 15, 20$ factors in the model. The different trajectories corresponds to the analysis of 50 different data sets generated with $n = 100$, $p = 100$, $K = 10$ (represented by the vertical dashed line). The columns of $\widehat{\mathbf{U}}$ are sorted by decreasing value of their norm.

where $\mu > 0$ is a fixed rate. Thus, the variables j with a lower count average $\bar{\mathcal{X}}_j$ will have a higher probability $1 - \pi_j^{\text{D}}$ of drop-out events. In our simulations, we set μ so that the drop-out probabilities, i.e. $1 - \pi_j^{\text{D}}$, lie between 50% and 90%, which corresponds to an average of 70% over all j. The proportion of drop-out events is therefore high but consistent with current single-cell technologies[20] that catch between 30% and 50% of the genetic material in a single cell (i.e. $\pi_j^{\text{D}} \in [0.3\,;\,0.5]$). Finally, the entries of the zero-inflated data matrix $\mathbf{X}$ (with drop-out events) are constructed as $X_{ij} = \mathcal{X}_{ij} \times D_{ij}$.

Reconstruction of the true signal

The first question when analyzing zero-inflated data is whether or not the considered statistical method is sensitive to the drop-out events, that correspond to corrupted observations. In particular, in our count matrix factorization problem, we want to verify how the true signal $\mathcal{X}$ is reconstructed based on the decomposition of the zero-inflated matrix $\mathbf{X}$. We will especially show that our method that accounts for zero-inflation is less sensitive to drop-out events that other approaches.

In this regard, we simulate zero-inflated data in different configurations with $n = 100$, $K^* = 10$ and $p = 50, 100, 300, 500$. In particular, we again generate 50 different data sets for each configuration and run the different methods on each one with different numbers of factors $K = 10, 15, 20$. All models are fit with the zero-inflated matrix $\mathbf{X}$, however we check how the estimated factors $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$ reconstruct the true uncorrupted signal $\mathcal{X}$. In other words, we study:

$$k \mapsto D\big(\mathcal{X} \,|\, \widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T\big)$$

Thus, we will see which methods are the most sensitive to drop-out events.

Figure 6.5 shows the results. We fit the different models for different number of factors $K = 10, 15, 20$. When comparing the variational-EM algorithm for the ZI-GaP model and the two NMF approaches (Poisson-NMF and ls-NMF), we clearly observe that our method reconstructs the true signal $\mathcal{X}$ with a better efficiency.

The reader should note that the Bregman divergence are presented in log-scale because of the huge differences between the three methods. When considering a standard scale, the shape of the curves are similar to the ones in Figure 6.3, with a high decrease of $k \mapsto D\big(\mathcal{X} \,|\, \widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T\big)$ for small k before reaching a plateau when k becomes bigger than the true K^*. The compression abilities of our methods are again highlighted in the zero-inflated case. Indeed, the factors estimated by our variational-EM algorithm efficiently reconstruct the true signal as soon as $k > 5$, even if the true number of factors

[20] c.f. Section 4.2.1 in Chapter 4

is $K^* = 10$. Besides, to choose the number of factors K, we can again check the norm of the columns $\widehat{\mathbf{u}}_k$ of $\widehat{\mathbf{U}}$ as it tends toward zero when k is near or bigger the true value K^*. The figure is joined in Appendix Section D.3.1.

In addition, it can be noted that, even when considering the Euclidean metric, our model-based approach is better to reconstruct the uncorrupted signal than least-squares-based methods as the SVD or the ls-NMF. This result is detailed in Appendix Section D.3.2 and confirms that, as expected, the Euclidean geometry is not appropriate for zero-inflated count data.

Visualization of zero-inflated data

A standard issue with high dimensional data concerns the question of data visualization. Indeed, the first step when exploring data is to try to visualize them in 2 dimensions (or at most 3 dimensions), in order to understand the underlying organization of the data, for instance regarding the potential existence of groups of observations or groups of variables. In this case, an option would be to use a method that projects the data in a 2-dimensional space so that the underlying structure is summarized in 2 dimensions.

A method widely used for data visualization is the PCA. Thus, a visualization tool is to consider the graph of the observation coordinates regarding the first two components. In our context of matrix factorization, we recall that it corresponds to constructing the graph of the scatter plot $(\widehat{u}_{i1}, \widehat{u}_{i2})_{i=1,\dots,n}$ from the coordinates of the first two columns $\widehat{\mathbf{u}}_1$ and $\widehat{\mathbf{u}}_2$ of the matrix $\widehat{\mathbf{U}}$.

Instead of using PCA to construct the matrix $\widehat{\mathbf{U}}$, we can use other approaches as the NMF or our method based on the GaP factor model (or the ZI-GaP factor model in the context of zero-inflated data). In particular, we assess the ability of the different methods for data visualization in the case of zero-inflated data. We will see that our variational framework with the specific ZI-GaP factor model appears to be an interesting tool for data visualization.

We generate a data set $\mathbf{X}$ with $n = 100$ and $p = 1000$ with drop-out events as in the previous section. We set $K^* = 20$ and simulate different configurations with 2, 3 or 4 groups of observations by setting different values of hyper-parameters between the different groups (c.f. Section 6.3.1). We run each method with $K = 2$ as we want to represent the first two factors. In particular, we consider the PCA applied to the data transformed thanks to the Anscombe transform[21], the Poisson-NMF and our method.

[21] By experience with count data, the Anscombe transform generally gives better results than the $\log(\text{count} + 1)$ transform or than not transforming the count data.

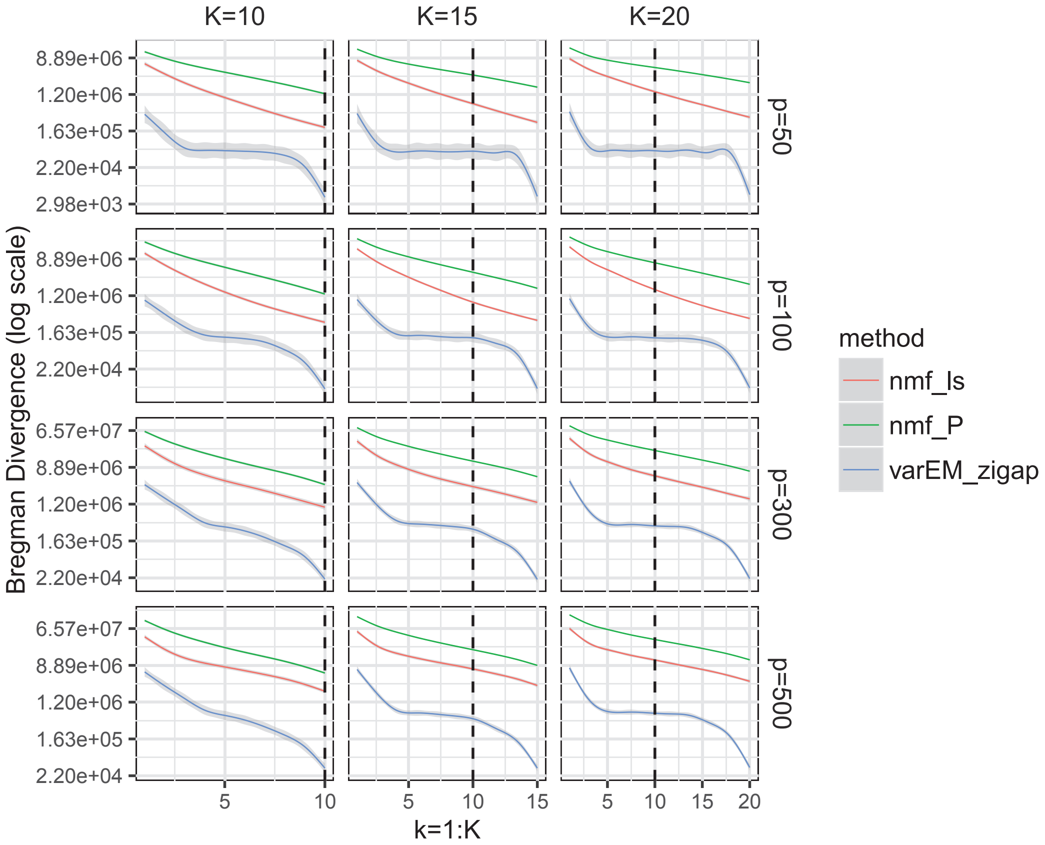


Figure 6.5 – Bregman divergence $D(\mathcal{X}\,|\,\widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T)$ between the true $\mathcal{X}$ and $\widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T$ (learned with the zero-inflated data $\mathbf{X}$) depending on $k = 1\,\ldots, K$. The data are generated with $n = 100$, $K^* = 10$ (represented by the vertical dashed line) and different values $p = 50, 100, 300, 500$. The different methods are the Poisson-NMF (`nmf_P`), the ls-NMF (`nmf_ls`) and the variational-EM algorithm for the ZI-GaP model (`varEm_zigap`). For each configuration, 50 data sets are generated and fitted. The line corresponds to the average Bregman divergence over the 50 repetitions with the confidence bandwidth in shaded grey. The y-axis is in log-scale.

Figure 6.6 shows the representation of the observations regarding the first two factors for the different methods and depending on the different data configurations (2, 3 or 4 groups). In the case of 2 groups, the factors **U** estimated by our ZI-specific variational algorithm clearly identify the 2 groups of observations. Despites the zero-inflation, the 2 groups also appear in the factors estimated by the NMF based on a Poisson model (Poisson-NMF). In contrast, the principal components from the PCA does not clearly highlight the 2 groups, as the colored points are not clearly separated. In a real experiment, the PCA would not make it possible to identify the organization of the individuals.

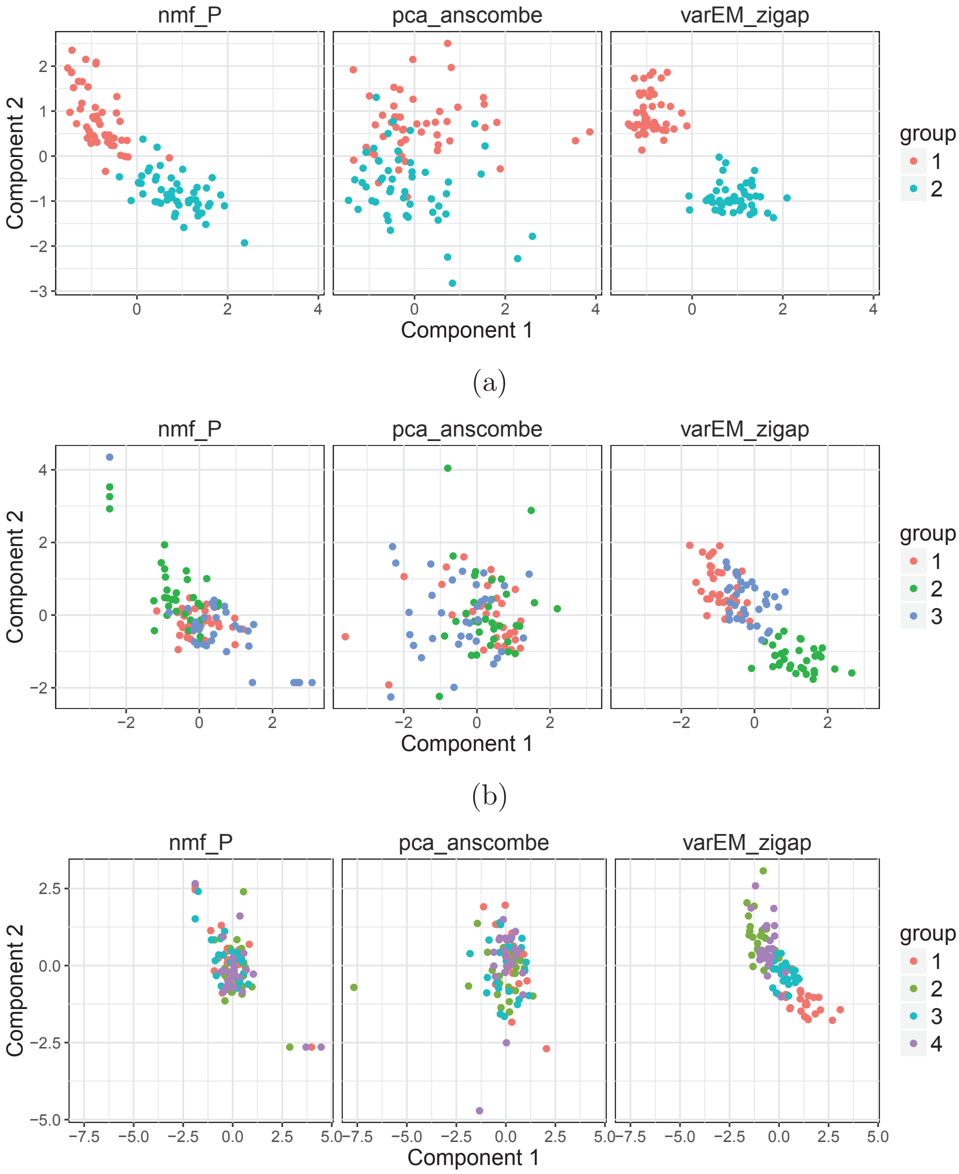
In the case of 3 or 4 groups, the factors from the PCA or from the Poisson-NMF totally mix the observations, and it becomes totally impossible to distinguish the groups. These results illustrate the importance to account for the zero-inflation when analyzing such data. Indeed, our ZI-specific approach catches the underlying structure since the groups still appear on the graph. When the number of groups increases, the distinction between the scatter plot from each group becomes less and less clear and some points are mixed with the wrong groups. Nonetheless, the group organization still remains and the groups are not totally mixed.

In practice, finding a representation of high-dimensional data in two dimensions that shows the underlying organization of the data is a huge challenge. Indeed, a 2-dimensional space is certainly not sufficient to identify complex latent structures with numerous groups of observations. Our point here is that a method that accounts for the specificity of the data, as the zero-inflation, will more likely be able to identify the dependencies and the diversity within a data set, so that such methods should be more widely used when analyzing such data, or at least should be considered in addition to the PCA. As we will see in a moment, we may also consider other approaches for data exploration than 2-dimensional (or 3-D) visualization.

Clustering

In unsupervised problems, the objective is generally to identify clusters of observations or variables, i.e. in our applications, clusters of cells or genes. Such question is solved by clustering approaches. An interest of matrix factorization is that is can be viewed as a dimension reduction procedure that is appropriate for clustering.

We first briefly recall some general concepts about clustering to explain how it is linked to matrix factorization. Spectral clustering is based on the eigen-decomposition of the similarity matrix (for instance between observations). In particular, spectral clustering is related to the standard k-means approach as shown by Dhillon et al. (2004). PCA corresponds to an eigen-decomposition of the empirical covariance matrix, thus if considering the covariance as a similarity, PCA can be seen as a spectral clustering approach. In this



(a)

(b)

(c)

Figure 6.6 – Coordinates of the observations when considering two factors. Comparison of the Poisson-NMF (`pnmf`), the PCA with a pre-transformation of the data by the Anscombe transform (`pca_anscombe`) and our variational-EM algorithm for the ZI-GaP model (`varEM_zigap`). All the data set were generated with $n = 100$, $p = 1000$ and $K^* = 20$. Only the number of groups of observations changed. (a) 2 groups of observations. (b) 3 groups of observations. (c) 4 groups of observations.

regard, Ding & He (2004) investigated the link between PCA and k-means clustering. It can be noted that Yeung & Ruzzo (2001) used PCA to cluster gene expression data. Regarding matrix factorization, Ding et al. (2005) showed an equivalence between the ls-NMF and the k-means/spectral clustering. More recently, Lee et al. (2010) proposed a clustering method based on sparse SVD. Eventually, Wang et al. (2013) introduced a NMF approach for clustering based on a correntropy criterion instead of the ℓ_2 metric or the Bregman divergence for count data.

Based on these different remarks, we assess the interest of our count matrix factorization method for the clustering of zero-inflated data. We process as follows. We use the estimated factor $\widehat{\mathbf{U}}$ to cluster the observations, learned on the same data sets as in the previous section with $n = 100$, $p = 1000$, $K^* = 20$ and different groups of observations. Figure 6.7 shows the heatmap of the matrix $\widehat{\mathbf{U}}$ computed by the different approaches (PCA with Anscombe transform, Poisson-NMF and our ZI-specific variational algorithm) where the rows are reorganized following a hierarchical clustering. If factor $\widehat{\mathbf{U}}$ catches the underlying organization of the data in the observation space, we expect the clusters to correspond to the groups of observations.

We represent the results for different models with different numbers K of factors for each method. Figures 6.7 to 6.9 show the results of the clustering as heatmaps of the matrices $\widehat{\mathbf{U}}$ estimated by the different approaches. The rows are organized according to the clustering. The colors on the left of the heatmap indicate the original group of the corresponding observation. We present the results when considering 2 groups (results with 3 groups are joined in Appendix Section D.3.3)

A first striking point is that the results of the clustering based on our ZI-specific model do not depend on the number of factors set in the model. Indeed, the clustering retrieves the original groups of observations when considering models with different values of K. When K becomes large ($>$15), we see that some columns of $\widehat{\mathbf{U}}$ are set to zero, as shown in Section 6.3.3, however it does not affect the result of the clustering. Therefore, the interest here is that the number of factors can be approximately chosen by checking the norm of the columns of $\widehat{\mathbf{U}}$, it will not affect the clustering of the observations.

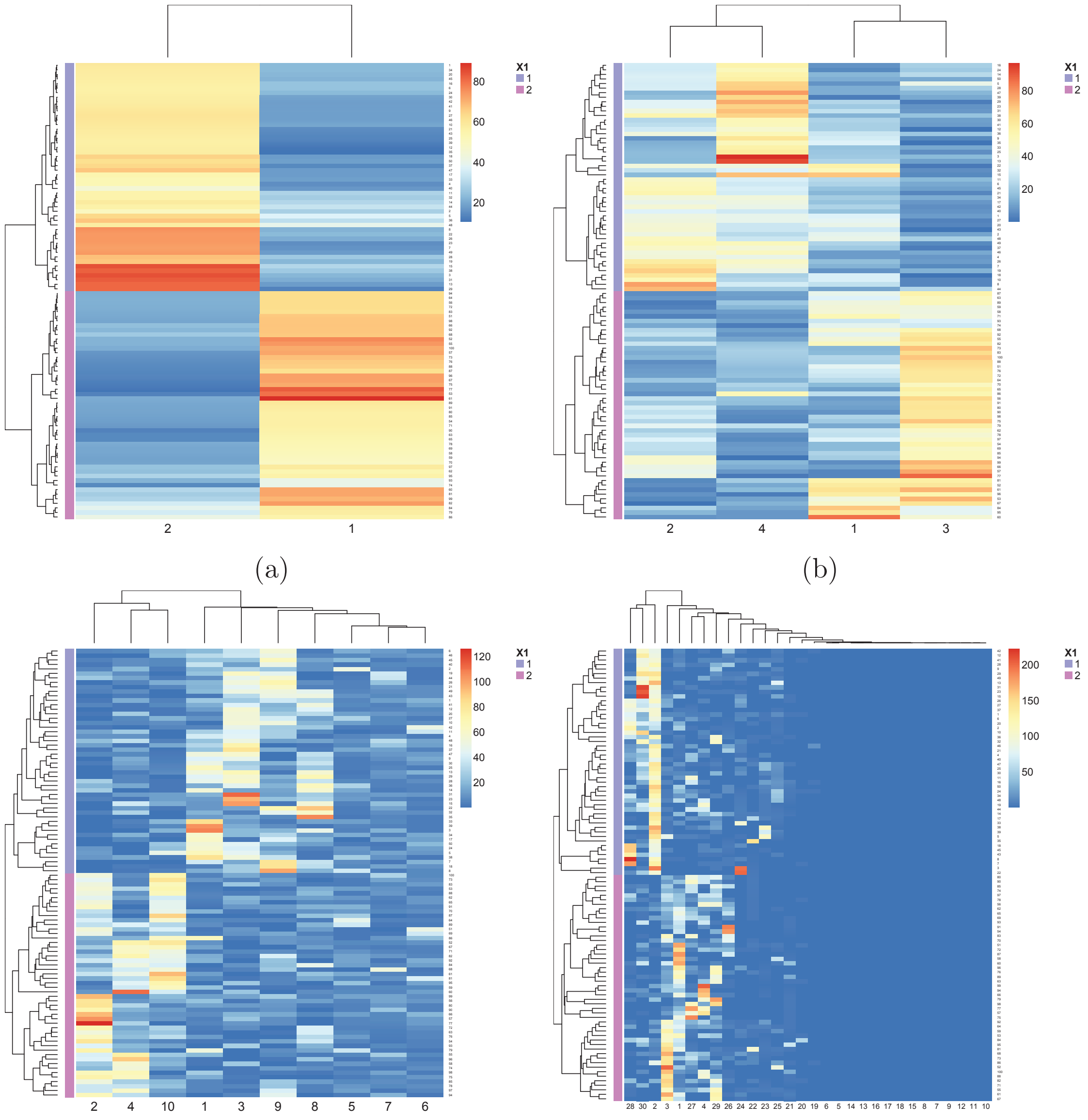
On the contrary, the choice of K is essential in the case of the Poisson-NMF, since the clustering varies a lot between the different fitted models. In the case of data generated with 2 groups, the clusters correspond to the groups when K is small (< 5) but the clusters change totally when considering larger K. A similar behavior is observed in the case of 3 original groups in the data. This is a concern because the choice of K in the case of the Poisson-NMF is not straightforward. For instance, the cumulative Bregman divergence $k \mapsto D\big(\mathbf{X} \,|\, \widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T\big)$ decreases regularly with $k = 1, \ldots, K$ (c.f. Appendix Section D.3.3).

Eventually, PCA is highly affected by the zero-inflation as the clustering based on the estimated $\widehat{\mathbf{U}}$ does not retrieve the original groups. We tried even larger number of factors K for the same results. In this case, determining K based on the cumulative explained variance is problematic as it regularly increases (impossible to use an elbow criterion).

In addition, we ran k-means clustering of the rows of the matrix $\widehat{\mathbf{U}}$ estimated by the different methods and the clusters found by the k-means procedure corresponds to the clusters found by the hierarchical clustering (for all three methods), which confirms the interest of our approach to identify clusters of observations.

Robust inference in variational framework

The zero-inflated data are an example of corrupted data. Such context raises the question about the robustness of our statistical analysis. In the previous section, we saw that our approach is not affected by random drop-out events in the data, in the sense that the matrix factorization procedure is still able to recover the underlying structure even when the data are zero-inflated. This question of robustness in the framework of variational inference is quite recent. In particular, the study of robust variational inference is at the core of very novel works by Giordano et al. (2015a) and Westling & McCormick (2015). In the context of matrix factorization, variational inference have been already used to develop robust version of the PCA based on variational inference in the context of a Laplacian noise (Gao, 2008; Zhao et al., 2015). However, theoretically studying the effective robustness of variational matrix factorization of count data is yet to be done.



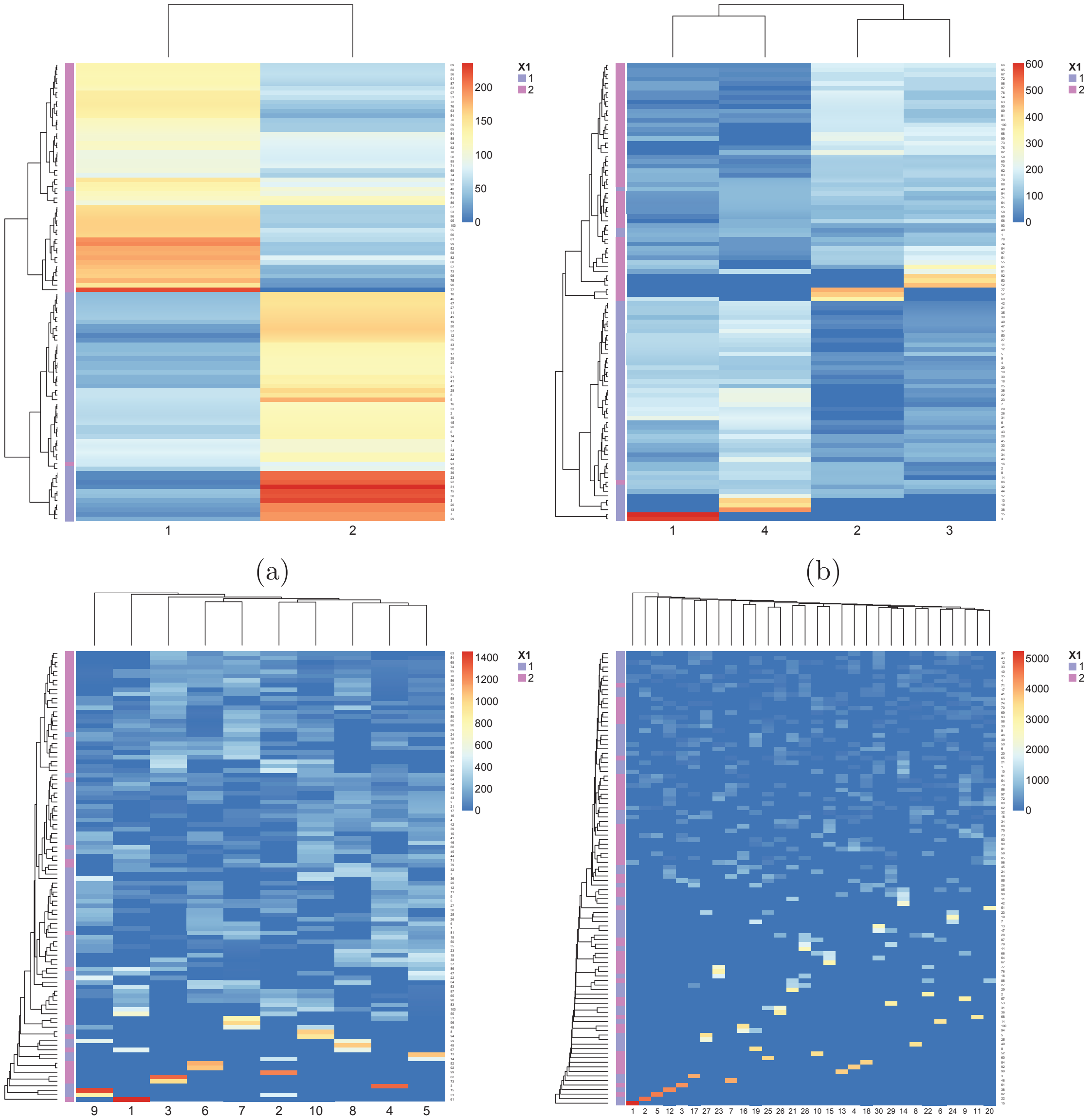
(a)

(b)

(c)

(d)

Figure 6.7 – Clustering of the rows from $\widehat{\mathbf{U}}$ estimated by our variational-EM algorithm for the ZI-GaP factor model. The two colors on the left side correspond to the original two groups of observations. Model with different number of factors: (a) $K = 2$ (b) $K = 4$ (c) $K = 10$ (d) $K = 30$. All models were fitted on the same zero-inflated data set, where $n = 100$, $p = 1000$ and $K^* = 20$.



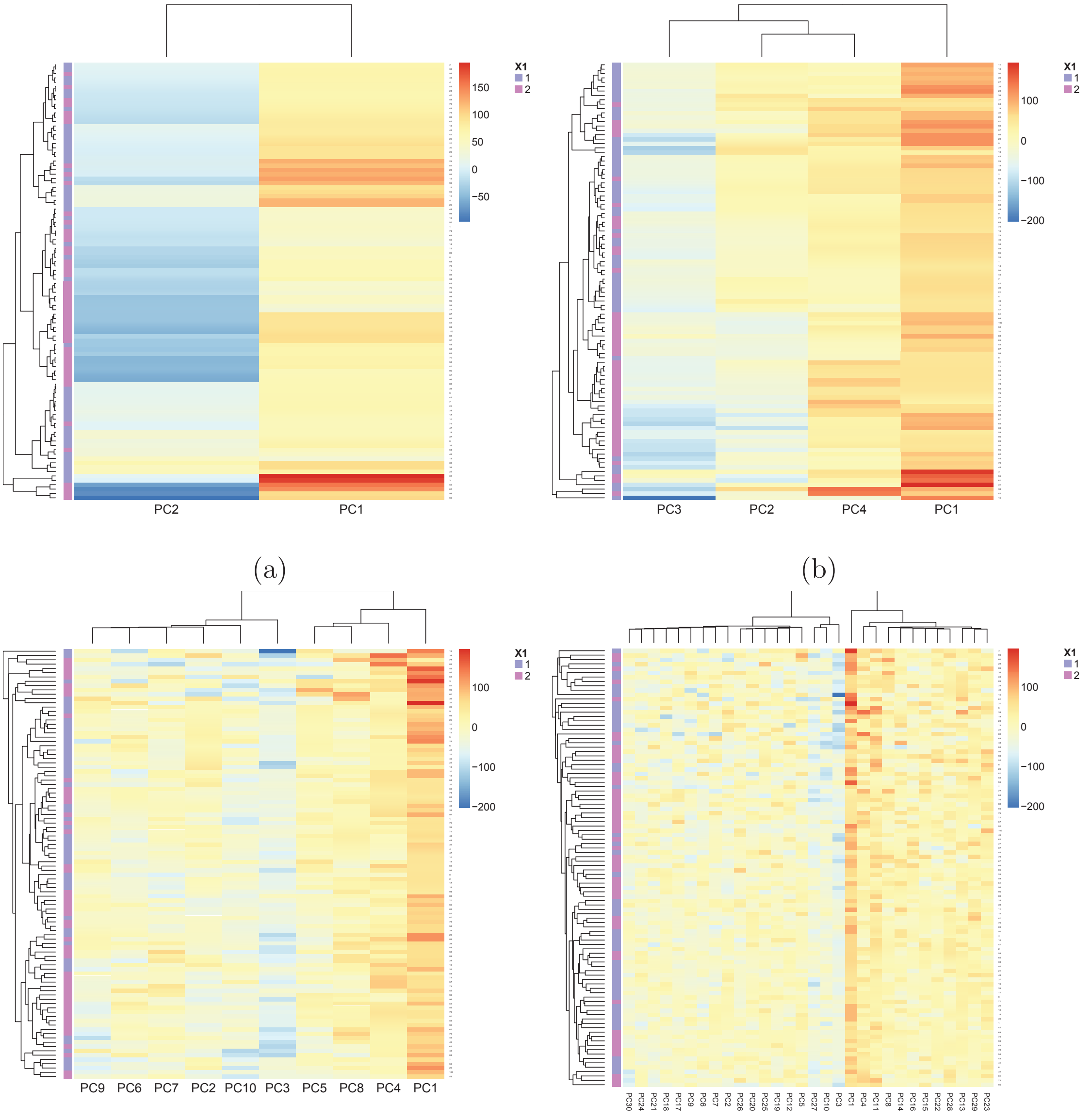
(a)

(b)

(c)

(d)

Figure 6.8 – Clustering of the rows from $\widehat{\mathbf{U}}$ estimates by Poisson-NMF. The two colors on the left side correspond to the original two groups of observations. Model with different number of factors: (a) $K = 2$ (b) $K = 4$ (c) $K = 10$ (d) $K = 30$. All models were fitted on the same zero-inflated data set, where $n = 100$, $p = 1000$ and $K^* = 20$.



(a)

(b)

(c)

(d)

Figure 6.9 – Clustering of the rows from $\widehat{\mathbf{U}}$ estimates by PCA with a pre-transformation of the data by the Anscombe transform. The two colors on the left side correspond to the original two groups of observations. We consider different number of factors: (a) $K = 2$ (b) $K = 4$ (c) $K = 10$ (d) $K = 30$. We analyze the same zero-inflated data set, where $n = 100$, $p = 1000$ and $K^{*} = 20$.

6.3.5 Variable selection with the sparse Gamma-Poisson factor model

The sparse-GaP factor model is supposed to determine by itself which variables are pertinent and which variables should be discarded from the model because of a low contribution to the factors. We propose to assess the accuracy of the variable selection in our inference framework on simulations.

We will compare our approach to other state-of-the art methods for sparse matrix factorization. In particular, we will use the sparse PCA (SPCA), as introduced by Witten et al. (2009) (c.f. Section 4.1.2) that is implemented in the `PMA` `R`-package. To be fair, the data will be transformed thanks to the Anscombe transform at first before running the SPCA. Regarding sparse NMF (SNMF), we will consider the SNMF/R by Kim & Park (2007) (c.f. Section 5.3) that is included in the `NMF` `R`-package. The SPCA procedure requires a tuning parameter that is chosen by 5-fold cross-validation, following the recommendation of the documentation. The cross-validation is based on the reconstruction of the data regarding the Euclidean metric. On the contrary, the SNMF/R procedure depends on a penalty parameter, however at the moment there exists no calibration procedure to choose this parameter. Thus, it would require multiple tries with different values and decide which one is the best value based on an arbitrary criterion. In our simulation, we fix a value of this parameter following the recommendation of the authors.

Our results will show that our method selects almost exactly the relevant variables j when the noise in the model is not too high, even in high dimension. However, our selection procedure seems sensitive to high noise in the data. Surprisingly, the sparse PCA responds better when the noise is high in the data. This point will be discussed in the following.

Model of simulation

The question of simulating data to assess the accuracy of the selection in our context of matrix factorization is not straightforward. We will generate a sparse matrix $\mathbf{V}$. In this context, we will impose that some variables j do not contribute to any component k, i.e. that V_{jk} is null for any k, so that the recorded variable j (i.e. the j^{th} column in $\mathbf{X}$) will be irrelevant in the model. However, in this case, if $V_{jk} = 0$ for any k, then $\sum_k U_{ik} V_{jk}$ is always null for $i = 1, \ldots, n$. Thus, the recorded variable X_{ij} will be deterministic and null for any observation i. There is no interest to generate null columns in the matrix $\mathbf{X}$, since it is unnecessary to use a statistical analysis to determine that a column of zeros will not be informative. This question is not an issue about the formulation of the model, but rather concerns the generation of non informative columns in $\mathbf{X}$ that will correspond to null rows in the matrix $\mathbf{V}$.

To overcome this issue, we use the following generative process. We separate the variables $j = 1, \dots, p$ into a set of active variables, defined as:

$$\mathcal{A} = \{j \ : \ \exists k, V_{jk} \neq 0\} \,,$$

which corresponds to the recorded variables j that contribute to at least one component. And we define the set of non relevant variables, that corresponds to the complementary of $\mathcal{A}$ and verifies:

$$\mathcal{A}^c = \{j \ : \ \forall k, V_{jk} = 0\} \,,$$

i.e. the variables j that never contribute to the latent factors. Once the null entries in $\mathbf{V}$ are defined by $\mathcal{A}^c$, the contributions of the variables j in $\mathcal{A}$ are generated as in the standard GaP model by drawing from a Gamma distribution. The cardinal of the active set $\mathcal{A}$ is denoted p_0, hence the cardinal of $\mathcal{A}^c$ is $p - p_0$.

To avoid for X_{ij} with $j \in \mathcal{A}^c$ to be deterministic and null, the data are generated as $X_{ij} \sim \mathscr{P}(\varepsilon_{ij} + \sum_k U_{ik} V_{jk})$, where $\varepsilon_{ij} > 0$ is a term of noise, so that X_{ij} with $j \in \mathcal{A}^c$ can be interpreted as a noisy variable, that we expect to discard from the model.

To be complete, we set $n = 100$, $p_0 = 50$ and $K^* = 4$. The total number of variables p is set to 100 or 200, so that the ratio p_0/p of relevant variables is either 0.5 or 0.25. We also set different levels of noise that are constant across i and j, i.e. $\varepsilon_{ij} = \varepsilon$ with $\varepsilon = 2, 4, 20$, the higher values correspond to higher noise.

Accuracy of selection

In each different configuration of the data, we generate 50 data sets and run the different methods. For each method, we define the estimated active set of variables that contribute to the model:

$$\widehat{\mathcal{A}} = \{j \ : \ \exists k, \widehat{V}_{jk} \neq 0\} \,.$$

Based on this estimated active set, we compute the accuracy of each selection procedure (c.f. Chapter 2), i.e. the percentage of correctly selected and correctly non selected variables. The results are summarized in Table 6.1.

A first point is that when the noise level is low, the sparse-GaP model inferred by variational-EM is accurate and retrieves the good variables, i.e. it selects the relevant ones and discards the irrelevant ones (accuracy near 100%), even when the number of noisy variables grows (i.e. in both cases $p_0/p = 0.5$ and $p_0/p = 0.25$). However, when the noise is high, the model selects all the variables as the accuracy is exactly p_0/p, i.e. 50% when $p = 100$ and 25% when $p = 200$. This can be explained by the definition of the algorithm, the updates for the spike-and-slab variational parameter p^{s}_{jk} depend on X_{ij} through $\mathbb{E}_q[Z_{ijk}]$. Thus, if the level of the non-pertinent variables becomes high (when the noise is high), the model wrongly selects these variables. Such property means that the sparse-GaP model is more appropriate to handle a sparse signal.

A second comment is that the accuracy of our approach does not decrease when the number of factors K in the model increases. Hence, adding factors in the model is not a problem and will not lead to a phenomenon of over-selection. For instance, as we will discuss below, this sparse PCA has a different behavior.

The results of the sparse NMF approach are not informative. Indeed, in any configuration, it selects all the variables, as the accuracy is always p_0/p. The problem of such approach is clearly the absence of calibration for the tuning parameter, such that we are not sure how to choose it to obtain best results. We did test different values for this parameter. When increasing its value, matrix $\widehat{\mathbf{V}}$ is expected to be more sparse. Nonetheless, in this case, some columns $\widehat{\mathbf{v}}_k$ are set to zero but all variables j contribute to the non-null components. Such an approach would require a calibration procedure to assess its accuracy regarding selection more precisely.

The results of the SPCA are quite surprising too. Indeed, it performs better when the noise is high: accuracy near 100% when $\varepsilon = 20$ (level of the noise) and $K = 2$. On the contrary, when the noise level is low, the accuracy falls near or below 50%. This point is potentially linked to the calibration of the tuning parameter, the cross-validation procedure could choose a parameter that does not enforce a strong sparsity in the model. Another explanation is that, when adding an important noise, the Poisson intensity used to generate the data becomes higher. In this context, when λ is large, $\mathscr{P}(\lambda)$ can be approximated by a Gaussian distribution. The sparse PCA is expected to perform well when the data are near Gaussian. We also observed that the accuracy generally decreases when the number K of factors increases. In practice, this phenomenon seems to correspond to a problem of over-selection, i.e. the variables selected to construct the first components are the more relevant and then the procedure selects less relevant variables to construct the next components.

As previously mentioned, in our method, the tuning parameter π_{thr} that is linked to the degree of sparsity is not tuned. This arbitrary choice is natural because it quantifies the minimal frequency at which a variable j contributes to the factor k. Thus, we chose $\pi_{\mathrm{thr}} = 0.5$ because this is how we would predict the label of a random binary variable depending on its estimated probability (as we do in the logistic regression in Part I). Changing the values of π_{thr} does not seem to improve the selection in the noisy case. To overcome this issue, we are currently considering some modifications about the model (c.f. Chapter 7). Nonetheless, the good behavior in the non-noisy case is encouraging. Indeed, the interest of our method compared to the SPCA for instance is that the tuning parameter can be set without requiring a calibration procedure which is time consuming and a potential source of instability (as we observed in Chapter 2). On this matter, we recall that other hyper-parameters of the model (i.e. parameters of the prior distributions) are estimated within the variational-EM algorithm and do not require to be calibrated.

6.4 Analysis of single cell data

We will now present an application of our approach for the exploration and the representation of single-cell data. These are preliminary results as part of an on-going analysis. We recall some context about the data (c.f. Chapter 4). We have the expression profiles regarding ~ 20000 genes for ~ 1000 single lymphocyte T cells that were sampled on the same human at three time points after a yellow fever vaccine shot, precisely 15, 136 and 908 days after the injection.

The main question in this context concerns the data visualization. Indeed, the data are very noisy because of the particular design of the experiment. As we saw, single-cell data are highly zero-inflated. Moreover, due to the long duration of the experiment (more than two years and a half between the first and the last sequencing experiments), the technologies of sequencing slightly evolved, inducing a potential batch effect in the data, i.e. a technical bias induced by the sequencing machine. To avoid such issue, the data were normalized to remove this batch effect (based on a zero-inflated Negative Binomial GLM with a batch factor). All results presented here were derived on the normalized data.

We will use our specific variational algorithm for zero-inflated data. The objective will be to reduce the dimension for data exploration and visualization, especially to find and illustrate the underlying organization and the diversity between cells. We recall that the single T cells are organized according to their different cell types, i.e. "Effector" or "Memory" (that were predicted, c.f. Section 2.4). Groups of T cells originating from the same ascendent after successive divisions were also identified based on a unique genetic marker that is transmitted during the cellular division.

We will see that there is an important day effect in the data. In particular, the expression of genes in cells depends on the time since the injection. We will also see that the composition of the sample regarding the cell types varies across time. In particular,

Method	p	Noise	$K = 2$	$K = 4$	$K = 5$
var-EM/sparse-GaP	100	2	97.02%	100 %	100 %
		4	50%	90.6%	92.3 %
		20	50%	50%	49.98 %
	200	2	96.23%	100%	100 %
		4	25%	84.05%	89.77 %
		20	25%	25 %	25 %
SPCA	100	2	48.88%	49.92%	50.00%
		4	48.44%	49.52 %	49.96 %
		20	99.82%	78.76 %	62.32 %
	200	2	54.69%	32.19%	24.98 %
		4	66.59%	37.35 %	25.15 %
		20	99.89 %	85.95 %	73.72 %
SNMF/R	100	2	50 %	50 %	50 %
		4	50 %	50 %	50 %
		20	50 %	50 %	50 %
	200	2	25 %	25 %	25 %
		4	25 %	25 %	25 %
		20	25 %	25 %	25 %

Table 6.1 – Accuracy of the variable selection for the different methods: our variational-EM for the sparse-GaP model (var-EM/sparse-GaP), the sparse PCA (SPCA) and the sparse NMF (SNMF/R). The data are generated with $n = 100$, $K^* = 4$ and different values for $p = 100, 200$. In both configurations, the number of relevant variables is $p_0 = 50$. The models are fitted with $K = 2, 4, 5$ factors.

the differences between T cells from different types are more pronounced as time goes on. Eventually, we will discuss the visualization of the organization linked to the clonal structure between cells. The issue here is that the groups of cells are generally very small, since there are numerous clones that contain only a few cells. The solution in this specific case will be to use a supervised approach, since neither our method nor PCA were not able to identify the clonal structure. This point opens a perspective of development for our matrix factorization framework to incorporate some prior information about the individuals (c.f. Chapter 7).

In addition, we precise that we will compare the results obtained by our approach and by the PCA to a dimension reduction method called t-Distributed Stochastic Neighbor Embedding (t-SNE) that was especially developed for visualization and clustering (van der Maaten & Hinton, 2008). Such an approach is based on the derivation of a probability distribution on pairs of individuals in the high-dimensional space, such that pairs of similar individuals have a higher probability than pairs of dissimilar individuals. Then, the algorithm builds a probability distribution on the pairs of individuals in a low dimensional space (for instance in 2-D) such that the low dimensional probability distribution is the closest to the high dimensional probability distribution. Finally, the individuals are positioned in the low dimensional space according to the estimated probability distribution. Hence, t-SNE can be viewed as a clustering method for data visualization.

In the following, the PCA is always applied to the data transformed by the Anscombe transform.

6.4.1 Differences between cells across time

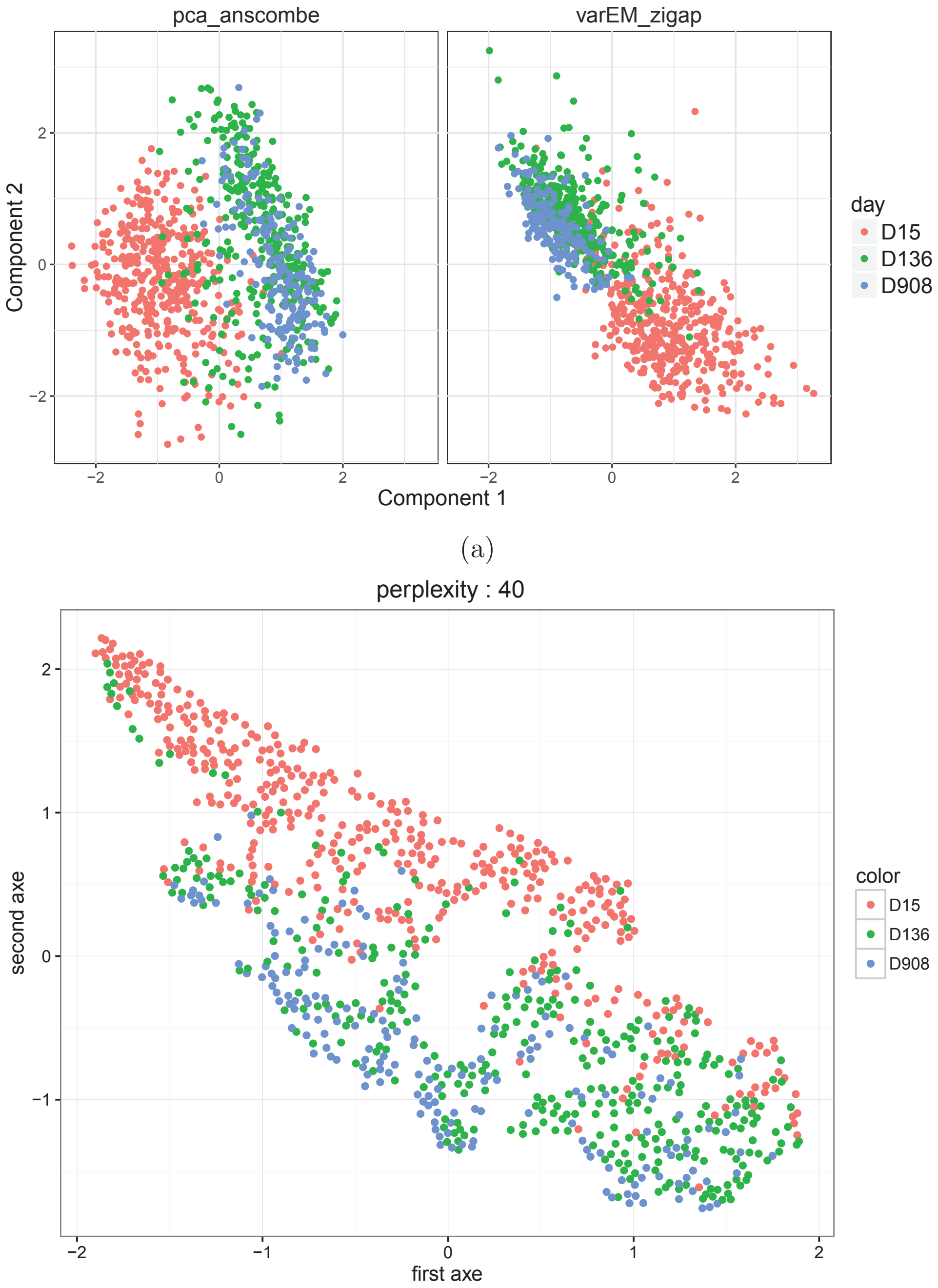
We first tried to investigate the difference between the expression in T cells across time. We considered the ~ 1000 cells from the 3 time points ("D15", "D136" and "D908"). We did not consider the ~ 20000 genes at once. Indeed, since the data are very noisy, a lot of genes are not informative and should not be considered. We restrained this first analysis to ~ 2000 genes that were the union of the 10% most differentially expressed genes between cell-types and the 10% most differentially expressed genes between clones (the differential expression analysis was based on a zero-inflated Negative Binomial model). Hence, both structures are accounted for in the data matrix that we study.

We run our method based on variational-EM for the sparse-GaP model and the PCA with $K = 2$ factors and represent the data according to the columns $\widehat{\mathbf{u}}_1$ and $\widehat{\mathbf{u}}_2$ of $\widehat{\mathbf{U}}$ in the observation space (c.f. Figure 6.10a). We also run t-SNE on the same data subset (c.f. Figure 6.10b). t-SNE depends on a parameter (called perplexity) that we empirically tuned to find the best representation (i.e. with the best visual clustering).

A fist striking point is that the day effect is very important in the data since all methods are able to detect an underlying organization of the data according to the day. Cells at D15 are mostly very different from the rest of the cells. On the contrary, the cells at D136 and D908 are more similar. This observation is however not surprising. Indeed, directly after the vaccine shot, the cells are sampled during the primary immune response when the organism is reacting to the antigen in the vaccine, whereas after 136 and 908 days, the primary immune response is expected to be over, hence the T lymphocytes specific to the antigen in the vaccine are not fighting it anymore, but rather waiting for another infection.

Concerning the comparison of the 3 methods, the signal seems to be very strong between days as the approaches that do not account for the drop-out events are able to retrieve the latent structure despite the zero-inflation. The results of our matrix factorization method is consistent with the results from the PCA and t-SNE that are recognized to be efficient dimension reduction approaches. This point is comforting about the ability of our method to be used in real-life analysis.

An additional comment can be made on the organization of the cells in the latent subspace. Our method seems to identify groups of similar observations that are more compact. Indeed, since the different scatter plots are normalized to the same scale, we can visually identify two compact clusters of cells (one from “D15” and the other from both “D136” and “D908”). The scatter plots from the other two methods are more uniformly spread in the space. This point could somehow be an advantage for clustering as we will see in a moment.



(a)

(b)

Figure 6.10 – Day effect. Data visualization in 2-dimension. Each method is learned on ∼ 1000 cells and ∼ 2000 differentially expressed genes. (a) PCA with a pre-transformation of the data by the Anscombe transform (`pca_anscombe`) and variational-EM for the ZI-GaP model (`varEM_zigap`). (b) t-SNE with a perplexity parameter of 40.

6.4.2 Representation of the different T cell types

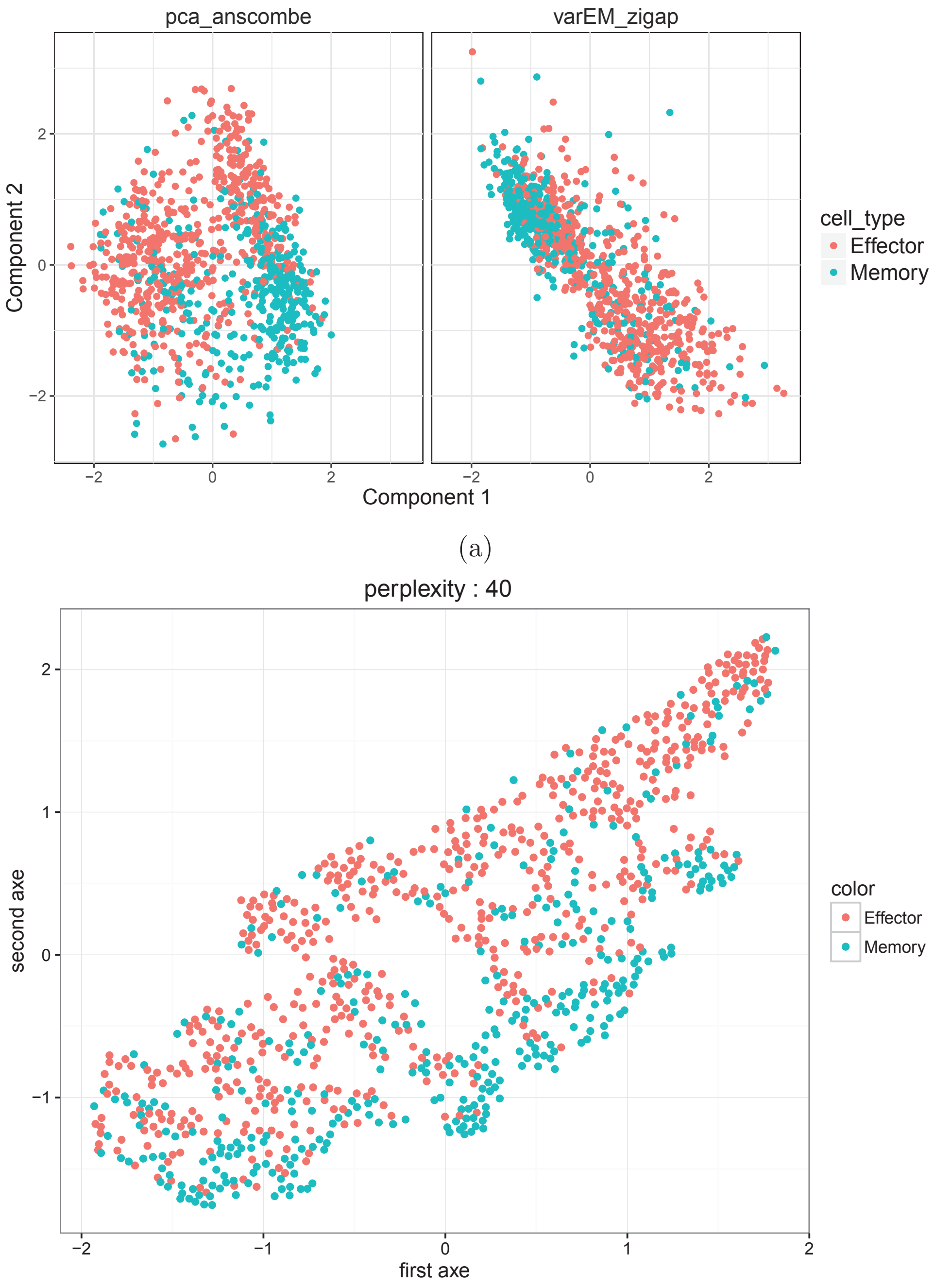
We consider the same subset of the data as in the previous section but we are now interested in the composition of the data regarding the different T cell types. We consider the two main categories of cells "Effector" and "Memory" that have very different functions in the immune system.

Figure 6.11 shows the same cell coordinates in the latent subspace than Figure 6.10 but colored by the cell types predicted in Section 2.4. On this matter, the results of t-SNE are identifiable up to a symmetry or a rotation of the axes, which explains the difference between Figures 6.10b and 6.10b. The day effect is apparently stronger than the cell type effect. Indeed, when comparing Figures 6.10 and 6.11, "Memory" cells at "D15" appear more similar to "Effector" cells at "D15" than to "Memory" cells at the other time points, according to the underlying structure reconstructed by all methods. This is specifically identifiable on the graph based on our variational-EM approach. On the contrary, our approach identifies "Memory" cells from "D136" and "D908" that are visually grouped in a compact cluster.

It can also be noted that the T cell populations are not balanced between the different cell types across time. At "D15", a short time after the injection, there is a majority of "Effector" cells that are recruited during the primary immune response. Then, as time goes on, the proportion of "Memory" cells increases until "D908" when they are a majority, these cells are the one conserved across time so that the immune system is able to respond to an infection by a similar antigen in the future.

In order to assess more precisely the interest of our methods regarding the question of the reconstruction of the groups of cells, we considered the clustering of the cells according to the rows of the matrix $\widehat{\mathbf{U}}$ learned by the PCA and by our variational-EM algorithm. When considering cells from all days, the clustering (hierarchical or k-means) retrieves clusters of cells that correspond either to "D15" or to a mix of "D136" and "D908" (which is consistent with the 2-dimensional representation in the previous section). Therefore, we consider each day separately, to study the cell type organization in each sub-sample of cells (that were sampled in the same conditions).

We focus on "D908" and consider $\sim$ 200 cells and $\sim$ 3000 differentially expressed genes (for the cell type effect only). Figure 6.12 shows the results of the clustering as a heatmap of the matrix $\widehat{\mathbf{U}}$ whose rows are reordered according to a hierarchical clustering. We choose to consider 3 components in our model based on the normed of the factors (as developed in Section 6.3.3). Regarding the PCA, it was more difficult to choose a number of components as the cumulative proportion of explained variance increases regularly with the number of components K. We did try various different numbers of components but



(a)

(b)

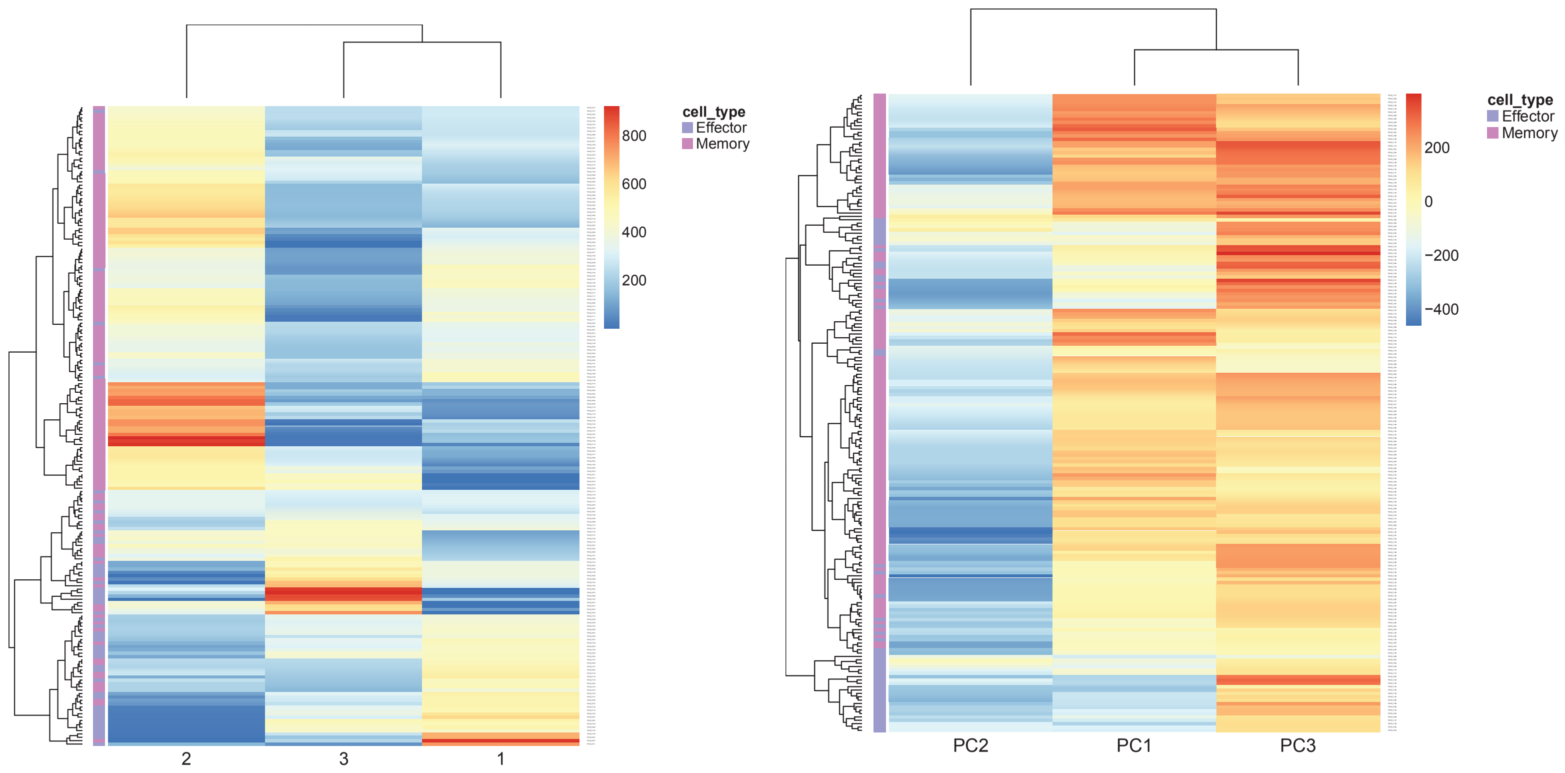
Figure 6.11 – Cell type effect. Data visualization in 2-dimension. Each method is learned on ∼ 1000 cells and ∼ 2000 differentially expressed genes. (a) PCA with a pre-transformation of the data by the Anscombe transform (`pca_anscombe`) and variational-EM for the ZI-GaP model (`varEM_zigap`). (b) t-SNE with a perplexity parameter of 40.

we cannot obtain satisfying results regarding the clustering. Thus, we show an example with 3 components as well.

The main point here is that the clustering based on the matrix $\widehat{\mathbf{U}}$ estimated by our method allows to find two large clusters of observations that are clearly identified in the dendrogram in Figure 6.12a concerning the clustering of the observations. A first cluster corresponds to the “Memory” cells and the second to majority of “Effector”.

On the one hand, this results highlight the efficiency of our method for dimension reduction, as it constructs a space of dimension 3 that summarizes the information within ~ 3000 genes. On the other hand, this result potentially shows that the prediction of the cell types is not perfect and that some cells are potentially misclassified. We will have to investigate this point more precisely.

Concerning the other time points, i.e. “D15” and ‘908”, both methods were not able to retrieve the groups of cells. This result is somehow expected, since the T cells appears to be more and more different across time regarding their expression profiles. Thus, it seems complicated to expect to identify the group structure in the early day sample. This point will be completed in the next section.



(a)

(b)

Figure 6.12 – Clustering of the rows from $\widehat{\mathbf{U}}$. Each method is learned on $\sim$ 200 cells at “D908” and $\sim$ 3000 differentially expressed genes. The two colors on the left side correspond to the original two groups of observations. The dendrogram on the left corresponds to the hierarchical clustering of the individuals. (a) Variational-EM for the ZI-GaP model. (b) PCA with a pre-transformation of the data by the Anscombe transform.

6.4.3 Identification of the clones

Eventually, we did focus on the clonal structure. Working with ~ 1000 differentially expressed genes between clones, we tried to use our approach and the PCA to reconstruct the underlying organization linked to the clones of cells. Unfortunately, the latent subspace reconstructed by all methods (variational-EM for the ZI-GaP model, PCA and t-SNE) does not highlight a clear clustering of the cells according to the clones.

In order to find the latent directions that explain the clonal organization, we had to use a supervised approach. Indeed, since there are numerous clones (i.e. group of cells originating from the same ascendents) with only a few members, we assume that the signal structuring the clonal groups is not sufficiently strong in the data so that an unsupervised approach may catch it.

As stated, we were able to find the latent structure associated to the clonal organization thanks to a supervised approach, that accounts for the classes of observations but that is not suitable for prediction. The Between-Class Analysis (BCA) (Baty et al., 2006) that we considered corresponds to a PCA decomposition of the matrix of clone centroids, i.e. the matrix of averaged expression for each gene by clone. After constructing the latent subspace associated to the clone centroids, the observations, i.e. the expression profiles of the single cells, are projected in this subspace. The coordinates of the cells can be visualized as in the standard PCA. Averaging the expression over the clones reduces the effect of the zero-inflation in the data which explains why PCA-based methods can be appropriate in this specific case.

Figure 6.13 shows the results of the BCA at "D908" when considering ~ 75 cells and ~ 1000 differentially expressed genes between clones. In practice, the analysis is restricted to the clones with at least 3 cells, which corresponds to 14 clones. The ellipses correspond to the different clones, they are colored according to the majority cell types in the clones. The points corresponding to the cells are colored according to the cell types. We also represent the coordinate of the genes that contribute the most to the first two components. The interest here is to identify the genes that are associated to the inter-clone variability.

These results clearly highlight the interest of supervised approaches in the case of complex organization by groups (numerous groups with few individuals). As discussed in Chapter 7, we project to adapt our GaP factor model in order to consider an a priori information on the data.

In addition, we precise that the differences between clones appear more and more clear across time. The figures at "D15" and "D136" are not joined but we comment the results.

Indeed, in the BCA at “D15”, the clones are all similar since there is no distinction between the ellipse identifying the clones. At “D136”, the BCA find that the clones are more distinct but remains visually concentrated in the 2-dimensional space, whereas the differences clearly appears at “D908” (c.f. Figure 6.13).

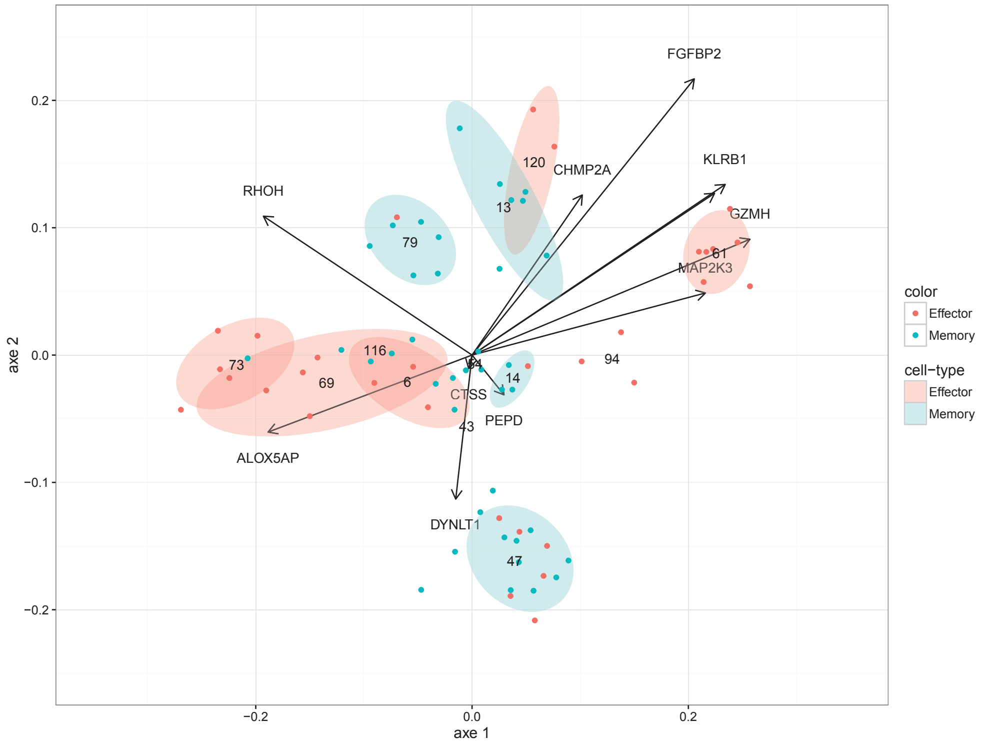


Figure 6.13 – BCA based on the clonal organization with ~ 75 cells and ~ 1000 differentially expressed genes between clones. The points correspond to the coordinates of the cells in the subspace learned with the centroids of the clones. The ellipses gather the cells from the same clones and are colored according to the majority cell types in the clones. The points (i.e. the cells) are colored according to their cell type. The arrows represent the coordinates of the genes that contribute the most to the first two components.

Chapter 7

Conclusion and perspectives about count matrix factorization

In the second part of this manuscript, we focused on dimension reduction approaches in an unsupervised context. In particular, we presented the interest of matrix factorization for data exploration and data visualization. The question of representing the data is always central in any analysis, in particular to understand the underlying organization between observations and between the recorded variables. This point is particularly crucial in the case of high-dimensional data since it can be very complicated to represent the entire complexity of a high-dimensional space in 2 or 3-dimensions.

Different methods for matrix factorization exist, some are based on a geometric criterion, other an a model-based formulation. Their purpose is essentially to approximate a data matrix $\mathbf{X}$ by the product $\mathbf{U}\mathbf{V}^T$ where $\mathbf{U}$ and $\mathbf{V}$ lie in a low dimensional subspace. The Principal Component Analysis (PCA) is a geometrical approach that is widely used for unsupervised dimension reduction and data visualization. We explained how the PCA is linked to the Euclidean geometry and the Singular Value Decomposition (SVD) that finds $\mathbf{U}$ and $\mathbf{V}$ by a least squares approximation $\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2$. Although it is not defined with a model, the PCA is highly appropriate to analyze Gaussian data, because of the link between the Euclidean geometry and the Gaussian model.

When dealing with other types of data involving other distributions in the exponential family, model-based approaches are another solution to develop a method for matrix factorization. This framework is more rich and flexible to comply with the specific nature of the data and model the complex dependencies within the data. We saw that recent genomic data associated to Next-Generation Sequencing (NGS) are typically non-Gaussian. In particular, gene expression profiles are over-dispersed count data. Thus, we presented a Gamma-Poisson (GaP) factor model that accounts for over-dispersion. The hierarchical formulation does not require for the variables X_{ij} to be marginally independent. In practice, the use of prior onto the latent factors U_{ik} and V_{jk} allows to incorporate the structure of dependencies within the model.

The question of the estimation of the parameters is central in model-based approaches. In our case, standard methods such as the Maximum Likelihood Estimation (MLE) or the Expectation-Maximization (EM) algorithm are intractable due to the complex hierarchical formulation of the model. Instead of point estimation, the Bayesian inference aims at computing the posterior distribution of the latent variables in the model, here the latent factors $\mathbf{U}$ and $\mathbf{V}$. To do so, Markov Chain Monte Carlo (MCMC) methods are able to approximate the exact posterior but have a computational cost that cannot be ignored, especially when developing a concurrent to the PCA that is instantaneous, even in the case of high dimensional data. Therefore, we turned to the framework of variational inference that allows to approximate the posterior of the model. This approximation of the posterior regarding the Kullback-Leibler (KL) divergence is shown to be equivalent to minimizing a lower bound on the marginal likelihood, namely the Evidence Lower Bound (ELBO). To be tractable, the variational distribution that approximate the posterior is assumed to comply with a mean-field assumption (regarding independence and the type of distribution in the exponential family).

We derived a variational-EM algorithm that approximates the posterior and estimates the hyper-parameters of the Gamma priors in the GaP factor model. After focusing on the convergence of our procedure, we showed on simulations the interest of our model for dimension reduction in the context of count data, in particular compared to other count specific approaches.

Single-cell sequencing is a recent technology that allows to capture the genetic material of a single cell, for instance to quantify the gene expression in each individual cell of a population. Such data are characterized by drop-out events, which correspond to random missing observations quantified as null values in $\mathbf{X}$. The statistical analysis of such zero-inflated (ZI) data represents a challenge since the huge proportions of corrupted zeros may have an effect on the dimension reduction process. The interest of our GaP factor model for matrix factorization is that it can be refined to account for the zero-inflation in the data. We explicitly derived a variational inference algorithm for this model and show the interest of such an approach for data visualization and clustering on simulated data. In addition, we compare our method to existing procedures for matrix factorization that do not account for zero-inflation. As expected, our zero-inflated Gamma-Poisson (ZI-GaP) model is more appropriate to reconstruct the true signal in data corrupted by drop-out events.

Following the paradigm that we also developed in Part I, we developed a variable selection procedure for matrix factorization based on sparsity-inducing, also known as spike-and-slab, priors in the GaP factor model. We assessed the performance regarding selection accuracy of this method on simulated data. It did show particular abilities to distinguish pertinent variables from irrelevant ones, even in high dimension, however it appeared to be very sensitive to the noise in the data, as it reacted poorly in the case of highly noisy data. This point calls for some improvements as discussed below. Nonetheless, the interest of such an approach compared so standard procedure as sparse PCA (SPCA) is that the tuning parameter (π_{thr}) is interpretable and can be set without using a time-consuming calibration procedure.

Eventually, we proposed an application of our method to an experimental data set. As part of an on-going study, we worked on the expression profiles of single lymphocytes T cells that were sampled at different time points after a vaccine shot. The global interest of the study is to characterize an immune response at the molecular level. Our part in this work was to develop a framework for data exploration and visualization that allows to understand the latent organization in the data. Because of the complexity of the data (single-cells sampled at very different time), it appears very difficult to find latent directions that clearly explain the organization of the cells (between cell types, etc.). Our method clearly identified the evolution of the cells across time (effect of the sampling date in the data). We could also cluster the cells and partially retrieve the cell types that were predicted previously (c.f. Chapter 2). Compared to other standard procedures as PCA or t-SNE, our approach seems to identify more compact clusters of similar cells. However, the limits of unsupervised methods appeared when considering an organization based on numerous small groups of observations. In this case, the only possibility was to use a supervised procedure.

Our contribution in this part can be summarized as follows. We developed a zero-inflated version and a sparse version of the GaP factor model for matrix factorization. We derived an inference algorithm based on the variational-EM framework, that is computationally efficient to infer the model. In our simulations and experimental analysis, we highlighted the interest of our approach for dimension reduction but also some of the limits of our model. Therefore, we will investigate different options in order to improve the performance of our approach. In the same time, we will continue to work on the single-cell data analysis. The experimental results were not as good as the results on the simulations. These specific data are very particular due to the complexity of the experiment. Thus, we plan to apply our approach to other single-cell data to verify how it may behave in other contexts. In addition, the collaboration regarding the T cell data is not over yet. We just presented some preliminary results here. In particular, we did not apply our variable selection procedure to the single-cell data set. We first have to completely redefine a sparse and zero-inflated GaP model, since we encountered an identifiability issue between the drop-out parameter π_j^{D} and the spike-and-slab parameter π_j^{S}. Indeed, in some cases, it cannot be possible to differentiate between a drop-out event or a variable that should be discarded from the model. We will have to find a trick to overcome this issue. It can be noted that such combination of sparsity and zero-inflation has never been proposed in the literature. In addition, we will investigate the different following points.

Introducing structure in the model

A first refinement of our GaP model would be to discard the assumptions about the independence between the latent factors. For instance, we could consider a model where the observations are structured thanks to a prior information. This would require to consider a multivariate distribution on the factors U_{ik}. Some generalizations of the Gamma distribution to the multivariate case exist, e.g. the Wishart distribution. The derivation of the inference algorithm will not be as straightforward as in our GaP model. Thus, we would have to consider a variational framework that relaxes the mean-field assumption, especially regarding the independence in the variational distribution. On this matter, some approaches that consider dependency in the variational framework have been recently proposed, see Hoffman & Blei (2014) and Giordano et al. (2015b), and will possibly be adapted to our framework.

Variable selection

Regarding variable section in the sparse Gamma-Poisson (sparse-GaP) factor model, in the very short-term, we plan to run simulations at larger scale in order to assess the sensibility of our variational algorithm to the signal-to-noise ratio (SNR). In particular, the sense of the SNR is not clear in the context of Poisson distribution compared to the Gaussian case. Based on the results of our simulations, we should also integrate a quantification of the noise in the sparse model, to avoid phenomenon of over-selection when the noise is high, as observed on our simulations.

Meanwhile, we will also focus on the on-going analysis of the single-cell data set. Questions remain, especially regarding the selection of genes that are linked to the underlying organization of the cells regarding the clonal and the phenotypic structures. In particular, our interest here would be to assess the contribution of the differentially expressed genes to the latent structures. In the same time, we have to investigate the differences between the results of the prediction of the cell type by the sparse PLS (c.f. Section 2.4) and the results of the clustering based on the matrix factorization of the single-cell expression profiles (c.f. Section 6.4). Knowing the genes that are relevant to explain the underlying geometry of the data would be an advantage to refine the identification of the cells regarding their phenotype.

On a longer-term, we project to work on a model with a a structuring of the spike-and-slab formulation between the different factors V_{jk}. The idea here would be to generalize the notion of structured penalty, following the framework of the group-Lasso (Yuan & Lin, 2006) or the fused Lasso (Tibshirani et al., 2005; Rinaldo, 2009). In the context of regression, these two approaches are respectively based on a penalization of groups of coefficients (to enforce the selection or non-selection of groups of variables together) and on a penalization of the difference between coefficients on groups of variables (so that related variables have the same effect on the response). To do so, we could consider joined spike-and-slab indicators for different factors V_{jk}. Such approach would require to leave the mean-field family and the independence between the variational distribution regarding the factors V_{jk} (as previously mentioned).

Initialization and local optima

The optimization framework in variational inference leads to a local optimum of the ELBO. Thus, the estimated values depend on the initialization of the iterative scheme. To overcome this potential issue, we proposed to use multi-random initialization that are iterated for hundred iterations, the run associated to the best value of the objective function is then iterated until convergence. We saw that such procedure is relatively stable as there is not much variability when measuring the reconstruction of the data by the model (based on the Bregman divergence).

Another option would be to consider an optimization based on a simulated annealing procedure. The reader may refer to Kirkpatrick et al. (1983) and Brooks & Morgan (1995) for a detailed introduction. The objective of this framework is to find an approximation of the global optimum of an objective function. The principle consists in adding a stochastic perturbation in the trajectory of the parameters every N^{th} iterations (N has to be chosen), so that the iterative procedure will not get stuck near a bad local optimum (i.e. that is far from being the global optimum), and may possibly jump to another local optimum, until finding the best local optimum. The amplitude of the perturbation, also called the temperature, decreases with the growing number of iterations, so that the algorithm explores a wide range of candidate values at the beginning and then narrows its research around the potential global optimum.

Such an approach would slow down the convergence, however if the temperature is correctly set, the interest is that it finds a good approximation of the global optimum. In the context of variational inference, simulated annealing have been recently considered, see Obermeyer et al. (2014) and Gultekin et al. (2015), and can be integrated in our optimization framework.

Model selection

The question about the choice of the number of components in the GaP factor model admits different answers. At the moment, the procedure to choose K is empirical and based on the construction of the matrix $\widehat{\mathbf{U}}$. The model appears to cut out the unnecessary factors by setting the norm of the associated column $\widehat{\mathbf{u}}_k$ in $\widehat{\mathbf{U}}$ to zero. However, the question of comparing different models with different numbers of factors remains unclear. Indeed, at the moment, we fit the model with a large number of factors. Depending on the number of factors with a null norm, we can refit the model with a more appropriate number of components.

We plan to develop an automatic procedure to choose K based on a model selection criterion. The Bayesian Information Criterion (BIC) introduced by Schwarz (1978) is a criterion based on the penalized marginal log-likelihood, the penalty depends on the number of parameters in the model so that it counter-balances the phenomenon of over-fit. The BIC is derived thanks to a Laplace approximation (Tierney & Kadane, 1986) when integrating the marginal likelihood over the parameters. The BIC is finally computed with the MLE of the parameters of the model. Thus, it could be used to choose K in the case of the Non-negative Matrix Factorization (NMF) as the factors $\mathbf{U}$ and $\mathbf{V}$ are parameters of the model, hence the marginal likelihood is explicit. On the contrary, in our hierarchical GaP model, the marginal and the MLE are intractable.

To overcome such an issue, Biernacki et al. (2000) derived the Integrated Completed Likelihood (ICL) criterion[1] that is based on a Laplace approximation of the integral of the joint likelihood over the hyper-parameters of the model. The interest of such an approach is to derive a BIC-like penalty that accounts for the hierarchy in the model. Using an ICL formulation for model selection would require to investigate the asymptotic validity of the Laplace approximation in our model. In the case of the standard GaP factor model, the main interest of such a method is that deriving the asymptotic behavior regarding the number of observations n when the number of variables p is fixed would lead to an heuristic in the case of p growing and n fixed. Indeed, thanks to the definition of the model, the role of U_{ik} and V_{jk} are symmetrical in the joint likelihood. However, such a property would not be expected in the ZI-GaP or sparse-GaP factor models, since the role of U_{ik} and V_{jk} are not symmetrical anymore.

[1] See also Biernacki et al. (2010).

Multiple discriminant analysis

Eventually, we project to explore the possibility to derive a supervised approach, similar to the Between-Class Analysis (BCA) introduced in the previous chapter or to the Multiple Discriminant Analysis (MDA). The purpose of such approaches is to find latent directions that explain the partitioning of the observations in different classes. In this supervised framework, the objective remains data exploration (and visualization) and not prediction. The BCA is based on an eigen-decomposition of the matrix of the class centroids, and especially thanks to a PCA applied to matrix of averaged observations by groups. The MDA constructs the latent components that maximize the between-class variability and minimize the intra-class variability.

Such approaches are still based on the Euclidean geometry. Following our reasoning from the previous chapter, we aim at generalizing such framework to count data, by considering an appropriate underlying geometry. To do so, we could consider a GaP factor model where the prior hyper-parameters depend on the classes of observations. We could even imagine using structured multivariate priors that account for the difference between classes. The objective here is clearly to infer models that account for the organization of the data in groups. For example, we saw in our application that considering an a priori on the data organization is necessary to process an efficient dimension reduction and retrieve small groups of observations (the clones in our application on single-cell data).

Bibliography

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*(4), 433–459.

Acharya, A., Ghosh, J., & Zhou, M. (2015). Nonparametric Bayesian Factor Analysis for Dynamic Count Matrices. In *AISTATS*.

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *International Conference on Database Theory*, (pp. 420–434). Springer.

Albert, A., & Anderson, J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, *71*(1), 1–10.

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. (2007). *Molecular Biology of the Cell, 5th Edition*. New York: Garland Science, 5th edition ed.

Aldrich, J. (1997). RA Fisher and the making of maximum likelihood 1912-1922. *Statistical Science*, *12*(3), 162–176.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, *96*(12), 6745–6750.

Anders, S., & Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, *11*(10), R106.

Antoniadis, A., Lambert-Lacroix, S., & Leblanc, F. (2003). Effective dimension reduction methods for tumor classification using gene expression data. *Bioinformatics*, *19*(5), 563–570.

Archambeau, C., & Bach, F. R. (2009). Sparse probabilistic projections. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.) *Advances in Neural Information Processing Systems 21*, (pp. 73–80). Curran Associates, Inc.

Babacan, S. D., Nakajima, S., & Do, M. N. (2014). Bayesian group-sparse modeling and variational inference. *Signal Processing, IEEE Transactions on*, *62*(11), 2906–2921.

Bach, F., Jenatton, R., Mairal, J., & Obozinski, G. (2012). Optimization with Sparsity-Inducing Penalties. *Found. Trends Mach. Learn.*, *4*(1), 1–106.

Bach, F., Mairal, J., & Ponce, J. (2008). Convex Sparse Matrix Factorizations. *arXiv:0812.1869 [cs]*.

Bailey, S. (2012). Principal Component Analysis with Noisy and/or Missing Data. *Publications of the Astronomical Society of the Pacific*, *124*(919), 1015–1023.

Banerjee, A., Merugu, S., Dhillon, I. S., & Ghosh, J. (2005). Clustering with Bregman Divergences. *Journal of Machine Learning Research*, *6*(Oct), 1705–1749.

Barker, M., & Rayens, W. (2003). Partial least squares for discrimination. *Journal of chemometrics*, *17*(3), 166–173.

Bartholomew, D. J. (2004). Latent Variable Modelling: Old and New Approaches. Tech report.

Baty, F., Facompré, M., Wiegand, J., Schwager, J., & Brutsche, M. H. (2006). Analysis with respect to instrumental variables for the exploration of microarray data structures. *BMC Bioinformatics*, *7*, 422.

Beal, M. J., & Ghahramani, Z. (2003). The variational Bayesian EM algorithm for incomplete data: With application to scoring graphical model structures. *Bayesian statistics*, *7*, 453–464.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, (pp. 289–300).

Bhattacharya, A., & Dunson, D. B. (2011). Sparse Bayesian infinite factor models. *Biometrika*, *98*(2), 291–306.

Bickel, P. J., & Levina, E. (2004). Some theory for Fisher's linear discriminant function,'naive Bayes', and some alternatives when there are many more variables than observations. *Bernoulli*, (pp. 989–1010).

Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *22*(7), 719–725.

Biernacki, C., Celeux, G., & Govaert, G. (2010). Exact and Monte Carlo calculations of integrated likelihoods for the latent class model. *Journal of Statistical Planning and Inference*, *140*(11), 2991–3002.

Bishop, C. M., & Tipping, M. E. (1998). A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(3), 281–293.

Blazère, M., Gamboa, F., & Loubes, J.-M. (2014). PLS: A new statistical insight through the prism of orthogonal polynomials. *arXiv:1405.5900 [math, stat]*.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2016). Variational Inference: A Review for Statisticians. *arXiv preprint arXiv:1601.00670*.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, *3*(Jan), 993–1022.

Bonafede, E., Picard, F., Robin, S., & Viroli, C. (2015). Modeling overdispersion heterogeneity in differential expression analysis using mixtures. *Biometrics*, (pp. n/a–n/a).

Boulesteix, A.-L. (2004). PLS dimension reduction for classification with microarray data. *Statistical applications in genetics and molecular biology*, *3*(1), 1–30.

Boulesteix, A.-L., & Strimmer, K. (2007). Partial least squares: A versatile tool for the analysis of high-dimensional genomic data. *Briefings in bioinformatics*, *8*(1), 32–44.

Brooks, J., Dulá, J., & Boone, E. (2013). A Pure L1-norm Principal Component Analysis. *Computational statistics & data analysis*, *61*, 83–98.

Brooks, S. P., & Morgan, B. J. (1995). Optimization using simulated annealing. *The Statistician*, (pp. 241–257).

Brown, P. O., & Botstein, D. (1999). Exploring the new world of the genome with DNA microarrays. *Nature genetics*, *21*, 33–37.

Brunet, J.-P., Tamayo, P., Golub, T. R., & Mesirov, J. P. (2004). Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, *101*(12), 4164–4169.

Buganim, Y., Faddah, D. A., Cheng, A. W., Itskovich, E., Markoulaki, S., Ganz, K., Klemm, S. L., van Oudenaarden, A., & Jaenisch, R. (2012). Single-Cell Expression Analyses during Cellular Reprogramming Reveal an Early Stochastic and a Late Hierarchic Phase. *Cell*, *150*(6), 1209–1222.

Bullard, J. H., Purdom, E., Hansen, K. D., & Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC bioinformatics*, *11*(1), 94.

Buntine, W., & Jakulin, A. (2006). Discrete component analysis. In *Subspace, Latent Structure and Feature Selection*, (pp. 1–33). Springer.

Candès, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis? *Journal of the ACM (JACM)*, *58*(3), 11.

Canny, J. (2004). GaP: A factor model for discrete data. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 122–129). ACM.

Carbonetto, P., Stephens, M., & others (2012). Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian analysis*, *7*(1), 73–108.

Caron, F., & Doucet, A. (2008). Sparse Bayesian nonparametric regression. In *Proceedings of the 25th International Conference on Machine Learning*, (pp. 88–95). ACM.

Cemgil, A. T. (2009). Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, *2009*.

Chen, P., Chen, Y., & Rao, M. (2008). Metrics defined by Bregman Divergences. *Communications in Mathematical Sciences*, *6*(4), 915–926.

Chong, I.-G., & Jun, C.-H. (2005). Performance of some variable selection methods when multicollinearity is present. *Chemometrics and Intelligent Laboratory Systems*, *78*(1), 103–112.

Christiansen, C. L., & Morris, C. N. (1997). Hierarchical Poisson regression modeling. *Journal of the American Statistical Association*, *92*(438), 618–632.

Chun, H., & Keleş, S. (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(1), 3–25.

Chung, D., & Keleş, S. (2010). Sparse Partial Least Squares Classification for High Dimensional Data. *Statistical Applications in Genetics and Molecular Biology*, *9*(1).

Cleynen, A., Dudoit, S., & Robin, S. (2013). Comparing Segmentation Methods for Genome Annotation Based on RNA-Seq Data. *Journal of Agricultural, Biological, and Environmental Statistics*, *19*(1), 101–118.

Coelho, C. A., & Arnold, B. C. (2014). On the exact and near-exact distributions of the product of generalized Gamma random variables and the generalized variance. *Communications in Statistics-Theory and Methods*, *43*(10-12), 2007–2033.

Collins, M., Dasgupta, S., & Schapire, R. E. (2001). A generalization of principal components analysis to the exponential family. In *Advances in Neural Information Processing Systems*, (pp. 617–624).

Comon, P. (1994). Independent component analysis, a new concept? *Signal processing*, *36*(3), 287–314.

Dai, J. J., Lieu, L., & Rocke, D. (2006). Dimension reduction for classification with gene expression microarray data. *Statistical applications in genetics and molecular biology*, *5*(1).

Dalrymple, M. L., Hudson, I. L., & Ford, R. P. K. (2003). Finite mixture, zero-inflated Poisson and hurdle models with application to SIDS. *Computational Statistics & Data Analysis*, *41*(3), 491–504.

Datta, J., & Dunson, D. B. (2015). Priors for High-Dimensional Sparse Poisson Means. *arXiv preprint arXiv:1510.04320*.

De La Torre, F., & Black, M. J. (2003). A framework for robust subspace learning. *International Journal of Computer Vision*, *54*(1-3), 117–142.

De Leuuw, J. (1986). The Role of Models in Principal Component and Factor Analysis. In *Multidimensional Data Analysis*. Leiden, The Netherlands: DSWO Press.

De Mol, C., De Vito, E., & Rosasco, L. (2009). Elastic-net regularization in learning theory. *Journal of Complexity*, *25*(2), 201–230.

Delyon, B., Lavielle, M., & Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *Annals of statistics*, (pp. 94–128).

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, (pp. 1–38).

Dhillon, I. S., Guan, Y., & Kulis, B. (2004). Kernel k-means: Spectral clustering and normalized cuts. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 551–556). ACM.

Diaconis, P., Ylvisaker, D., & others (1979). Conjugate priors for exponential families. *The Annals of statistics*, *7*(2), 269–281.

Dikmen, O., & Févotte, C. (2012). Maximum marginal likelihood estimation for nonnegative dictionary learning in the Gamma-Poisson model. *Signal Processing, IEEE Transactions on*, *60*(10), 5163–5175.

Ding, B., & Gentleman, R. (2005). Classification Using Generalized Partial Least Squares. *Journal of Computational and Graphical Statistics*, *14*(2), 280–298.

Ding, C., & He, X. (2004). K-means Clustering via Principal Component Analysis. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, (pp. 29–). New York, NY, USA: ACM.

Ding, C. H., He, X., & Simon, H. D. (2005). On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. In *SDM*, vol. 5, (pp. 606–610). SIAM.

Donoho, D., & Stodden, V. (2003). When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in Neural Information Processing Systems*, (p. None).

Donoho, D. L. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, (pp. 1–32).

Dudoit, S., Fridlyand, J., & Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, *97*(457), 77–87.

Dunson, D. B., & Herring, A. H. (2005). Bayesian latent variable models for mixed discrete outcomes. *Biostatistics*, *6*(1), 11–25.

Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, *1*(3), 211–218.

Eckstein, J., & Yao, W. (2012). Augmented Lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results. *RUTCOR Research Reports*, *32*.

Eggert, J., & Korner, E. (2004). Sparse coding and NMF. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference On*, vol. 4, (pp. 2529–2533). IEEE.

Eilers, P. H., Boer, J. M., van Ommen, G.-J., & van Houwelingen, H. C. (2001). Classification of microarray data with penalized logistic regression. In *BiOS 2001 The International Symposium on Biomedical Optics*, (pp. 187–198). International Society for Optics and Photonics.

Engelhardt, B. E., & Adams, R. P. (2014). Bayesian Structured Sparsity from Gaussian Fields. *arXiv:1407.2235 [q-bio, stat]*.

Fahrmeir, L., & Tutz, G. (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer Science & Business Media.

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, *96*(456), 1348–1360.

Févotte, C., & Cemgil, A. T. (2009). Nonnegative matrix factorizations as probabilistic inference in composite models. In *Signal Processing Conference, 2009 17th European*, (pp. 1913–1917). IEEE.

Fevotte, C., & Godsill, S. J. (2006). A Bayesian approach for blind separation of sparse sources. *IEEE Transactions on Audio, Speech, and Language Processing*, *14*(6), 2174–2188.

Fink, D. (1997). A compendium of conjugate priors. *Tech Report, Cornell University*, (p. 46).

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, *80*(1), 27–38.

Fort, G., & Lambert-Lacroix, S. (2005). Classification using partial least squares with penalized logistic regression. *Bioinformatics*, *21*(7), 1104–1111.

Fort, G., Lambert-Lacroix, S., & Peyre, J. (2005). Réduction de dimension dans les modèles linéaires généralisés: application à la classification supervisée de données issues des biopuces (in French). *Journal de la société française de statistique*, *146*(1-2), 117–152.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, *33*(1), 1.

Friguet, C. (2010). *Impact de La Dépendance Dans Les Procédures de Tests Multiples En Grande Dimension*. Ph.D. thesis, Rennes, AGROCAMPUS-OUEST.

Gao, C., & Engelhardt, B. E. (2012). A sparse factor analysis model for high dimensional latent spaces. In *NIPS: Workshop on Analysis Operator Learning vs. Dictionary Learning: Fraternal Twins in Sparse Modeling*.

Gao, J. (2008). Robust L1 principal component analysis and its Bayesian variational inference. *Neural computation*, *20*(2), 555–572.

Gaublomme, J. T., Yosef, N., Lee, Y., Gertner, R. S., Yang, L. V., Wu, C., Pandolfi, P. P., Mak, T., Satija, R., Shalek, A. K., Kuchroo, V. K., Park, H., & Regev, A. (2015). Single-Cell Genomics Unveils Critical Regulators of Th17 Cell Pathogenicity. *Cell*, *163*(6), 1400–1412.

Gawad, C., Koh, W., & Quake, S. R. (2016). Single-cell genome sequencing: Current state of the science. *Nature Reviews Genetics*, *17*(3), 175–188.

Geiger, B. C., & Kubin, G. (2013). Signal enhancement as minimization of relevant information loss. In *Systems, Communication and Coding (SCC), Proceedings of 2013 9th International ITG Conference On*, (pp. 1–6). VDE.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, *6*(6), 721–741.

George, E. I., & McCulloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, *88*(423), 881–889.

Ghosh, S. (2007). Adaptive Elastic Net: An Improvement of Elastic Net to achieve Oracle Properties. *Tech Report PR 07-01, Indiana University-Purdue University, Indianapolis, USA*.

Giordano, R., Broderick, T., & Jordan, M. (2015a). Robust Inference with Variational Bayes. *arXiv preprint arXiv:1512.02578*.

Giordano, R. J., Broderick, T., & Jordan, M. I. (2015b). Linear response methods for accurate covariance estimates from mean field variational Bayes. In *Advances in Neural Information Processing Systems*, (pp. 1441–1449).

Gopalan, P., Ruiz, F. J., Ranganath, R., & Blei, D. M. (2014). Bayesian Nonparametric Poisson Factorization for Recommendation Systems. In *AISTATS*, (pp. 275–283).

Graves-Morris, P. R., Roberts, D. E., & Salam, A. (2000). The epsilon algorithm and related topics. *Journal of Computational and Applied Mathematics*, *122*(1–2), 51–80.

Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society. Series B (Methodological)*, (pp. 149–192).

Griffiths, T. L., & Ghahramani, Z. (2011). The Indian Buffet Process: An Introduction and Review. *J. Mach. Learn. Res.*, *12*, 1185–1224.

Guedj, M., Marisa, L., de Reynies, A., Orsetti, B., Schiappa, R., Bibeau, F., MacGrogan, G., Lerebours, F., Finetti, P., Longy, M., Bertheau, P., Bertrand, F., Bonnet, F., Martin, A. L., Feugeas, J. P., Bièche, I., Lehmann-Che, J., Lidereau, R., Birnbaum, D., Bertucci, F., de Thé, H., & Theillet, C. (2012). A refined molecular taxonomy of breast cancer. *Oncogene*, *31*(9), 1196–1206.

Gultekin, S., Zhang, A., & Paisley, J. (2015). Stochastic Annealing for Variational Inference. *arXiv:1505.06723 [stat]*.

Gupta, S. K., Phung, D., & Venkatesh, S. (2012). A nonparametric Bayesian Poisson Gamma model for count data. In *Pattern Recognition (ICPR), 2012 21st International Conference On*, (pp. 1815–1818). IEEE.

Hall, D. B. (2000). Zero-inflated Poisson and binomial regression with random effects: A case study. *Biometrics*, *56*(4), 1030–1039.

Han, F., & Liu, H. (2013). Principal Component Analysis on non-Gaussian Dependent Data. In *ICML (1)*, (pp. 240–248).

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY: Springer New York, second ed.

Hawkins, R. D., Hon, G. C., & Ren, B. (2010). Next-generation genomics: An integrative approach. *Nature Reviews Genetics*, *11*(7), 476–486.

Hestenes, M. R., & Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, *49*(6).

Hinneburg, A., Aggarwal, C. C., & Keim, D. A. (2000). What is the nearest neighbor in high dimensional spaces? In *26th Internat. Conference on Very Large Databases*, (pp. 506–515).

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67.

Hoffman, M., Blei, D. M., & Cook, P. R. (2010). Bayesian nonparametric matrix factorization for recorded music. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, (pp. 439–446).

Hoffman, M. D., & Blei, D. M. (2014). Structured stochastic variational inference. *arXiv preprint arXiv:1404.4114*.

Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic Variational Inference. *J. Mach. Learn. Res.*, *14*(1), 1303–1347.

Höskuldsson, A. (1988). PLS regression methods. *Journal of chemometrics*, *2*(3), 211–228.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, *24*(6), 417.

Hoyer, P. O. (2002). Non-negative sparse coding. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop On*, (pp. 557–565). IEEE.

Hoyer, P. O. (2004). Non-negative Matrix Factorization with Sparseness Constraints. *Journal of Machine Learning Research*, *5*(Nov), 1457–1469.

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural networks*, *13*(4), 411–430.

Ishwaran, H., & Rao, J. S. (2005). Spike and slab variable selection: Frequentist and Bayesian strategies. *Annals of Statistics*, (pp. 730–773).

Jaakkola, T., & Jordan, M. I. (1997). A variational approach to Bayesian logistic regression models and their extensions. In *Sixth International Workshop on Artificial Intelligence and Statistics*, vol. 82.

Jia, J., & Yu, B. (2010). On model consistency selection of the Elastic Net when p > n. *Statistica Sinica*, *20*(2), 595–611.

Johnson, N. L., Kemp, A. W., & Kotz, S. (2005). *Univariate Discrete Distributions*. John Wiley & Sons.

Jolliffe, I. T. (1982). A note on the use of principal components in regression. *Applied Statistics*, (pp. 300–303).

Jolliffe, I. T., Trendafilov, N. T., & Uddin, M. (2003). A modified principal component technique based on the LASSO. *Journal of computational and Graphical Statistics*, *12*(3), 531–547.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An Introduction to Variational Methods for Graphical Models. *Machine Learning*, *37*(2), 183–233.

Karlis, D. (2005). EM algorithm for mixed Poisson and other discrete distributions. *Astin bulletin*, *35*(01), 3–24.

Kim, H., & Park, H. (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, *23*(12), 1495–1502.

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., & others (1983). Optimization by simmulated annealing. *science*, *220*(4598), 671–680.

Klema, V., & Laub, A. (1980). The singular value decomposition: Its computation and some applications. *IEEE Transactions on automatic control*, *25*(2), 164–176.

Knowles, D., & Ghahramani, Z. (2007). Infinite sparse factor analysis and infinite independent components analysis. In *International Conference on Independent Component Analysis and Signal Separation*, (pp. 381–388). Springer.

Knowles, D., & Ghahramani, Z. (2011). Nonparametric Bayesian sparse factor models with application to gene expression modeling. *The Annals of Applied Statistics*, (pp. 1534–1552).

Krämer, N. (2007). An overview on the shrinkage properties of partial least squares regression. *Computational Statistics*, *22*(2), 249–273.

Lakshminarayanan, B., Bouchard, G., & Archambeau, C. (2011). Robust Bayesian Matrix Factorisation. In *AISTATS*, (pp. 425–433).

Lambert, D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics*, *34*(1), 1–14.

Lanczos, C. (1950). An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, *45*(4).

Landgraf, A. J., & Lee, Y. (2015). Generalized principal component analysis: Projection of saturated model parameters. *Technical Report 892, Department of Statistics, The Ohio State University*.

Laurberg, H., esbøll Christensen, M. G., Plumbley, M. D., Hansen, L. K., & Jensen, S. H. (2008). Theorems on positive data: On the uniqueness of NMF. *Computational intelligence and neuroscience*, *2008*.

Lê Cao, K.-A., Boitard, S., & Besse, P. (2011). Sparse PLS discriminant analysis: Biologically relevant feature selection and graphical displays for multiclass problems. *BMC Bioinformatics*, *12*, 253.

Lê Cao, K.-A., & Le Gall, C. (2011). Integration and variable selection of 'omics' data sets with PLS: A survey. *Journal de la Société Française de Statistique*, *152*(2), 77–96.

Lê Cao, K.-A., Rossouw, D., Robert-Granié, C., & Besse, P. (2008). A sparse PLS for variable selection when integrating omics data. *Statistical applications in genetics and molecular biology*, *7*(1).

Le Cessie, S., & Van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Applied statistics*, (pp. 191–201).

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788–791.

Lee, D. D., & Seung, H. S. (2001). Algorithms for Non-negative Matrix Factorization. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.) *Advances in Neural Information Processing Systems 13*, (pp. 556–562). MIT Press.

Lee, M., Shen, H., Huang, J. Z., & Marron, J. S. (2010). Biclustering via Sparse Singular Value Decomposition. *Biometrics*, *66*(4), 1087–1095.

Lee, S., Chugh, P. E., Shen, H., Eberle, R., & Dittmer, D. P. (2013). Poisson factor models with applications to non-normalized microRNA profiling. *Bioinformatics*, (p. btt091).

Linsker, R. (1988). Self-organization in a perceptual network. *Computer*, *21*(3), 105–117.

Liu, W., Zheng, N., & Lu, X. (2003). Non-negative matrix factorization for visual coding. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference On*, vol. 3, (pp. III–293). IEEE.

Mairal, J., Bach, F., & Ponce, J. (2012). Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(4), 791–804.

Mairal, J., Bach, F., Ponce, J., & Sapiro, G. (2009). Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, (pp. 689–696). ACM.

Malsiner-Walli, G., & Wagner, H. (2011). Comparing spike and slab priors for Bayesian variable selection. *Austrian Journal of Statistics*, *40*(4), 241–264.

Marimont, R. B., & Shapiro, M. B. (1979). Nearest Neighbour Searches and the Curse of Dimensionality. *IMA Journal of Applied Mathematics*, *24*(1), 59–70.

Marinov, G. K., Williams, B. A., McCue, K., Schroth, G. P., Gertz, J., Myers, R. M., & Wold, B. J. (2014). From single-cell to cell-pool transcriptomes: Stochasticity in gene expression and RNA splicing. *Genome Research*, *24*(3), 496–510.

Marioni, J. C., Mason, C. E., Mane, S. M., Stephens, M., & Gilad, Y. (2008). RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, *18*(9), 1509–1517.

Marx, B. D. (1996). Iteratively reweighted partial least squares estimation for generalized linear regression. *Technometrics*, *38*(4), 374–381.

McCullagh, P., & Nelder, J. A. (1989). *Generalized Linear Models, Second Edition*. CRC Press.

McGettigan, P. A. (2013). Transcriptomics in the RNA-seq era. *Current opinion in chemical biology*, *17*(1), 4–11.

McLachlan, G., & Krishnan, T. (2008). *The EM Algorithm and Extensions*. Hoboken, N.J: Wiley-Blackwell, 2nd edition ed.

Meinshausen, N., & Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(4), 417–473.

Meng, D., & De La Torre, F. (2013). Robust Matrix Factorization with Unknown Noise. In *Proceedings of the IEEE International Conference on Computer Vision*, (pp. 1337–1344).

Minka, T. (2000). *Estimating a Dirichlet Distribution*. Technical report, MIT.

Mitchell, T. J., & Beauchamp, J. J. (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, *83*(404), 1023–1032.

Moschopoulos, P. G. (1985). The distribution of the sum of independent gamma random variables. *Annals of the Institute of Statistical Mathematics*, *37*(1), 541–544.

Nathoo, F. S., Lesperance, M. L., Lawson, A. B., & Dean, C. B. (2013). Comparing variational Bayes with Markov chain Monte Carlo for Bayesian computation in neuroimaging. *Statistical methods in medical research*, *22*(4), 398–423.

Nelder, J. A., & Wedderburn, R. W. M. (1972). Generalized Linear Models. *Journal of the Royal Statistical Society. Series A (General)*, *135*(3), 370–384.

Nguyen, D. V., & Rocke, D. M. (2002). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, *18*(1), 39–50.

Obermeyer, F., Glidden, J., & Jonas, E. (2014). Scaling Nonparametric Bayesian Inference via Subsample-Annealing. In *AISTATS*, (pp. 696–705).

O'Hara, R. B., & Sillanpää, M. J. (2009). A review of Bayesian variable selection methods: What, how and which. *Bayesian Analysis*, *4*(1), 85–117.

Orbanz, P. (2009). Functional Conjugacy in Parametric Bayesian Models. *Tech Report, Columbia University*.

Paatero, P. (1997). Least squares formulation of robust non-negative factor analysis. *Chemometrics and intelligent laboratory systems*, *37*(1), 23–35.

Paisley, J., Blei, D., & Jordan, M. I. (2014). Bayesian nonnegative matrix factorization with stochastic variational inference. In *Handbook of Mixed Membership Models and Their Applications*. Chapman and Hall/CRC.

Park, T., & Casella, G. (2008). The bayesian lasso. *Journal of the American Statistical Association*, *103*(482), 681–686.

Pascual-Montano, A., Carazo, J. M., Kochi, K., Lehmann, D., & Pascual-Marqui, R. D. (2006). Nonsmooth nonnegative matrix factorization (nsNMF). *IEEE transactions on pattern analysis and machine intelligence*, *28*(3), 403–415.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, *2*(11), 559–572.

Phatak, A., & de Hoog, F. (2002). Exploiting the connection between PLS, Lanczos methods and conjugate gradients: Alternative proofs of some properties of PLS. *Journal of Chemometrics*, *16*(7), 361–367.

Phatak, A., & De Jong, S. (1997). The geometry of partial least squares. *Journal of Chemometrics*, *11*(4), 311–338.

Phatak, A., Reilly, P. M., & Penlidis, A. (2002). The asymptotic variance of the univariate PLS estimator. *Linear Algebra and its Applications*, *354*(1), 245–253.

Picelli, S., Björklund, Å. K., Faridani, O. R., Sagasser, S., Winberg, G., & Sandberg, R. (2013). Smart-seq2 for sensitive full-length transcriptome profiling in single cells. *Nature Methods*, *10*(11), 1096–1098.

Pierson, E., & Yau, C. (2015). ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*, *16*, 241.

Reuter, J. A., Spacek, D. V., & Snyder, M. P. (2015). High-Throughput Sequencing Technologies. *Molecular Cell*, *58*(4), 586–597.

Rinaldo, A. (2009). Properties and refinements of the fused lasso. *The Annals of Statistics*, *37*(5B), 2922–2952.

Rosipal, R., & Krämer, N. (2006). Overview and recent advances in partial least squares. In *Subspace, Latent Structure and Feature Selection*, (pp. 34–51). Springer.

Salakhutdinov, R., & Mnih, A. (2011). Probabilistic matrix factorization. In *NIPS*, vol. 20, (pp. 1–8).

Salmon, J., Harmany, Z., Deledalle, C.-A., & Willett, R. (2014). Poisson noise reduction with non-local PCA. *Journal of mathematical imaging and vision*, *48*(2), 279–294.

Schmidt, M. N., Winther, O., & Hansen, L. K. (2009). Bayesian non-negative matrix factorization. In *International Conference on Independent Component Analysis and Signal Separation*, (pp. 540–547). Springer.

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, *6*(2), 461–464.

Seber, G. A. F., & Lee, A. J. (2003). *Linear Regression Analysis*. Hoboken, N.J: Wiley-Blackwell, 2nd edition ed.

Shah, A., Knowles, D., & Ghahramani, Z. (2015). An Empirical Study of Stochastic Variational Inference Algorithms for the Beta Bernoulli Process. In *Proceedings of The 32nd International Conference on Machine Learning*, (pp. 1594–1603).

Shen, H., & Huang, J. Z. (2008). Sparse principal component analysis via regularized low rank matrix approximation. *Journal of multivariate analysis*, *99*(6), 1015–1034.

Shen, Y., Archambeau, C., Cornford, D., Opper, M., Shawe-Taylor, J., & Barillec, R. (2010). A comparison of variational and Markov chain Monte Carlo methods for inference in partially observed stochastic dynamic systems. *Journal of Signal Processing Systems*, *61*(1), 51–59.

Sill, M., Saadati, M., & Benner, A. (2015). Applying stability selection to consistently estimate sparse principal components in high-dimensional molecular data. *Bioinformatics*, *31*(16), 2683–2690.

Simchowitz, M. (2013). Zero-Inflated Poisson Factorization for Recommendation Systems. *Junior Independent Work (advised by D. Blei), Princeton University, Department of Mathematics*.

Singh, A. P., & Gordon, G. J. (2008). A Unified View of Matrix Factorization Models. In W. Daelemans, B. Goethals, & K. Morik (Eds.) *Machine Learning and Knowledge Discovery in Databases*, no. 5212 in Lecture Notes in Computer Science, (pp. 358–373). Springer Berlin Heidelberg.

Sra, S., & Dhillon, I. S. (2005). Generalized nonnegative matrix approximations with Bregman divergences. In *Advances in Neural Information Processing Systems*, (pp. 283–290).

Srivastava, S., & Chen, L. (2010). A two-parameter generalized Poisson model to improve the analysis of RNA-seq data. *Nucleic Acids Research*, *38*(17), e170–e170.

Stegle, O., Teichmann, S. A., & Marioni, J. C. (2015). Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, *16*(3), 133–145.

Tenenhaus, M. (1998). *La Régression PLS: Théorie et Pratique*. Editions Technip.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(1), 91–108.

Tierney, L., & Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the american statistical association*, *81*(393), 82–86.

Tipping, M. E., & Bishop, C. M. (1999). Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society, Series B*, *61*, 611–622.

Titsias, M. K. (2008). The Infinite Gamma-Poisson Feature Model. In J. C. Platt, D. Koller, Y. Singer, & S. T. Roweis (Eds.) *Advances in Neural Information Processing Systems 20*, (pp. 1513–1520). Curran Associates, Inc.

Titsias, M. K., & Lázaro-Gredilla, M. (2011). Spike and slab variational inference for multi-task and multiple kernel learning. In *Advances in Neural Information Processing Systems*, (pp. 2339–2347).

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*(Nov), 2579–2605.

Vavasis, S. A. (2009). On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, *20*(3), 1364–1377.

Wang, C.-Y., Chen, C.-T., Chiang, C.-P., Young, S.-T., Chow, S.-N., & Chiang, H. K. (1999). A Probability-based Multivariate Statistical Algorithm for Autofluorescence Spectroscopic Identification of Oral Carcinogenesis. *Photochemistry and photobiology*, *69*(4), 471–477.

Wang, H.-Q., Zheng, C.-H., & Zhao, X.-M. (2015). jNMFMA: A joint non-negative matrix factorization meta-analysis of transcriptomics data. *Bioinformatics*, *31*(4), 572–580.

Wang, J. J.-Y., Wang, X., & Gao, X. (2013). Non-negative matrix factorization by maximizing correntropy for cancer clustering. *BMC Bioinformatics*, *14*, 107.

Wang, M., Kuroda, M., Sakakihara, M., & Geng, Z. (2008). Acceleration of the EM algorithm using the vector epsilon algorithm. *Computational Statistics*, *23*(3), 469–486.

Wei, G. C., & 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.

Westling, T., & McCormick, T. H. (2015). Establishing consistency and improving uncertainty estimates of variational inference through M-estimation. *arXiv:1510.08151 [stat]*.

Witten, D. M. (2011). Classification and clustering of sequencing data using a Poisson model. *The Annals of Applied Statistics*, (pp. 2493–2518).

Witten, D. M., Tibshirani, R., & Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, *10*(3), 515–534.

Wold, H. (1975). Soft Modeling by Latent Variables; the Nonlinear Iterative Partial Least Squares Approach. *Perspectives in Probability and Statistics. Papers in Honour of M. S. Bartlett.*

Wold, S., Martens, H., & Wold, H. (1983). The multivariate calibration problem in chemistry solved by the PLS method. In *Matrix Pencils*, (pp. 286–293). Springer.

Wolf, J. B. (2013). Principles of transcriptome analysis and gene expression quantification: An RNA-seq tutorial. *Molecular ecology resources*, *13*(4), 559–572.

Wright, S. J. (2015). Coordinate descent algorithms. *Mathematical Programming*, *151*(1), 3–34.

Xu, W., Liu, X., & Gong, Y. (2003). Document Clustering Based on Non-negative Matrix Factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, (pp. 267–273). New York, NY, USA: ACM.

Yang, Z., & Michailidis, G. (2016). A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data. *Bioinformatics*, *32*(1), 1–8.

Yeung, K. Y., & Ruzzo, W. L. (2001). Principal component analysis for clustering gene expression data. *Bioinformatics*, *17*(9), 763–774.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(1), 49–67.

Zhao, Q., Meng, D., Xu, Z., Zuo, W., & Yan, Y. (2015). L1-Norm Low-Rank Matrix Factorization by Variational Bayesian Method. *IEEE Transactions on Neural Networks and Learning Systems*, *26*(4), 825–839.

Zhou, M., & Carin, L. (2012). Augment-and-Conquer Negative Binomial Processes. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.) *Advances in Neural Information Processing Systems 25*, (pp. 2546–2554). Curran Associates, Inc.

Zhou, M., Hannah, L. A., Dunson, D. B., & Carin, L. (2012). Beta-negative binomial process and Poisson factor analysis. In *In AISTATS*.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, *101*(476), 1418–1429.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), 301–320.

Zou, H., Hastie, T., & Tibshirani, R. (2006). Sparse principal component analysis. *Journal of computational and graphical statistics*, *15*(2), 265–286.

Zou, H., & Zhang, H. H. (2009). On the adaptive elastic-net with a diverging number of parameters. *Annals of statistics*, *37*(4), 1733–1751.

Publications and communications

Publications

- Durif, G., Picard, F. and Lambert-Lacroix, S. (2016). Count Matrix Factorization for zero-inflated single-cell data analysis. *In prep.*

- Durif, G., Picard, F. and Lambert-Lacroix, S. (2016). Adaptive Sparse PLS for Logistic Regression with Application to High Dimensional Classification. *In prep, preprint available on arXiv*, (`http://arxiv.org/abs/1502.05933`).

Conferences

- **Durif, G.**, Picard, F. and Lambert-Lacroix, S. (June 2016). Count Matrix Factorization and Single-Cell Data Analysis. "*Journées Ouvertes de Biologie Informatique et Mathématiques*" (JOBIM) 2016, *ENS de Lyon*, Lyon (France).

- **Durif, G.**, Picard, F. and Lambert-Lacroix, S. (January 2016). Count Matrix Factorization for Dimension Reduction and Data Visualization. Statistical Methods for Post-Genomic Data (SMPGD) 2016, Lille University, Lille (France).

- **Durif, G.**, Picard, F. and Lambert-Lacroix, S. (June 2015). Adaptive Sparse PLS for Logistic Regression: Compression, variable selection and classification. "*Groupement de Recherche (GDR) Stat et Santé*" Days, Paris-Descartes University, Paris (France).

- **Durif, G.**, Picard, F. and Lambert-Lacroix, S. (June 2015). Adaptive Sparse PLS for Logistic Regression: Compression, variable selection and classification. "*Société Française de Statistiques*" (SFdS) Days, Lille University, Lille (France).

- **Durif, G.**, Picard, F. and Lambert-Lacroix, S. (February 2015). Adaptive Sparse PLS for Logistic Regression: Compression, variable selection and classification. Statistical Methods for Post-Genomic Data (SMPGD) 2015, *Ludwig-Maximilians-Universität*, Munich (Germany).

Seminars

- Durif, G. (June 2016). Count Matrix Factorization and Single Cell Data Analysis. Statistics seminar, *Laboratoire TIMC-IMAG*, Joseph Fourier University, Grenoble (France).
- Durif, G. (June 2015). Dimension reduction, Variable selection and classification: Sparse PLS and Generalized Linear Model. Statistics seminar, *SupAgro*, Montpellier (France).
- Durif, G. (February 2015). Adaptive Sparse PLS for Logistic Regression: Compression, variable selection and classification. Statistics seminar, *Laboratoire MAP5*, Paris-Descartes University, Paris (France).
- Durif, G. (September 2014). Adaptive Sparse PLS for Logistic Regression: Compression, variable selection and classification. Statistics seminar, *Laboratoire J.-A. Dieudonné*, Nice-Sophia-Antipolis University, Nice (France).
- Durif, G. (September 2013). Multivariate analysis and dimension reduction: Sparse PLS, a comparative study. Statistics for System Biology seminar, AgroParisTech, Paris (France).

Softwares

- `R`-package `CMF` (Count Matrix Factorization): matrix factorization for zero-inflated count data, based on Gamma-Poisson latent factor model. *In development.*
- Incorporation to the `R`-package `plsgenomics`: PLS analyses for genomics data, including adaptive sparse PLS. *Available on the CRAN* (`https://cran.r-project.org/web/packages/plsgenomics/`)

Appendices

Appendix A

Natural parametrization in the exponential family

In this chapter, we recall some definitions and notations about the natural parametrization in the exponential family and about some probability distributions that are used in this manuscript.

A.1 The natural exponential family

We first focus on the natural parametrization of the exponential family. A random variable U follows a distribution of parameters $\boldsymbol{\theta}$ in the exponential family and is parametrized following the natural parametrization of the exponential family if the density (or the probability mass function if U is discrete) of U is defined such as:

$$q(u\,;\boldsymbol{\theta}) = h(u)\,\exp\left(\boldsymbol{\theta}^T\,t(u) - a(\boldsymbol{\theta})\right),$$

where:

- $h(\cdot)$ is the base measure;
- $a(\cdot)$ is the log-normalizer;
- $\boldsymbol{\theta}$ are the natural parameters;
- $t(\cdot)$ is the vector of sufficient statistics.

Then, we introduce some distributions in the exponential family that are used in this manuscript.

The Gamma distribution

Let U be a continuous random variable that follows a Gamma distribution of parameters $\alpha_1 > 0$ (shape) and $\alpha_2 > 0$ (scale). We denote $U \sim \Gamma(\alpha_1, \alpha_2)$. The natural parametrization of the Gamma distribution in the exponential family is defined as follows:

$$\begin{aligned} \boldsymbol{\theta} &= \begin{pmatrix} \alpha_1 - 1 \\ -\alpha_2 \end{pmatrix}, & h(u) &= 1\,, \\ t(u) &= \begin{pmatrix} \log u \\ u \end{pmatrix}, & a(\boldsymbol{\theta}) &= \log \Gamma(\alpha_1) - \alpha_1 \log(\alpha_2)\,, \end{aligned}$$

where $\Gamma(\cdot)$ is the Gamma function, defined by $\Gamma : z \mapsto \int_0^{+\infty} t^{z-1}\, e^{-t}\, \mathrm{d}t$ for any $z > 0$. The usual density function of U is defined on $\mathbb{R}_{>0}$ such as:

$$p(u\,;\,\alpha_1, \alpha_2) = u^{\alpha_1 - 1}\, \frac{(\alpha_2)^{\alpha_1}\, e^{-\alpha_2\, u}}{\Gamma(\alpha_1)}\,.$$

Hence, the log-density is:

$$\log p(u\,;\,\alpha_1, \alpha_2) = (\alpha_1 - 1)\log(u) + \alpha_1 \log(\alpha_2) - \alpha_2\, u - \log \Gamma(\alpha_1)$$

The expectation, variance and log-moment of U are respectively:

$$\begin{aligned} \mathbb{E}[U] &= \frac{\alpha_1}{\alpha_2}\,, \\ \mathrm{Var}(U) &= \frac{\alpha_1}{(\alpha_2)^2}\,, \\ \mathbb{E}[\log(U)] &= \psi(\alpha_1) - \log(\alpha_2)\,. \end{aligned}$$

where ψ is the digamma function, defined as $\psi(x) = \frac{\mathrm{d}}{\mathrm{d}x} \log \Gamma(x) = \frac{\Gamma'(x)}{\Gamma(x)}$.

The Poisson distribution

The Poisson distribution is a discrete distribution on the integer. Let X be a Poisson random variable of intensity (or rate) $\lambda > 0$, then $X \sim \mathscr{P}(\lambda)$. The natural parametrization of the Poisson distribution in the exponential family is:

$$\begin{aligned} \theta &= \log(\lambda)\,, & h(x) &= \frac{1}{x\,!}\,, \\ t(x) &= x\,, & a(\theta) &= e^{\theta} = \lambda\,. \end{aligned}$$

Hence, the usual probability mass function of the Poisson distribution is:

$$p(x\,;\,\lambda) = e^{-\lambda}\,\frac{\lambda^x}{x\,!}$$

and the log-likelihood is therefore

$$\log p(x\,;\,\lambda) = -\lambda + x\log(\lambda) - \log\Gamma(x+1)\,,$$

by recalling that $x\,! = \Gamma(x+1)$ for any integer x. The expectation and variance are respectively:

$$\begin{aligned} &\mathbb{E}[X] = \lambda \\ &\mathrm{Var}(X) = \lambda \end{aligned}$$

The multinomial distribution

Let $Z = (Z_1, \ldots, Z_K)$ a random vector of dimension K that follows a multinomial distribution $\mathcal{M}(N, \mathbf{p})$ where $N \in \mathbb{N}$ and $\mathbf{p} = (p_1, \ldots, p_K) \in [0,1]^K$ with $\sum_k p_k = 1$. The multinomial distribution is a generalization of the binomial distribution. The likelihood of the vector Z, assuming $\mathbf{z} \in \mathbb{N}^K$ with $\sum_k z_k = N$ is then:

$$\begin{aligned} p(\mathbf{z}\,;\,N, \mathbf{p}) = \;& \mathbf{1}_0\Big(N - \sum_k z_k\Big) \\ & \times \exp\left(\log\Gamma(N+1) + \sum_k \Big(z_k \log p_k - \log\Gamma(z_k+1)\Big)\right), \end{aligned}$$

where $\mathbf{1}_0(x) = 1$ if $x = 0$. This corresponds to the following natural parametrization in the exponential family:

$$\boldsymbol{\theta} = \begin{pmatrix} \log(p_1) \\ \vdots \\ \log(p_K) \end{pmatrix}, \qquad h(\mathbf{z}) = \frac{N!}{\prod_k z_k}\,,$$
$$t(u) = \begin{pmatrix} z_1 \\ \vdots \\ z_K \end{pmatrix}, \qquad a(\boldsymbol{\theta}) = 0\,.$$

The expectation and variance of each Z_k are respectively:

$$\mathbb{E}[Z_k] = N\, p_k\,,$$
$$\mathrm{Var}(Z_k) = N\, p_k\,(1 - p_k)\,,$$

Indeed, it is possible to show that the marginal distribution of each Z_k is a binomial distribution $\mathcal{B}(N, p_k)$. Eventually, the log-likelihood of Z is:

$$\begin{aligned} \log p(\mathbf{z}\,;\, N, \mathbf{p}) = \quad & \log\left(\mathbf{1}_0\Big(N - \sum_k z_k\Big)\right) + \log\Gamma(N+1) \\ & + \sum_k \Big(z_k \log p_k - \log\Gamma(z_k + 1)\Big)\,. \end{aligned}$$

The Bernoulli distribution

A variable Y following the Bernoulli distribution of parameter $p \in [0,1]$ takes its values in $\{0,1\}$. It is denoted by $Y \sim \mathcal{B}(p)$. The natural parametrization in the exponential family is:

$$\begin{aligned} \theta &= \log\frac{p}{1-p}\,, & h(y) &= 1\,, \\ t(y) &= y\,, & a(\theta) &= -\log(1-p) \end{aligned}\,,$$

which corresponds to the standard Bernoulli distribution:

$$p(y\,;\, p) = p^y\,(1-p)^{1-y}\,.$$

The log-likelihood is thus:

$$\log p(y\,;\, p) = y\,\log(p) + (1-y)\,\log(1-p)\,.$$

A.2 Conjugacy in the exponential family

In this section, we briefly remind the concept of conjugate prior. We consider a hierarchical model with some data x, a parameter θ. The data are assumed to be drawn from a conditional distribution $p(x \,|\, \theta)$ depending on θ. We also consider a prior distribution on the parameter θ that is $p(\theta \,;\, \alpha)$ where α is an hyper-parameter.

Thanks to the Bayes rule, the posterior distribution of θ is defined as:

$$p(\theta \,|\, x \,;\, \alpha) = \frac{p(x \,|\, \theta)\, p(\theta \,;\, \alpha)}{\int_{\vartheta} p(x \,|\, \vartheta)\, p(\vartheta \,;\, \alpha)\, \mathrm{d}\vartheta} \,.$$

If the posterior $p(\theta \,|\, x \,;\, \alpha)$ is explicit and lies in the same exponential family as the prior, then the prior distribution is called a conjugate prior for the conditional distribution $p(x \,|\, \theta)$. In this case, if the conjugate prior is defined in the natural parametrization of the exponential family such as:

$$p(\theta \,;\, \alpha) = h(\theta)\, \exp\left(\alpha^T\, t(\theta) - a(\alpha)\right) ,$$

then the posterior can be derived as:

$$p(\theta \,|\, x \,;\, \alpha) = h(\theta)\, \exp\left(\boldsymbol{\eta}(x, \alpha)^T\, t(\theta) - a(\boldsymbol{\eta}(x, \alpha))\right) ,$$

where the base measure $h(\theta)$ and the sufficient statistics $t(\theta)$ remain unchanged, but the parameter $\boldsymbol{\eta}(x, \alpha)$ depends on the data and the prior.

Appendix B

Optimization in the sparse PLS

Reformulation of sparse PLS

As introduced in Chapter 1, the sparse PLS constructs components as sparse linear combination of the covariates. When considering the first components, i.e. $\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1$, the weight vector $\mathbf{w}_1 \in \mathbb{R}^p$ is defined to maximize the empirical covariance between the component and the response, i.e. $\widehat{\text{Cov}}(\mathbf{X}\mathbf{w}, \mathbf{y}) \propto \mathbf{w}^T\mathbf{X}^T\mathbf{y}$ ($\mathbf{X}$ and $\mathbf{y}$ being centered) with a penalty on the ℓ_1-norm of $\mathbf{w}_1$ to enforce sparsity in the weights. Thus, the weight vector $\mathbf{w}_1$ is computed as the solution of the following optimization problem:

$$\begin{cases} \underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \; - \mathbf{w}^T\mathbf{X}^T\mathbf{y} + \nu \sum_j |w_j| \,, \\ \|\mathbf{w}\|_2 = 1 \text{ (additional constraint)}\,, \end{cases} \tag{B.1}$$

with $\nu > 0$. The problem (B.1) is equivalent to the following, when denoting the standard scalar product by $\langle \cdot, \cdot \rangle$:

$$\begin{cases} \underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \; - 2\left\langle \mathbf{w}\,,\, \mathbf{X}^T\mathbf{y} \right\rangle + \|\mathbf{w}\|_2^2 + 2\nu \sum_j |w_j| \,, \\ \|\mathbf{w}\|_2 = 1\,, \end{cases}$$

because the term $\|\mathbf{w}\|_2$ is constant thanks to the additional constraint. This new problem remains equivalent to the following:

$$\begin{cases} \underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \; \|\mathbf{X}^T\mathbf{y}\|_2^2 - 2\left\langle \mathbf{w}\,,\, \mathbf{X}^T\mathbf{y} \right\rangle + \|\mathbf{w}\|_2^2 + 2\nu \sum_j |w_j| \,, \\ \|\mathbf{w}\|_2 = 1\,, \end{cases}$$

since the norm of the empirical covariance $\|\mathbf{X}^T\mathbf{y}\|_2^2$ is constant. Then, thanks to the Euclidean norm properties, it can be rewritten as:

$$\begin{cases} \underset{\mathbf{w}\in\mathbb{R}^p}{\operatorname{argmin}} \|\mathbf{C}-\mathbf{w}\|_2^2 + 2\nu\,|\mathbf{w}|_1 \\ \|\mathbf{w}\|_2 = 1 \end{cases}$$

with $\mathbf{C} = \mathbf{X}^T\mathbf{y}$. Applying the method of Lagrange multipliers, the problem finally becomes:

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\operatorname{argmin}} \|\mathbf{C}-\mathbf{w}\|_2^2 + \nu'\,|\mathbf{w}|_1 + \mu\,(\|\mathbf{w}\|_2^2 - 1), \tag{B.2}$$

where $\nu' = 2\nu$. We have thus reformulated the problem defining the sparse PLS as a least squares problem with an Elastic Net penalty.

Actually, in the case of a univariate response, the formulation (B.2) is natural. Indeed, in the standard (non-sparse) PLS, the optimal weight vector $\mathbf{w}$ is the normalized dominant singular vector of the covariance matrix $\mathbf{X}^T\mathbf{y}$. However, when the response is univariate, the matrix $\mathbf{X}^T\mathbf{y}$ is a vector and the solution for $\mathbf{w}$ is the vector $\mathbf{X}^T\mathbf{y}$ (normalized to 1). This corresponds exactly to the solution of the problem:

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\operatorname{argmin}} \|\mathbf{C}-\mathbf{w}\|_2^2 + \mu\,(\|\mathbf{w}\|_2^2 - 1)$$

(without the ℓ_1 penalty).

The solution of the penalized problem (B.2) defines the first component of the sparse PLS.

Proximal operator

The problem (B.2) may be solved by the proximal gradient method. This approach uses proximal operator. We will not detail the theory here but introduce a few examples to explain our interest in such optimization methodology. A complete presentation can be found in Bach et al. (2012).

We considered the least squares problem consisting in finding the vector $\mathbf{v} \in \mathbb{R}^p$ that is the most close to a fixed vector $\mathbf{u} \in \mathbb{R}^p$ when considering a sparse-inducing penalty on the ℓ_1-norm of $\mathbf{v}$:

$$\underset{\mathbf{v}\in\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{u}-\mathbf{v}\|_2^2 + \nu\sum_{j=1}^{p}|v_j|\,.$$

It appears that the solution of this problem is given by the proximal operator $\operatorname{Prox}_{\nu\,|\cdot|_1}$ (Bach et al., 2012), defined such that the j^{th} coordinate of the solution is:

$$\left[\operatorname{Prox}_{\nu\,|\cdot|_1}(\mathbf{u})\right]_j = u_j\left(1-\frac{\nu}{|u_j|}\right)_+,$$

where $(\cdot)_+ = \max(0, \cdot)$. This correspond exactly to applying the soft-thresholding operator[1] to the coordinates of $\mathbf{u}$.

We now consider the same least squares problem but with a penalty on the ℓ_2-norm of $\mathbf{v}$, i.e. a Ridge penalty:

$$\underset{\mathbf{v}\in\mathbb{R}^p}{\operatorname{argmin}} \ \frac{1}{2}\left\|\mathbf{u}-\mathbf{v}\right\|_2^2 + \frac{\mu}{2}\sum_{j=1}^{p}|v_j|^2 \,.$$

The solution is given by the proximal operator $\operatorname{Prox}_{\frac{\mu}{2}\|\cdot\|_2}$ (Bach et al., 2012) such that:

$$\operatorname{Prox}_{\frac{\mu}{2}\|\cdot\|_2}(\mathbf{u}) = \frac{1}{1+\mu}\,\mathbf{u}\,,$$

as we would expect in the case of a least squares problem regularized by Ridge.

In fact, the proximal operators for the ℓ_1 and the ℓ_2 problems will be useful to solve the following Elastic Net problem:

$$\underset{\mathbf{v}\in\mathbb{R}^p}{\operatorname{argmin}} \ \frac{1}{2}\left\|\mathbf{u}-\mathbf{v}\right\|_2^2 + \nu\frac{\mu}{2}\sum_{j=1}^{p}|v_j|^2 + \nu\sum_{j=1}^{p}|v_j| \,. \tag{B.3}$$

The closed-form solution is given by the proximal operator $\operatorname{Prox}_{\frac{\nu\mu}{2}\|\cdot\|_2+\nu|\cdot|_1}$ that is in particular the composition of $\operatorname{Prox}_{\frac{\nu\mu}{2}\|\cdot\|_2}$ and $\operatorname{Prox}_{\nu|\cdot|_1}$ previously defined (Bach et al., 2012):

$$\operatorname{Prox}_{\frac{\nu\mu}{2}\|\cdot\|_2+\nu|\cdot|_1}(\mathbf{u}) = \operatorname{Prox}_{\frac{\nu\mu}{2}\|\cdot\|_2}\circ\operatorname{Prox}_{\nu|\cdot|_1}(\mathbf{u})\,.$$

The coordinates of the solution are then:

$$\left[\operatorname{Prox}_{\frac{\nu\mu}{2}\|\cdot\|_2+\nu|\cdot|_1}(\mathbf{u})\right]_j = \frac{1}{1+\nu\mu}\operatorname{sgn}(u_j)\left(|u_j|-\nu\right)_+ . \tag{B.4}$$

The problem (B.3) is exactly the Elastic Net problem (B.2) that defines the sparse PLS with $\mathbf{u} = \mathbf{C}$ and the argument $\mathbf{v} = \mathbf{w}$. The constant μ in (B.3) just has to be chosen so that $\|\mathbf{w}\|_2 = 1$. Therefore, the weight vector of the sparse PLS is indeed given by the soft-thresholding operator applied to the empirical covariance vector $\mathbf{C} = \mathbf{X}^T\mathbf{y}$ and then normalized as stated by Chun & Keleş (2010). In their work, they proposed another proof of this result.

[1] The soft-thresholding operator is defined as $x \mapsto \operatorname{sgn}(x)\left(|x|-\nu\right)_+$ for any $x \in \mathbb{R}$ where $\nu > 0$ is a penalty constant and $(\cdot)_+ = \max(0, \cdot)$.

Appendix C

The Negative Binomial distribution

We consider a random variable X that follows a Negative Binomial distribution $\mathcal{NB}(r, \pi)$ with the parameter r is a non-null positive integer and the probability $\pi \in (0, 1)$. The distribution of X is therefore defined as:

$$p(x\,;\,r, \pi) = \frac{(x+r-1)\,!}{x\,!\,(r-1)\,!}\,\pi^r\,(1-\pi)^x\,.$$

The expectation of the Negative Binomial distribution can be derived as follows:

$$\begin{aligned}\mathbb{E}[X] = & \sum_{x=0}^{+\infty} x\,\frac{(x+r-1)\,!}{x\,!\,(r-1)\,!}\,\pi^r\,(1-\pi)^x \\ = & \;0 + \sum_{x=1}^{+\infty} x\,\frac{(x+r-1)\,!}{x\,!\,(r-1)\,!}\,\pi^r\,(1-\pi)^x \\ = & \sum_{x=1}^{+\infty} \frac{(x+r-1)\,!}{(x-1)\,!\,(r-1)\,!}\,\pi^r\,(1-\pi)^x\end{aligned}$$

We set the change of variable $x' = x - 1$, then:

$$\begin{aligned}\mathbb{E}[X] = & \sum_{x'=0}^{+\infty} \frac{(x'+r)\,!}{(x')\,!\,(r-1)\,!}\,\pi^r\,(1-\pi)^{x'+1} \\ = & \;r\,\frac{(1-\pi)}{\pi}\,\sum_{x'=0}^{+\infty} \frac{(x'+r)\,!}{(x')\,!\,r\,!}\,\pi^{r+1}\,(1-\pi)^{x'}\end{aligned}$$

The sum corresponds to the integral of the probability function of a distribution $\mathcal{NB}(r+1, \pi)$ and thus sums up to 1. Finally, the expectation is:

$$\mathbb{E}[X] = r\,\frac{(1-\pi)}{\pi}\,.$$

Similarly, the variance can be computed as:

$$\mathrm{Var}(X) = r \, \frac{(1-\pi)}{\pi^2} \, .$$

After deriving a few properties of the Negative Binomial distribution, we show that it is equivalent to a Gamma-Poisson (GaP) model. In particular, we now suppose that X follows a conditional Poisson distribution $\mathscr{P}(\lambda)$ and consider a prior distribution $\Gamma(\alpha_1, \alpha_2)$ on the Poisson rate λ (with $\alpha_1, \alpha_2 > 0$), i.e.

$$\begin{aligned} \lambda &\sim \Gamma(\alpha_1, \alpha_2) \, , \\ X \,|\, \lambda &\sim \mathscr{P}(\lambda) \, . \end{aligned}$$

We show that the marginal distribution of the variable X in such model is a Negative Binomial distribution. The marginal distribution $p(x)$ of X can be formulated as

$$\begin{aligned} p(x) &= \int_{\mathbb{R}} p(x, \lambda) \, \mathrm{d}\lambda \\ &= \int_{\mathbb{R}} p(x \,|\, \lambda) \, p(\lambda) \, \mathrm{d}\lambda \end{aligned}$$

Based on the density of the Gamma distribution that is valued on $\mathbb{R}_{>0}$ (c.f. Chapter A), the integral is explicitly:

$$\begin{aligned} p(x) &= \int_{\mathbb{R}_{>0}} \frac{e^{-\lambda} \, (\lambda)^x}{x\,!} \, \lambda^{\alpha_1 - 1} \, (\alpha_2)^{\alpha_1} \, e^{-\alpha_2 \, \lambda} \, \frac{1}{\Gamma(\alpha_1)} \, \mathrm{d}\lambda \\ &= \frac{1}{x\,! \, \Gamma(\alpha_1)} \, (\alpha_2)^{\alpha_1} \int_{\mathbb{R}_{>0}} \lambda^{x + \alpha_1 - 1} \, e^{-\lambda (1 + \alpha_2)} \, \mathrm{d}\lambda \end{aligned}$$

In the previous integral, we set the change of variable: $\lambda' = \lambda \, (1 + \alpha_2)$. When recalling the definition of the Gamma function[1], the integral is computed as:

$$\begin{aligned} \int_{\mathbb{R}_{>0}} \lambda^{x + \alpha_1 - 1} \, e^{-\lambda (1 + \alpha_2)} \, \mathrm{d}\lambda &= \left(\frac{1}{1 + \alpha_2} \right)^{x + \alpha_1} \int_{\mathbb{R}_{>0}} (\lambda')^{x + \alpha_1 - 1} \, e^{-\lambda'} \, \mathrm{d}\lambda' \\ &= \left(\frac{1}{1 + \alpha_2} \right)^{x + \alpha_1} \Gamma(x + \alpha_1) \end{aligned}$$

Thus, the marginal distribution of X can be rewritten:

$$p(x) = \frac{\Gamma(x + \alpha_1)}{x\,! \, \Gamma(\alpha_1)} \left(\frac{1}{1 + \alpha_2} \right)^x \left(\frac{\alpha_2}{1 + \alpha_2} \right)^{\alpha_1} .$$

[1] defined as $\Gamma : z \mapsto \int_{\mathbb{R}_{>0}} t^{z-1} \, e^{-t} \, \mathrm{d}t$ for any $z > 0$

When recalling that for any non-null positive integer k, we have $k\,! = \Gamma(k)$, we can conclude that:

$$X \sim \mathcal{NB}\left(\alpha_1, \frac{\alpha_2}{1+\alpha_2}\right) .$$

The moments are explicitly computed based on the conditional expectation of X:

$$\begin{aligned}
\mathbb{E}[X] = & \ \mathbb{E}\big[\mathbb{E}[X \,|\, \lambda]\big] \\
= & \ \mathbb{E}[\lambda] \\
= & \ \frac{\alpha_1}{\alpha_2}
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{Var}(X) = & \ \mathbb{E}\big[\mathbb{E}[X^2 \,|\, \lambda]\big] - (\mathbb{E}[X])^2 \\
= & \ \mathbb{E}[\lambda + \lambda^2] - (\mathbb{E}[X])^2 \\
= & \ \frac{\alpha_1}{\alpha_2} + \frac{\alpha_1}{(\alpha_2)^2} + \left(\frac{\alpha_1}{\alpha_2}\right)^2 - \left(\frac{\alpha_1}{\alpha_2}\right)^2 \\
= & \ \frac{\alpha_1\,(1+\alpha_2)}{(\alpha_2)^2}
\end{aligned}$$

The expectation and variance verify therefore $\mathbb{E}[X] < \mathrm{Var}(X)$ as expected.

Appendix D

Gamma-Poisson factor model and variational inference

In this chapter, we derive some properties about the GaP factor model (standard, zero-inflated or sparse) introduced in Chapters 5 and 6 and some complementary results about the inference algorithms.

We first recall the definition of the standard GaP factor model:

$$\begin{aligned}
&X_{ij} = \sum_k Z_{ijk}\,,\\
&Z_{ijk}\,|\,U_{ik}, V_{jk} \sim \mathscr{P}(v_{jk}\,u_{ik})\,,\\
&U_{ik} \sim \Gamma(\alpha_{k,1}, \alpha_{k,2})\,,\\
&V_{jk} \sim \Gamma(\beta_{k,1}, \beta_{k,2})\,,
\end{aligned}$$

where the latent variables Z_{ijk} are conditionally independent and the latent factors U_{ik} and V_{jk} are independent.

D.1 Some properties of the Gamma-Poisson

D.1.1 Moments of the marginal distribution

Using conditional expectation, we can compute the moments of the marginal distribution of X_{ij} even if it is not explicit. The first-order moment is:

$$\begin{aligned}
\mathbb{E}[X_{ij}] = & \ \mathbb{E}\big[\mathbb{E}[X_{ij} \,|\, \lambda_{ij}]\big] \\
& \ \text{with the Poisson rate } \lambda_{ij}{=}\textstyle\sum_k u_{ik}\, v_{jk} \\
= & \ \mathbb{E}[\lambda_{ij}] \\
& \ \text{because the expectation of a Poisson variable is its rate} \\
= & \ \mathbb{E}[\textstyle\sum_k U_{ik}\, V_{jk}] \\
= & \ \sum_k \mathbb{E}[U_{ik}]\, \mathbb{E}[V_{jk}] \\
& \ \text{by linearity and independence between the factors} \\
= & \ \sum_k \frac{\alpha_{k,1}}{\alpha_{k,2}} \frac{\beta_{k,1}}{\beta_{k,2}} \\
& \ \text{based on the expectation of the Gamma distributions}
\end{aligned}$$

Similarly, the second-order moment is:

$$\begin{aligned}
\mathbb{E}\big[(X_{ij})^2\big] = & \ \mathbb{E}\big[\mathbb{E}[(X_{ij})^2 \,|\, \lambda_{ij}]\big] \\
= & \ \mathbb{E}[(\lambda_{ij})^2 + \lambda_{ij}] \\
& \ \text{based on the moment of second-order from a Poisson variable}
\end{aligned}$$

The variance will then be:

$$\mathrm{Var}(X_{ij}) = \mathbb{E}\big[(\lambda_{ij})^2 + \lambda_{ij}\big] - \big(\mathbb{E}[\lambda_{ij}]\big)^2 .$$

We treat each term separately. We begin with the squared expectation:

$$\begin{aligned}
\big(\mathbb{E}[\lambda_{ij}]\big)^2 = & \ \Big(\sum_k \mathbb{E}[U_{ik}]\, \mathbb{E}[V_{jk}]\Big)^2 \\
= & \ \sum_k \Big\{ \big(\mathbb{E}[U_{ik}]\big)^2 \, \big(\mathbb{E}[V_{jk}]\big)^2 \Big\} \\
& \ + \sum_{k \neq \ell} \Big\{ \big(\mathbb{E}[U_{ik}]\, \mathbb{E}[V_{jk}]\big) \, \big(\mathbb{E}[U_{i\ell}]\, \mathbb{E}[V_{j\ell}]\big) \Big\}
\end{aligned}$$

Then, we handle the expectation of the squared Poisson rate:

$$
\begin{aligned}
\mathbb{E}\big[(\lambda_{ij})^2\big] = &\ \mathbb{E}\left[\left(\textstyle\sum_k U_{ik}\, V_{jk}\right)^2\right] \\
= &\ \mathbb{E}\Big[\textstyle\sum_k (U_{ik})^2\,(V_{jk})^2\Big] + \mathbb{E}\Big[\textstyle\sum_{k\neq\ell}\big(U_{ik}\,V_{jk}\big)\,\big(U_{i\ell}\,V_{j\ell}\big)\Big] \\
= &\ \sum_k \Big\{\mathbb{E}\big[(U_{ik})^2\big]\,\mathbb{E}\big[(V_{jk}])^2\big]\Big\} \\
&+ \sum_{k\neq\ell}\Big\{\big(\mathbb{E}[U_{ik}]\,\mathbb{E}[V_{jk}]\big)\,\big(\mathbb{E}[U_{i\ell}]\,\mathbb{E}[V_{j\ell}]\big)\Big\} \\
&\text{by linearity and independence between the factors}
\end{aligned}
$$

Eventually, thanks to the subtraction:

$$
\begin{aligned}
\mathrm{Var}(X_{ij}) = &\ \sum_k \Big\{\mathbb{E}[U_{ik}]\,\mathbb{E}[V_{jk}]\Big\} + \sum_k \Big\{\mathbb{E}\big[(U_{ik})^2\big]\,\mathbb{E}\big[(V_{jk}])^2\big]\Big\} \\
&- \sum_k \Big\{\big(\mathbb{E}[U_{ik}]\big)^2\,\big(\mathbb{E}[V_{jk}]\big)^2\Big\} \\
= &\ \sum_k \Big\{\frac{\alpha_{k,1}}{\alpha_{k,2}}\,\frac{\beta_{k,1}}{\beta_{k,2}}\Big\} - \sum_k \Big\{\frac{(\alpha_{k,1})^2}{(\alpha_{k,2})^2}\,\frac{(\beta_{k,1})^2}{(\beta_{k,2})^2}\Big\} \\
&+ \sum_k \Big\{\Big(\frac{\alpha_{k,1}}{(\alpha_{k,2})^2} + \frac{(\alpha_{k,1})^2}{(\alpha_{k,2})^2}\Big)\Big(\frac{\beta_{k,1}}{(\beta_{k,2})^2} + \frac{(\beta_{k,1})^2}{(\beta_{k,2})^2}\Big)\Big\} \\
&\text{thanks to the second-order moment of Gamma distribution} \\
= &\ \sum_k \Big\{\frac{\alpha_{k,1}}{\alpha_{k,2}}\,\frac{\beta_{k,1}}{\beta_{k,2}}\Big\} + \sum_k \Big\{\frac{\alpha_{k,1}}{(\alpha_{k,2})^2}\,\frac{\beta_{k,1}}{(\beta_{k,2})^2}\Big\} \\
&+ \sum_k \Big\{\frac{\alpha_{k,1}}{(\alpha_{k,2})^2}\,\frac{(\beta_{k,1})^2}{(\beta_{k,2})^2}\Big\} + \sum_k \Big\{\frac{(\alpha_{k,1})^2}{(\alpha_{k,2})^2}\,\frac{\beta_{k,1}}{(\beta_{k,2})^2}\Big\}
\end{aligned}
$$

We have finally proven that $\mathbb{E}[X_{ij}] < \mathrm{Var}(X_{ij})$. The GaP factor model is therefore suitable to model over-dispersed data (compared to the Poisson-NMF).

D.1.2 Joint log-likelihood

When not considering the latent Poisson variable $\mathbf{Z}$, the joint log-likelihood of the GaP factor model is:

$$\begin{aligned}\log p(\mathbf{X},\mathbf{U},\mathbf{V}\,;\,\boldsymbol{\Omega}) = \quad & \log p(\mathbf{X}\,|\,\mathbf{U},\mathbf{V}) + \log p(\mathbf{U}\,;\,,\boldsymbol{\alpha}) + \log p(\mathbf{V}\,;\,\boldsymbol{\beta}) \\ = \quad & \sum_{i=1}^{n}\sum_{j=1}^{p} \log p(x_{ij}\,|\,\mathbf{u}_i,\mathbf{v}_j) \\ & + \sum_{i=1}^{n}\sum_{k=1}^{K} \log p(u_{ik}\,;\,\boldsymbol{\alpha}_k) \\ & + \sum_{j=1}^{p}\sum_{k=1}^{K} \log p(v_{jk}\,;\,\boldsymbol{\beta}_k)\,,\end{aligned}$$

where $p(x_{ij}\,|\,\mathbf{u}_i,\mathbf{v}_j)$ is the conditional distribution of X_{ij} knowing $(U_{ik})_{k=1:K}$ and $(V_{jk})_{k=1:K}$. The prior over U_{ik} (resp. V_{jk}) is $p(u_{ik}\,;\,\boldsymbol{\alpha}_k)$ (resp. $p(v_{jk}\,;\,\boldsymbol{\beta}_k)$). The vectors $\boldsymbol{\alpha}_k$ and $\boldsymbol{\beta}_k$ ($k = 1,\ldots,K$) store the corresponding prior hyper-parameters. For Gamma distributions, they are two-dimensional positive vectors, respectively $\boldsymbol{\alpha}_k = (\alpha_{k,1},\alpha_{k,2})$ and $\boldsymbol{\beta}_k = (\beta_{k,1},\beta_{k,2})$. As previously, the whole set of hyper-parameters is denoted as $\boldsymbol{\Omega} = (\boldsymbol{\alpha},\boldsymbol{\beta})$. The hyper-parameter of the prior over $\mathbf{U}$ and $\mathbf{V}$ are respectively gathered in $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_k] \in \mathbb{R}^{K\times 2}$ and $\boldsymbol{\beta} = [\boldsymbol{\beta}_k] \in \mathbb{R}^{K\times 2}$. Based on the model, the joint log-likelihood explicitly becomes:

$$\begin{aligned}\log p(\mathbf{X},\mathbf{U},\mathbf{V}\,;\,\boldsymbol{\Omega}) = \quad & \sum_{i=1}^{n}\sum_{j=1}^{p} \Big\{ x_{ij}\,\log(\textstyle\sum_k u_{ik}\,v_{jk}) - (\textstyle\sum_k u_{ik}\,v_{jk}) - (x_{ij}\,!) \Big\} \\ & + \sum_{i=1}^{n}\sum_{k=1}^{K} \Big\{ (\alpha_{k,1}-1)\,\log(u_{ik}) + \alpha_{k,1}\,\log\alpha_{k,2} \\ & \qquad\qquad - \alpha_{k,2}\,u_{ik} - \log\Gamma(\alpha_{k,1}) \Big\} \\ & + \sum_{j=1}^{p}\sum_{k=1}^{K} \Big\{ (\beta_{k,1}-1)\,\log(v_{jk}) + \beta_{k,1}\,\log\beta_{k,2} \\ & \qquad\qquad - \beta_{k,2}\,v_{jk} - \log\Gamma(\beta_{k,1}) \Big\}\,.\end{aligned}$$

In any attempt to integrate over $\mathbf{U}$ and $\mathbf{V}$ in order to derive the marginal distribution of X_{ij} or the posterior of the factors, the problematic term is the non-expandable log, i.e. $\log(\sum_k u_{ik}\,v_{jk})$. Indeed, as stated in Chapter 5, the distribution of a multiplicative and additive combination of variables following Gamma distributions does not admit a closed-form formulation.

When considering the latent Poisson variable $\mathbf{Z}$, the joint log-likelihood of the GaP factor model is more simple and defined as follows:

$$\begin{aligned}
\log p(\mathbf{X}, \mathbf{Z}, \mathbf{U}, \mathbf{V}\,;\, \boldsymbol{\Omega}) = \;& \log p(\mathbf{X} \,|\, \mathbf{Z}) + \log p(\mathbf{Z} \,|\, \mathbf{U}, \mathbf{V}) \\
& + \log p(\mathbf{U}\,;\,, \boldsymbol{\alpha}) + \log p(\mathbf{V}\,;\, \boldsymbol{\beta}) \\
= \;& \sum_{i=1}^{n} \sum_{j=1}^{p} \log p(x_{ij} \,|\, (z_{ijk})_k) \\
& + \sum_{i=1}^{n} \sum_{j=1}^{p} \sum_{k=1}^{K} \log p(z_{ijk} \,|\, u_{ik}, v_{jk}) \\
& + \sum_{i=1}^{n} \sum_{k=1}^{K} \log p(u_{ik}\,;\, \boldsymbol{\alpha}_k) \\
& + \sum_{j=1}^{p} \sum_{k=1}^{K} \log p(v_{jk}\,;\, \boldsymbol{\beta}_k)\,,
\end{aligned}$$

The term $p(x_{ij} \,|\, (z_{ijk})_k)$ is deterministic because $X_{ij} = \sum_k Z_{ijk}$. Concerning each Z_{ijk}, the Poisson conditional log-likelihood is:

$$\log p(z_{ijk} \,|\, u_{ik}, v_{jk}) = z_{ijk} \, \log(u_{ik}\, v_{jk}) - (u_{ik}\, v_{jk}) - (z_{ijk}\,!)\,.$$

In particular, the term $\log(\sum_k u_{ik}\, v_{jk})$ disappears, however it remains impossible to derive the marginal and the posterior. Nonetheless, it will be more simple to derive the variational algorithm thanks to the variables Z_{ijk}.

D.2 Variational inference for the Gamma-Poisson factor model

D.2.1 Complete conditional distribution

In our model for matrix factorization, the posteriors are not explicit, however it is possible to exactly compute the complete conditional distributions of the latent variables, i.e. the conditional distributions of each variable knowing the other latent variables and the data. To do so, third-party latent variables are introduced to quantify the Poisson decomposition of the count X_{ij} over the different latent directions (Cemgil, 2009). The set of latent variables $(Z_{ijk})_{k=1:K}$ for any fixed i and j is defined such that $X_{ij} = \sum_k Z_{ijk}$ with the following conditional distribution

$$Z_{ijk} \,|\, U_{ik}, V_{jk} \sim \mathscr{P}(U_{ik}\, V_{jk})\,.$$

Hence, the conditional distribution of the data remains $\mathscr{P}(\sum_k U_{ik}\, V_{jk})$ thanks to the additive property of the Poisson distribution. The object $\mathbf{Z} = [z_{ijk}] \in \mathbb{R}^{n\times p\times K}$ collects the realization z_{ijk} or variables Z_{ijk} depending on the context.

The complete conditional regarding each U_{ik}, V_{jk} and Z_{ijk} admits closed-form formulation. Indeed, concerning $\mathbf{Z}$, the complete conditional of the random vector $(Z_{ijk})_k$ is a Multinomial distribution (Zhou et al., 2012):

$$(Z_{ijk})_k \,|\, X_{ij}, (U_{ik})_k, (V_{jk})_k \;\sim \mathcal{M}\Big(X_{ij}, (\rho_{ijk})_k\Big)\,,$$

with $\rho_{ijk} = \frac{u_{ik}v_{jk}}{\sum_\ell u_{i\ell}v_{j\ell}}$ and therefore $\sum_k \rho_{ijk} = 1$.

Regarding the factor U_{ik}, because of the independence between latent factors and thanks to the Bayes rule, the complete conditional of U_{ik} can be reduced to:

$$p\big(u_{ik} \,|\, (z_{ijk})_j, (v_{jk})_j\big) \propto p\big((z_{ijk})_j \,|\, u_{ik}, (v_{jk})_j\big)\; p(u_{ik})\,,$$

i.e.

$$\begin{aligned} p(u_{ik} \,|\, \text{—}\,) \propto \quad & \prod_j \Big\{ \exp\Big(- u_{ik}v_{jk} + \log(u_{ik}v_{jk})z_{ijk}\Big)\; \frac{1}{z_{ijk}\,!}\Big\} \\ & \times \frac{(\alpha_{k,2})^{\alpha_{k,1}}}{\Gamma(\alpha_{k,1})}\; \exp\Big(- \alpha_{k,2}\, u_{ik} + (\alpha_{k,1} - 1)\, \log(u_{ik})\Big)\,. \end{aligned}$$

When reordering all the terms, the complete conditional becomes:

$$\begin{aligned} p(u_{ik} \,|\, \text{—}\,) \propto \quad & \exp\Big(\big(\alpha_{k,1} - 1 + \sum_j z_{ijk}\big) \log(u_{ik})\Big) \\ & \times \exp\Big(- u_{ik}\, \big(\alpha_{k,2} + \sum_j v_{jk}\big)\Big)\,. \end{aligned}$$

This corresponds explicitly to the density of a Gamma distribution, hence:

$$U_{ik} \,|\, (Z_{ijk})_j, (V_{jk})_j \sim \Gamma(\alpha_{k,1} + \textstyle\sum_j z_{ijk},\; \alpha_{k,2} + \sum_j v_{jk})\,.$$

The property that the prior and the complete conditional lie in the same exponential family is characteristic of conditionally conjugate prior. Such models are called conditionally conjugate model. Similarly the complete conditional over the factor V_{jk}, i.e. $p\big(v_{jk} \,|\, (z_{ijk})_i, (u_{ik})_i\big)$, is:

$$V_{jk} \,|\, (Z_{ijk})_i, (U_{ik})_i \sim \Gamma(\beta_{k,1} + \textstyle\sum_i z_{ijk},\; \beta_{k,2} + \sum_i u_{ik})\,.$$

The complete conditional distribution will be useful to infer the posterior distributions of the latent factors.

The parametrization of the Gamma complete conditionals for U_{ik} and V_{jk} are respectively denoted by:

$$\begin{aligned} \boldsymbol{\eta}_{ik}\big((z_{ijk})_j,(v_{jk})_j\big) &= \big(\alpha_{k,1}+\textstyle\sum_j z_{ijk},\ \alpha_{k,2}+\sum_j v_{jk}\big)^T\,,\\ \boldsymbol{\eta}_{jk}\big((z_{ijk})_i,(u_{ik})_i\big) &= \big(\beta_{k,1}+\textstyle\sum_i z_{ijk},\ \beta_{k,2}+\sum_i u_{ik}\big)^T\,, \end{aligned} \tag{D.1}$$

respectively for U_{ik} and V_{jk}. It depends explicitly on the natural parametrization of the Gamma distribution in the exponential family and will be useful in the following.

D.2.2 Stationary point

We derive the stationary point of the Evidence Lower Bound (ELBO). For instance, regarding U_{ik}, with the notation in the exponential family, we have:

$$\begin{array}{ll} \text{Prior} & p(u_{ik}\,;\boldsymbol{\alpha}_k) = h(u_{ik})\,\exp\big(\boldsymbol{\alpha}_k^T\,t(u)-a(\boldsymbol{\alpha}_k)\big)\,,\\ \text{Variational} & q(u_{ik}\,;\mathbf{a}_{ik}) = h(u_{ik})\,\exp\big(\mathbf{a}_{ik}^T\,t(u)-a(\mathbf{a}_{ik})\big)\,,\\ \text{Complete conditional} & p(u_{ik}\,|\,—) = h(u_{ik})\,\exp\big(\boldsymbol{\eta}_{ik}(—)^T\,t(u)-a\big(\boldsymbol{\eta}_{ik}(—)\big)\big)\,, \end{array}$$

where $\boldsymbol{\alpha}_k \in \mathbb{R}^2$ are the prior hyper-parameters, $\mathbf{a}_{ik} \in \mathbb{R}^2$ are the variational parameters and $\boldsymbol{\eta}_{u_{ik}}(—) \in \mathbb{R}^2$ are the parameters of the complete conditional that especially depend on the other latent variables and the data (c.f. previous section). These three different distributions lie in the same exponential family (here Gamma distributions), therefore the base measure $h(\cdot)$ and the log-normalizer $a(\cdot)$ are the same, only the parameters change.

When considering the ELBO with respect to U_{ik}, the objective $J(q)$ becomes:

$$\begin{aligned} \widetilde{J}(\mathbf{a}_{ik}) = {} & \mathbb{E}_q\big[\boldsymbol{\eta}_{ik}(—)^T\,t(v_{ik}) - a(\boldsymbol{\eta}_{ik}(—))\big]\\ & - \mathbb{E}_q\big[\mathbf{a}_{ik}^T\,t(v_{ik}) - a(\mathbf{a}_{ik})\big]\\ & + \text{const}\\ = {} & \mathbb{E}_q\big[\boldsymbol{\eta}_{ik}(—)\big]^T\,\mathbb{E}_q\big[t(v_{ik})\big] - \underbrace{\mathbb{E}_q\big[a(\boldsymbol{\eta}_{ik}(—))\big]}_{\text{const}}\\ & - \mathbf{a}_{ik}^T\,\mathbb{E}_q\big[t(v_{ik})\big] - a(\mathbf{a}_{ik})\\ & + \text{const}\,. \end{aligned}$$

Following a property of the exponential family, we have $\mathbb{E}_q\big[t(v_{ik})\big] = \nabla_{\mathbf{a}_{ik}} a(\mathbf{a}_{ik})$, hence the previous objective function is finally:

$$\widetilde{J}(\mathbf{a}_{ik}) = \mathbb{E}_q\big[\boldsymbol{\eta}_{ik}(—)\big]^T\,\nabla_{\mathbf{a}_{ik}} a(\mathbf{a}_{ik}) - \mathbf{a}_{ik}^T\,\nabla_{\mathbf{a}_{ik}} a(\mathbf{a}_{ik}) - a(\mathbf{a}_{ik}) + \text{const}\,.$$

Taking the gradient:

$$\nabla_{\mathbf{a}_{ik}} \widetilde{J} = \nabla^2_{\mathbf{a}_{ik}} a(\mathbf{a}_{ik}) \ \cdot \ \left(\mathbb{E}_q \big[\boldsymbol{\eta}_{ik}(—) \big]^T - \mathbf{a}_{ik} \right) .$$

Eventually, at the optimum:

$$\mathbf{a}_{ik} = \mathbb{E}_q \big[\boldsymbol{\eta}_{ik}(—) \big] .$$

When considering the latent variables V_{jk}, we may show similarly that the stationary points $\mathbf{b}_{jk}$ verify:

$$\mathbf{b}_{jk} = \mathbb{E}_q \big[\boldsymbol{\eta}_{jk}(—) \big] ,$$

where $\boldsymbol{\eta}_{jk}(—) \in \mathbb{R}^2$ are the parameters of the complete conditional $V_{jk} \,|\, —$.

Regarding the latent variables Z_{ijk}, the parametrization in the exponential family of the Multinomial distribution is based on the log of the probability parameters. Therefore, the stationary points $(r_{ijk})_k$ similarly verify:

$$\log(r_{ijk}) = \mathbb{E}_q[\log(\rho_{ijk})] ,$$

for any $k = 1, \ldots, K$, where $(\rho_{ijk})_k$ are the parameters of the complete conditional regarding $(Z_{ijk})_k$. Then, when expending the term $\log(\rho_{ijk})$, it follows that:

$$\log(r_{ijk}) = \mathbb{E}_q[\log(U_{ik}) + \log(V_{jk})] - \mathbb{E}_q[\log(\textstyle\sum_k U_{ik} \, V_{jk})] .$$

We apply the exponential to obtain the formulation

$$r_{ijk} \propto \exp \Big(\mathbb{E}_q[\log(U_{ik})] + \mathbb{E}_q[\log(V_{jk})] \Big)$$

Finally, as the probabilities r_{ijk} sum up to 1:

$$r_{ijk} = \frac{\exp \Big(\mathbb{E}_q[\log(U_{ik})] + \mathbb{E}_q[\log(V_{jk})] \Big)}{\sum_\ell \exp \Big(\mathbb{E}_q[\log(U_{i\ell})] + \mathbb{E}_q[\log(V_{j\ell})] \Big)}$$

D.2.3 Variational EM

We explicitly derive the M-step in the variational Expectation-Maximization (EM) algorithm. The E-step corresponds to the inference of the variational distribution q so that $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are updated in the M-step as the values that maximize the following criterion:

$$\mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}, \mathbf{U}, \mathbf{V} \, ; \, \boldsymbol{\alpha}, \boldsymbol{\beta})] .$$

Based on the density of the Gamma distribution, this objective function can be reformulated as:

$$\begin{aligned}
\widetilde{Q}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \quad & \mathbb{E}_q[\log p(\mathbf{U}, \mathbf{V}\,;\, \boldsymbol{\alpha}, \boldsymbol{\beta})] + \text{const} \\
= \quad & \sum_{i=1}^{n} \sum_{k=1}^{K} \Big\{ (\alpha_{k,1} - 1)\, \mathbb{E}_q[\log(U_{ik})] + \alpha_{k,1}\, \log \alpha_{k,2} \\
& \qquad - \alpha_{k,2}\, \mathbb{E}_q[U_{ik}] - \log \Gamma(\alpha_{k,1}) \Big\} \\
& + \sum_{j=1}^{p} \sum_{k=1}^{K} \Big\{ (\beta_{k,1} - 1)\, \mathbb{E}_q[\log(V_{jk})] + \beta_{k,1}\, \log \beta_{k,2} \\
& \qquad - \beta_{k,2}\, \mathbb{E}_q[V_{jk}] - \log \Gamma(\beta_{k,1}) \Big\} \,.
\end{aligned} \tag{D.2}$$

When deriving $\widetilde{Q}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ regarding the hyper-parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we get[1]:

$$\begin{aligned}
\frac{\partial}{\partial \alpha_{k,1}} \widetilde{Q}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \quad & \sum_{i=1}^{n} \Big\{ \mathbb{E}_q[\log(U_{ik})] \Big\} + n\, \log(\alpha_{k,2}) - n\, \psi(\alpha_{k,1}) \,, \\
\frac{\partial}{\partial \alpha_{k,2}} \widetilde{Q}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \quad & n\, \frac{\alpha_{k,1}}{\alpha_{k,2}} - \sum_{i=1}^{n} \mathbb{E}_q[U_{ik}] \,, \\
\frac{\partial}{\partial \beta_{k,1}} \widetilde{Q}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \quad & \sum_{j=1}^{p} \Big\{ \mathbb{E}_q[\log(V_{jk})] \Big\} + p\, \log(\beta_{k,2}) - p\, \psi(\beta_{k,1}) \,, \\
\frac{\partial}{\partial \beta_{k,2}} \widetilde{Q}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \quad & p\, \frac{\beta_{k,1}}{\beta_{k,2}} - \sum_{j=1}^{p} \mathbb{E}_q[V_{jk}] \,.
\end{aligned}$$

Setting the gradient to zero, the stationary point is defined as:

$$\begin{aligned}
& \psi(\alpha_{k,1}) = \log(\alpha_{k,2}) + \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_q[\log(U_{ik})] \,, \\
& \alpha_{k,2} = n\, \frac{\alpha_{k,1}}{\sum_{i=1}^{n} \mathbb{E}_q[U_{ik}]} \,, \\
& \psi(\beta_{k,1}) = \log(\beta_{k,2} + \frac{1}{p} \sum_{j=1}^{p} \mathbb{E}_q[\log(V_{jk})] \,, \\
& \beta_{k,2} = p\, \frac{\beta_{k,1}}{\sum_{j=1}^{p} \mathbb{E}_q[V_{jk}]} \,.
\end{aligned}$$

To invert the digamma function, we use the procedure explained in the box below.

[1] The digamma function ψ is defined as $\psi(x) = \frac{\partial}{\partial x} \log \Gamma(x)$ for any $x > 0$.

Inversion of the digamma function ψ

See Minka (2000). The objective is to find the root of the problem $\psi(x) - y = 0$. The digamma function is strictly increasing on $\mathbb{R}^+$, hence the root is unique. The Newton-Raphson iterative algorithm is defined such that:

$$x^{(n+1)} = x^{(n)} - \frac{\psi(x^{(n)}) - y}{\psi'(x^{(n)})}$$

The convergence is quadratic if the starting point is well chosen. We use the following asymptotic formula for $\psi(x)$:

$$\psi(x) \approx \begin{cases} \log(x - 1/2) & \text{if } x \geq 0.6 \\ -\dfrac{1}{x} + \psi(1) & \text{if } x < 0.6 \end{cases}$$

Hence we approximate ψ^{-1} to get the starting point:

$$\psi^{-1}(y) \approx x^{(0)} = \begin{cases} \exp(y) + 1/2 & \text{if } y \geq -2.22 \\ -\dfrac{1}{y - \psi(1)} & \text{if } y < -2.22 \end{cases}$$

D.3 Variational inference for the zero-inflated Gamma-Poisson model

We present here some complementary results regarding the analysis of simulated zero-inflated count data.

D.3.1 Norm of factors

As in the standard case, the variational-EM algorithm determines the number of factors that have to be considered in the zero-inflated Gamma-Poisson (ZI-GaP) model. As shown in Figure D.1, the norm of the column $\widehat{\mathbf{u}}_k$ of $\widehat{\mathbf{U}}$ is shrunk toward zero when k become larger than the true value K^*. This is observed for different number K of factors in the model. On the contrary, the Poisson-NMF and the ls-NMF do not reproduce this behavior.

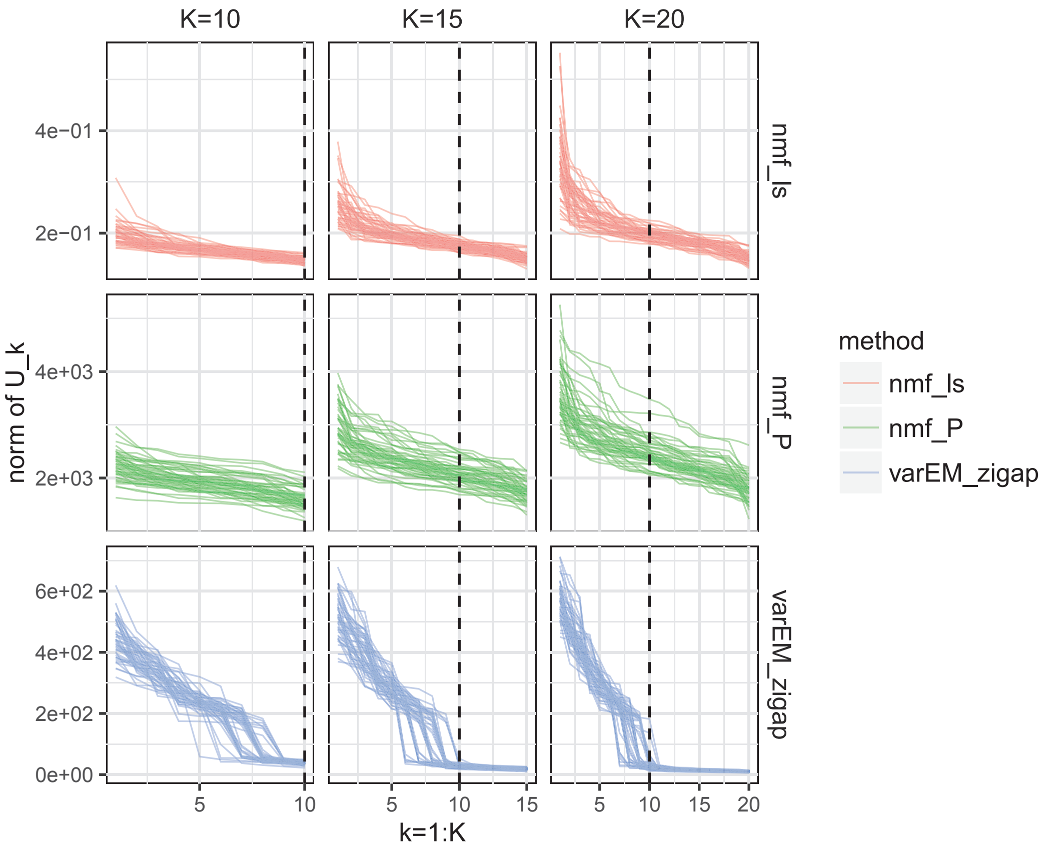


Figure D.1 – Evolution of $\|\widehat{\mathbf{u}}_k\|_2$ depending on $k = 1, \dots, K$ for the different methods Poisson-NMF (`nmf_P`), ls-NMF (`nmf_ls`) and the variational EM algorithm for the ZI-GaP model (`varEm_zigap`) when considering $K = 10, 15, 20$ factors in the model. The different trajectories corresponds to the analysis of 50 different zero-inflated data sets generated with $n = 100$, $p = 100$, $K^* = 10$ (represented by the vertical dashed line). The columns of $\widehat{\mathbf{U}}$ are sorted by decreasing value of their norm.

D.3.2 Reconstruction of the signal regarding Euclidean metric

In order to illustrate the fact that the underlying geometry associated with the Euclidean distance is not appropriate when dealing with non-Gaussian data, we now focus on the ℓ_2 distance between the true signal matrix $\mathcal{X}$ and $\widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T$, based on the Frobenius norm:

$$k \mapsto \|\mathcal{X} - \widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T\|_F\,,$$

when the factors $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$ are learned on the zero-inflated data matrix $\mathbf{X}$. We consider the SVD that finds the best approximation $\mathbf{U}\mathbf{V}^T$ of $\mathbf{X}$ regarding the least squares criterion, the ls-NMF method that is also based on the least squares criterion and we compare their results to our approach (not based on the Euclidean geometry). Figure D.2 shows the results. Our variational method finds the factors $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$ that reconstruct the true signal at best, better than the SVD and the ls-NMF. Concerning the SVD, the least squares criterion seems constant but its decreasing is just too small to be seen at this scale. This point is not surprising as the SVD approximates the corrupted $\mathbf{X}$ at best and the least squares criterion just reflects the difference between the corrupted signal and the true signal, illustrating the fact that the geometry induced by the Euclidean distance will not be appropriate when considering zero-inflated count data. On the contrary, when considering more factors in the ZI-GaP model, $\|\mathcal{X} - \widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T\|_F$ significantly decreases. When $k < 3$, the reconstruction of the SVD is better, however as soon as $k > 3$ (for any values of K), the reconstruction thanks to our method becomes better.

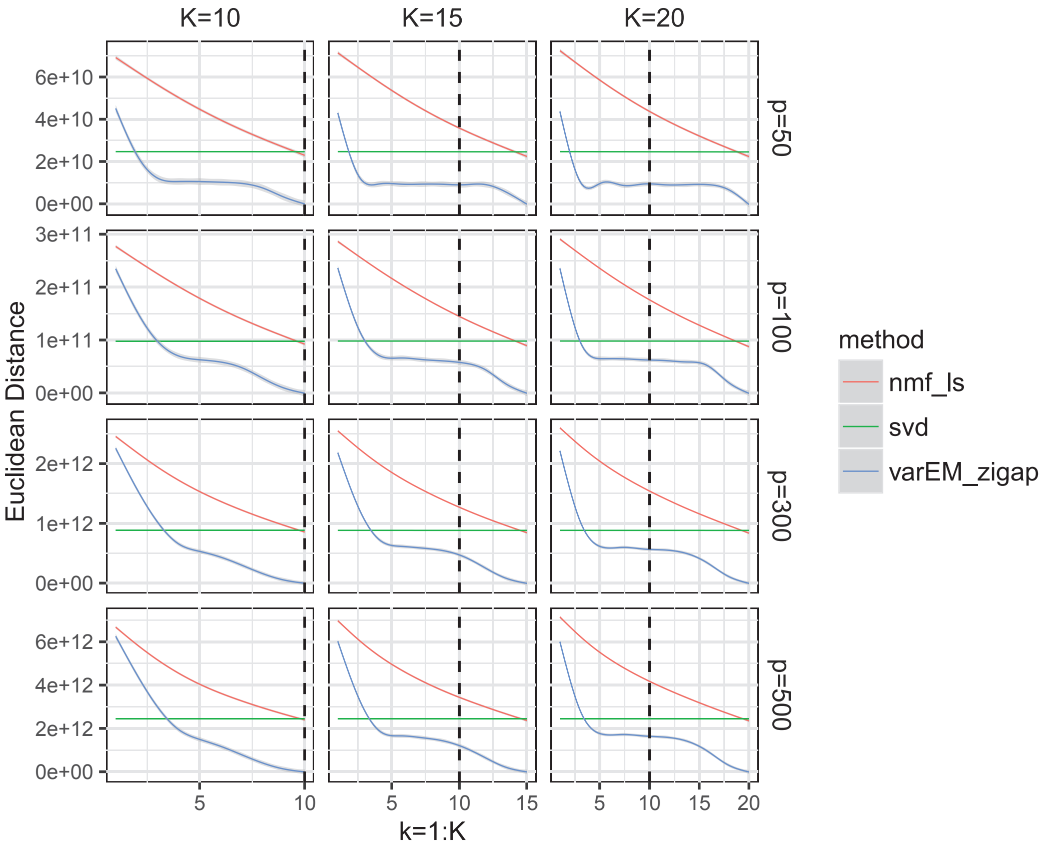
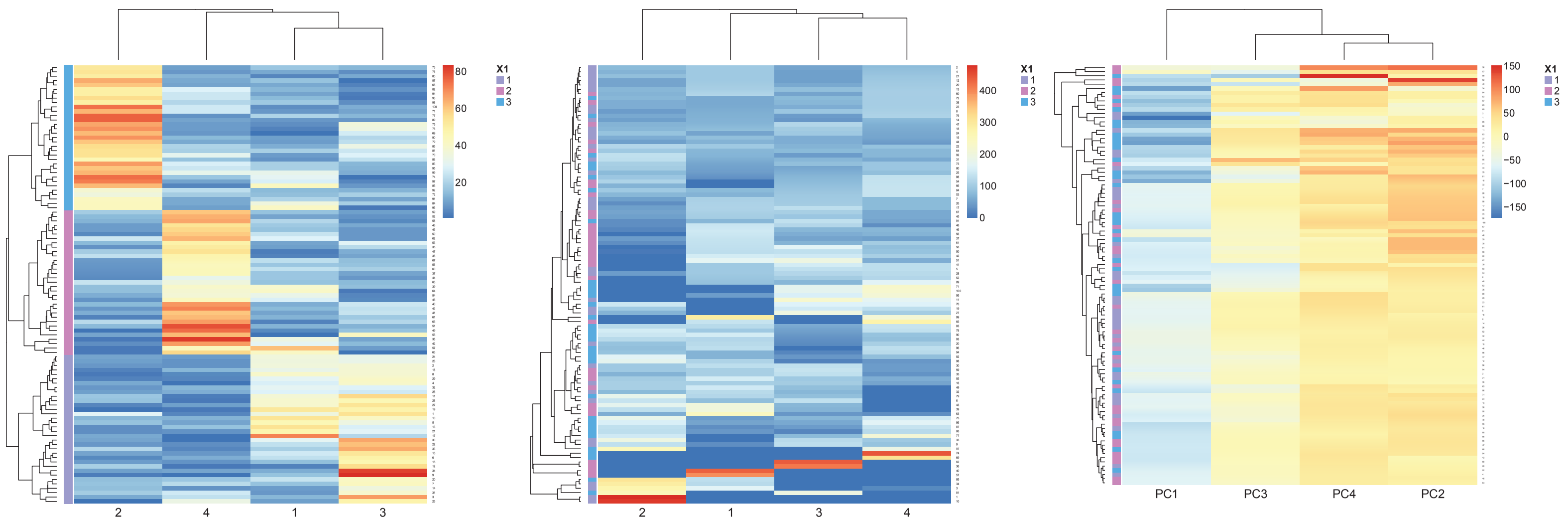


Figure D.2 – Euclidean distance $\|\mathcal{X}-\widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T\|_F$ between the true $\mathcal{X}$ and $\widehat{\mathbf{U}}_{1:k}(\widehat{\mathbf{V}}_{1:k})^T$ (learned with the zero-inflated data $\mathbf{X}$) depending on $k = 1 \ldots, K$. The data are generated with $n = 100$, $K^* = 10$ (represented by the vertical dashed line) and different values $p = 50, 100, 300, 500$. The different methods are the SVD (`svd`), the ls-NMF (`nmf_ls`) and the variational EM algorithm for the ZI-GaP model (`varEm_zigap`). For each configuration, 50 data sets are generated and fitted. The line corresponds to the average Euclidean distance over the 50 repetitions with the confidence bandwidth in shaded grey.

D.3.3 Clustering on the simulated data

We show the results of the clustering based on the rows of the matrix $\widehat{\mathbf{U}}$ when analyzing a data matrix $\mathbf{X}$ with zero-inflation, generated with $n = 100$, $p = 1000$, $K^* = 20$ and 3 groups of observations. We only show the results (c.f. Figure D.3) when considering $K = 4$ factors in the following methods: variational-EM algorithm for the ZI-GaP factor model, Poisson-NMF, PCA with a pre-transformation of the data by the Anscombe transform. When considering more factors, $K = 5, 10, 15, 20, 25$, we get similar results. The original groups are identified by our ZI-GaP model but not by the Poisson-NMF or the PCA. However, it can be noted that considering 2 factors is not sufficient to identify the original groups, even with our ZI-GaP model.



(a) (b) (c)

Figure D.3 – Clustering of the rows from $\widehat{\mathbf{U}}$ when considering $K = 4$ factors. The three colors on the left side correspond to the original two groups of observations. Model with different number of factors: (a) variational-EM algorithm for the ZI-GaP factor model (b) Poisson-NMF (c) PCA with a pre-transformation of the data by the Anscombe transform. All models were fitted on the same zero-inflated data set, where $n = 100$, $p = 1000$ and $K^* = 20$.

D.4 Complementary results

D.4.1 Computational cost

Since one of our objective was to control the computational cost of our method, we compared our variational EM algorithm to the Poisson-NMF and the SVD regarding computational time. The SVD is our baseline as its computation is almost instantaneous. For fair comparison, we fixed the same rules for the different methods. In order to improve its results, the algorithm from the `NMF` `R`-package propose to run multiple times and supply the estimated matrix $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$ as the averaged estimations over the multiple runs. We set the Non-negative Matrix Factorization (NMF) to run ten times on each data set. Therefore, we ran our variational EM with 10 random initializations. The slight difference is that, in our method, these 10 runs are not pursued until convergence. However, we decided to compare the "stock" implementation of each different algorithm that is supposed to give the best results.

Another comment is that each algorithm (which includes the multiple runs) was run sequentially. The `NMF` package provides a parallel implementation, so that the different runs are processed simultaneously. However, our computations were massively distributed on a large-scale cluster, which corresponds to a parallelization by the data, so that each individual process was running on a single core. Nonetheless, we will also implement a parallel version of our algorithm, so that it will be possible to compare the performance in the same conditions. As expected, the computational cost of the SVD is negligible and does not increase with p. It is also constant regarding the number of factor K as a single run provide all the decomposition for $K = 1, \dots, \text{rank}(\mathbf{X})$. The computational cost of our variational algorithm is larger and increases[2] with p and K, however it remains reasonable compared to the computational cost of the Poisson-NMF when p and K become larger. The multiple runs [3] to fit an averaged estimation of $\mathbf{U}$ and $\mathbf{V}$ have a huge impact on the computational performance.

To be complete, it can be noted that, when running on a realistic data set, i.e. $n = 100$ and $p = 1000$, our variational-EM algorithm (to infer a model with $K = 20$ factors) only takes a few minutes to converge, precisely ~ 180 seconds which corresponds to 10 runs of 100 iterations (for the multiple initializations) and then ~ 400 iterations on the best seed until convergence. These computational performances are directly related to an efficient implementation in `C++`.

[2] It similarly increases with the number of observations n, but we display the results for a fixed n.

[3] In the documentation, it is recommended to use much more than 10 runs.

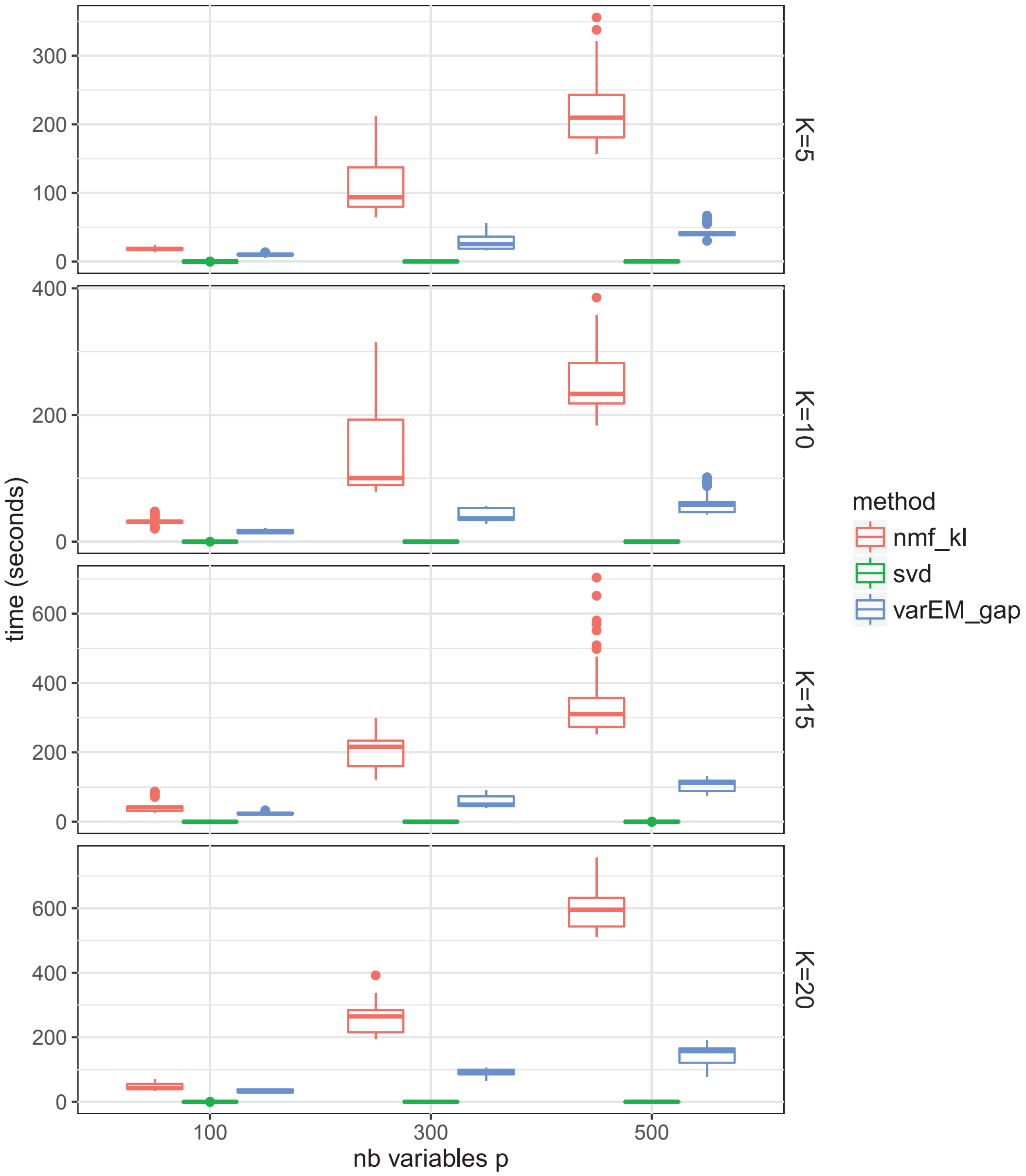


Figure D.4 – Computational time (in seconds) on a single core, depending on the number p of variables in the data (with $n = 100$ and $K^* = 10$) and the number K of factors in the model, for the different approaches: Poisson-NMF (`pnmf`), SVD (`svd`) and variational EM (`varEM_gap`).