On the Method of Moments for Estimation in Latent Variable Models

Anastasia Podosinnikova

THÈSE DE DOCTORAT
de l'Université de recherche
Paris Sciences Lettres –
PSL Research University

préparée à
l'École normale supérieure

Sur la méthode des moments pour d'estimation des modèles à variable latentes

On the method of moments for estimation in latent variable models

par Anastasia Podosinnikova

École doctorale n°386
Spécialité: Informatique
Soutenue le 01.12.2016

Composition du Jury :

Mme Animashree ANANDKUMAR
University of California Irvine
Rapporteur

M Samuel KASKI
Aalto University
Rapporteur

M Francis BACH
École normale supérieure
Directeur de thèse

M Simon LACOSTE-JULIEN
École normale supérieure
Directeur de thèse

M Alexandre D'ASPREMONT
École normale supérieure
Membre du Jury

M Pierre COMON
CNRS
Membre du Jury

M Rémi GRIBONVAL
INRIA
Membre du Jury

ENS ÉCOLE NORMALE SUPÉRIEURE | PSL RESEARCH UNIVERSITY PARIS

Abstract

Latent variable models are powerful probabilistic tools for extracting useful latent structure from otherwise unstructured data and have proved useful in numerous applications such as natural language processing and computer vision. A special case of latent variable models are latent linear models, where observations originate from a linear transformation of latent variables. Despite their modeling simplicity, latent linear models are useful and widely used instruments for data analysis in practice and include, among others, such notable examples as probabilistic principal component analysis and correlation component analysis, independent component analysis, as well as probabilistic semantic indexing and latent Dirichlet allocation. That being said, it is well known that estimation and inference are often intractable for many latent linear models and one has to make use of approximate methods often with no recovery guarantees.

One approach to address this problem, which has been popular lately, are methods based on the method of moments. These methods often have guarantees of exact recovery in the idealized setting of an infinite data sample and well specified models, but they also often come with theoretical guarantees in cases where this is not exactly satisfied. This is opposed to more standard and widely used methods based on variational inference and sampling and, therefore, makes moment matching based methods especially interesting.

In this thesis, we focus on moment matching based estimation methods for different latent linear models. In particular, by making more apparent connections between independent component analysis (ICA) and latent Dirichlet allocation (LDA), we introduce a topic model which we call *discrete ICA*. Through the close connection to ICA, which is a well understood latent linear model from the signal processing literature, we develop new estimation algorithms for the discrete ICA model with some theoretical guarantees and, in particular, with the improved sample complexity compared to the previous methods. Importantly, the discrete ICA model is *semiparametric* and the proposed estimation methods do not require any assumption on the prior distributions of the latent variables in the model.

Moreover, through the close connection between ICA and canonical correlation analysis (CCA), we propose several novel *semiparametric* multi-view models, closely related to both ICA and CCA, which are adapted to work with count or continuous data or with the mixture of the two. We prove the identifiability of these models, which is a necessary property to ensure their interpretability. It appears that some linear models which are widely used for interpretability are unidentifiable and more meaningful analogues are of interest. We also develop moment matching based estimation algorithms for the introduced semiparametric multi-view models, which again does not require any assumptions on the prior distributions of the latent variables.

For all mentioned models, we perform extensive experimental comparison of the proposed algorithms on both synthetic and real datasets and demonstrate their promising practical performance.

Acknowledgements

First and foremost, I would like to deeply thank my advisors Francis Bach and Simon Lacoste-Julien for their tremendous support, enthusiasm, motivation, and all the knowledge that they shared with me over the past three years. Francis and Simon shaped the way I think in a fundamental way, taught me to see important details and to address challenging problems in a clear and systematic way. Simon, Francis, thank you so much for this unique opportunity to work under your guidance and for the motivating environment.

I am grateful to Animashree Anandkumar and Samuel Kaski for accepting to review my thesis and to Alexandre d'Aspremont, Pierre Comon, and Rémi Gribonval for accepting to be the jury members of my thesis. I would like to express my sincere gratitude to all the jury committee members for reading the preliminary manuscript of my thesis, for their corrections, valuable feedback and challenging questions which improved this thesis.

I would like to thank Matthias Hein, with whom I worked prior to starting my PhD, for introducing me to the exciting field of machine learning and all the members of his group for their support. I would especially like to thank Martin Slawski for encouraging me to apply for a PhD position at the SIERRA project-team.

I am also thankful to Guy Bresler, Philippe Rigollet, David Sontag, Caroline Uhler from MIT for accepting me for a postdoc position.

I am very grateful to all the members (current and former) of the SIERRA and WILLOW project-teams, who are both great researches and friends, for all the support they provided over these three years.

This work was supported in part by the Microsoft Research—Inria Joint Centre and I would like to thank Laurent Massoulie for his continuous help.

Last but not least, I would like to thank my parents, my brother, all my relatives and friends for their tremendous support over these years.

Introduction

The goal of the estimation task in latent linear models is the estimation of latent parameters of a model, in particular, the estimation of a linear transformation matrix. One approach to this task is based on the method of moments, which has been more and more popular recently due to its theoretical guarantees. This thesis brings several contributions in the field of moment matching-based estimation for latent linear models in machine learning. The main part of this thesis consists of four chapters, where the first two chapters are introductory and bring together several concepts which are further used in the last two chapters for constructing new models and developing new estimation algorithms. Below we summarize main contributions of each chapter.

Chapter 1 : We start with an overview of some important latent linear models for different types of data : continuous vs. count data and single-view vs. multi-view, which we present in a unified framework with an emphasis on their identifiability properties.

Chapter 2 : Despite the simplicity of these models, the estimation and inference are often intractable. In this thesis, we focus on moment matching-based estimation methods, which often have some theoretical guarantees as opposed to widely used methods based on variational inference or sampling. In Chapter 2, we (a) review the main ideas of tensors and their decompositions, (b) connect higher-order statistics of latent linear models with the so-called canonical polyadic (CP) decomposition of tensors, and (c) briefly review the estimation and inference methods with the emphasis on the moment matching-based techniques with theoretical guarantees.

Chapter 3 : As the first contribution of this thesis, we present a novel *semiparametric* topic model—called *discrete ICA*—which is closely related to independent component analysis and such linear topic models as probabilistic latent semantic indexing and latent Dirichlet allocation, but has a higher expressive power since it does not make any assumptions on the distribution of latent variables. We prove that the higher-order cumulant-based tensors of discrete ICA, in the population case, are tensors in the form of the *symmetric CP* decomposition. This result is closely related to the previous result for higher-order moment-based tensors of LDA. However, the derivations in the discrete ICA case are somewhat more straightforward due to the properties of cumulants and the estimators have the improved sample complexity. The estimation methods are then based on the approximation of this symmetric CP decomposition from sample estimates using different (orthogonal) diagonalization algorithms, which include the eigendecomposition-based (spectral) algorithm and the tensor power method. Note that the theoretical guarantees for these algorithms extend directly to the discrete ICA case. We further improve this by using orthogonal joint

diagonalization techniques. In an extensive set of experiments on synthetic and real data, we compare these algorithms among each other and with the variational inference-based estimation methods.

Chapter 4 : As the second contribution of this thesis, we present a novel *semiparametric* linear model for multi-view data, which is closely related to the probabilistic version of canonical correlation analysis. We call this model *non-Gaussian CCA* and prove it is *identifiable* as opposed to many other related latent linear models, especially, for multi-view or aligned data. We further prove that the cumulant-based higher-order statistics of this new model, in the idealized population case, are tensors in the form of CP decomposition, i.e. they are equal to a diagonal tensor multiplied by matrices along all modes. As opposed to discrete ICA and many other latent linear models in the machine learning literature in the context of moment matching-based estimation, these CP decompositions are not symmetric. However, we show that the estimation still can be performed using algorithms for the computation of the non-symmetric CP decomposition which we refer to as *non-orthogonal joint matrix diagonalization (NOJD) algorithms by similarity* (which is only equivalent to NOJD by congruence only in the orthogonal case). Moreover, we consider another important tool from the ICA literature—*generalized covariance matrices*—which can replace cumulant tensors in this algorithmic framework, which significantly simplifies derivations. We demonstrate on a set of experiments with real and synthetic data the improved qualities of the new models and estimation method.

In final Chapter 5, we summarize the results and discuss future work. Note that the content of this thesis was previously published as Podosinnikova et al. [2015] and Podosinnikova et al. [2016]. Another publication by the author which is not included but was written while working on this thesis is Podosinnikova et al. [2014] and presents an extension of the author's Master's thesis.

Table des matières

Notation

Vectors are denoted with lower case bold letters, e.g., $\mathbf{x}$, $\mathbf{z}$, $\boldsymbol{\alpha}$, etc. Their elements are denoted with lower case non-bold letters with subscript, e.g., x_m denotes the m-th element of the vector $\mathbf{x}$ and α_k denotes the k-th element of the vector $\boldsymbol{\alpha}$. All vectors are assumed to be *column vectors*. *Matrices* are denoted with upper case bold letters, e.g., $\mathbf{D}$, $\mathbf{I}$, $\mathbf{A}$, $\mathbf{Q}$, etc. The matrix $\mathbf{I}$ always denotes the *identity matrix*. Unless otherwise specified, the matrix $\mathbf{Q}$ stands for an *orthogonal* matrix, i.e. $\mathbf{Q}^\top\mathbf{Q} = \mathbf{Q}\mathbf{Q}^\top = \mathbf{I}$, where $\mathbf{Q}^\top$ is the transpose of $\mathbf{Q}$. An element of a matrix is denoted with an upper case non-bold letter with subscripts, e.g., D_{mn} denotes the (m, n)-th element of the matrix $\mathbf{D}$. A column of a matrix is denoted with a lower case bold letter with a subscript, e.g., $\mathbf{d}_k$ stands for the k-th column of the matrix $\mathbf{D}$. *Tensors* of order-3 and higher are denoted with a capital calligraphic bold letter, e.g., $\boldsymbol{\mathcal{T}}$, $\boldsymbol{\mathcal{G}}$, etc. An element of a tensor is denoted with a capital calligraphic non-bold letter with appropriate subscripts, e.g., $\mathcal{T}_{m_1 m_2 \dots m_S}$, where S stands for the order of this tensor.

We use upper case non-bold letters to denote dimensions and sizes. For instance :

- M denotes the dimension of an observed variable, such as the dimension of the signal in the ICA context or the number of words in the vocabulary in the topic modeling context ;
- K stands for the dimension of a latent variable, such as the number of sources in the ICA context or the number of topics in the topic modeling context ;
- N stands for the number of observations in a sample, such as the number of documents in a corpus ;
- L stands for the document length (i.e. the number of tokens in a document) in the topic modeling context ;
- S is used to denote the order of a tensor, cumulant, or moment.

The indices for any dimension or size are denoted with the respective lower case letter, e.g., $m \in [M]$ or $n \in [N]$. The lower case non-bold letter i is reserved for the imaginary unit $\sqrt{-1}$.

The matrix $\mathbf{D}$ is always a real matrix of size $M \times K$ and denotes a linear transformation of the latent variables, e.g., the *mixing matrix* (in ICA), the *factor loading matrix* (in factor analysis), the *dictionary* (in dictionary learning), the *topic matrix* (in topic modeling), etc. In some cases, some structure is assumed for the matrix $\mathbf{D}$. The vector $\mathbf{x} \in \mathbb{R}^M$ always refers to an *observed variable* such as the signal (in ICA) or the count vector of a document (in topic modeling). The vector $\boldsymbol{\alpha} \in \mathbb{R}^K$ always refers to a *latent variable* such as the latent sources in ICA. If the vectors of latent variables additionally constrained to the probability simplex, e.g. the topic intensities or proportions in topic modeling, it is denoted as $\boldsymbol{\theta} \in \boldsymbol{\Delta}_K$, where the $(K-1)$-probability simplex is $\boldsymbol{\Delta}_K = \left\{\boldsymbol{\theta} \in \mathbb{R}^K : \sum_{k=1}^K \theta_k = 1, \theta_k \geq 0, \forall k \in [K]\right\}$. For a vector $\mathbf{x} \in \mathbb{R}^M$, the ℓ_2-norm is denoted as $\|\mathbf{x}\|_2 = \left(\sum_{m=1}^M x_m\right)^{1/2}$ and the ℓ_1-norm is denoted as $\|\mathbf{x}\|_1 = \sum_{m=1}^M |x_m|$. For a matrix $\mathbf{A} \in \mathbb{R}^{M \times K}$, the Frobenius norm is defined as $\|\mathbf{A}\|_F = \left(\sum_{m=1}^M \sum_{k=1}^K A_{mk}^2\right)^{1/2} = \sqrt{\mathrm{Tr}(\mathbf{A}\mathbf{A}^\top)}$. For any matrix $\mathbf{A}$,

the diagonal matrix $\text{Diag}(\mathbf{A})$ contains the diagonal values of $\mathbf{A}$ on its diagonal, i.e. $[\text{Diag}(\mathbf{A})]_{mm} = A_{mm}$. For any vector $\mathbf{a}$, the diagonal matrix $\text{Diag}(\mathbf{a})$ contains the vector $\mathbf{a}$ on its diagonal, i.e. $[\text{Diag}(\mathbf{a})]_{mm} = a_m$. The Kronecker delta $\delta(m_1, \ldots, m_S)$ is equal to 1 if and only if $m_1 = \cdots = m_S$ and 0 otherwise. In Appendix A.1, we recall some of the probability distributions used in this thesis.

For the illustration of probabilistic models, we use the standard in the graphical modeling literature *plate notation* [see, e.g., Buntine, 1994, Bishop, 2006]. In this notation, *bold black dots* denote parameters of a model; *transparent circles* stand for latent variables; *shaded circles* stand for observed variables; and *plates* denote repetitions of such (conditionally) independent variables, where the number on a plate denotes the number of such repetitions.

Chapitre 1

Latent Linear Models

Abstract

Latent linear models assume that the observations originate from a linear transformation of common latent variables and are widely used for *unsupervised* data modeling and analysis. Despite the modeling simplicity, there are a number of challenging problems that have to be addressed for such models. The *identifiability* property of latent linear models, which directly affects the *interpretability* of a model, is one of such problems. It appears that many popular latent linear models are *unidentifiable* and, therefore, unsuitable for interpretation purposes. In this chapter, we first outline some of the most important and widely used latent linear models in a unified framework with an emphasis on their identifiability properties, including : *factor analysis* (Section 1.1.2) and its particular case of *probabilistic principal component analysis* (Section 1.1.3) ; *independent component analysis* (Section 1.1.4) as its identifiable analogue ; *probabilistic canonical correlation analysis* as their extension to multi-view aligned data ; and the most basic *topic models—probabilistic latent semantic indexing* and *latent Dirichlet allocation* (Section 1.2)—designed to handle count data. This forms the modeling basis of this thesis and helps us to introduce new models in later sections.

1.1 Latent Linear Models for Single-View Data

1.1.1 Gaussian Mixture Models

One of the simplest latent linear models for continuous data is the *Gaussian mixture model (GMM)* [see, e.g., Bishop, 2006, Murphy, 2012, and references therein]. The model assumes K hidden states and a Gaussian distribution associated with each state. The generative process consists of two steps : (a) sampling the state from a discrete distribution and (b) sampling an observation from the Gaussian distribution associated to the sampled state (see a graphical representation of such model in Figure 1-1a).

To formalize this model, one introduces a latent variable z which can take one of K discrete states $\{1, 2, \ldots, K\}$. It is convenient to model z using *one-hot encoding*, i.e. as a K-vector $\mathbf{z}$ with only k-th element equal to one and the rest are zeros (the k-th canonical basis vector $\mathbf{e}_k$), which corresponds to $z = k$. The discrete prior (see (A.4) in Appendix A.1 for the definition) is then used for the state $\mathbf{z}$:

$$p(\mathbf{z}) = \text{Mult}(1, \boldsymbol{\theta}) = \prod_{k=1}^{K} \theta_k^{z_k}, \tag{1.1}$$

where the parameter $\boldsymbol{\theta}$ is constrained to the $(K-1)$-simplex, i.e. $\boldsymbol{\theta} \in \boldsymbol{\Delta}_K$. For every state k, a *base distribution* of the $\mathbb{R}^M$-valued observed variable $\mathbf{x}$ is modeled as a Gaussian distribution (A.1), i.e. $p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, which gives the conditional distribution

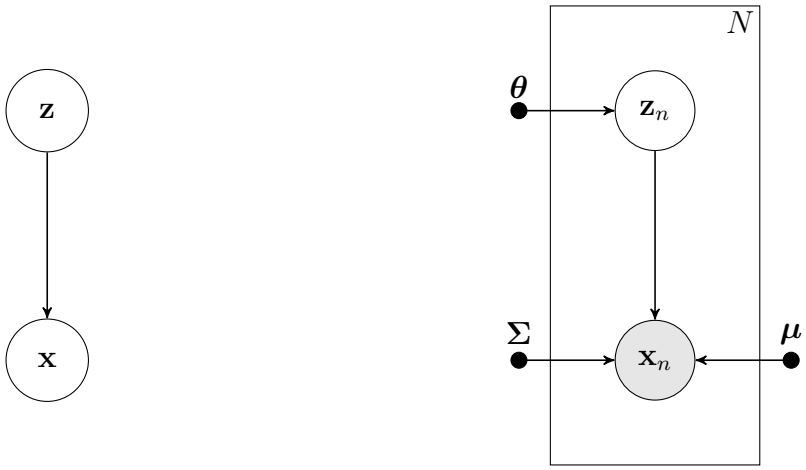
$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}. \tag{1.2}$$

Therefore, the marginal distribution of the observation variable $\mathbf{x}$ is given by

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})\, p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \theta_k\, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{1.3}$$

which is a convex combination of the base Gaussian distributions $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ or a *Gaussian mixture*, hence the name. The fact that the expectation $\mathbb{E}(\mathbf{x}) = \mathbf{D}\boldsymbol{\theta}$, where the matrix $\mathbf{D}$ is formed by stacking the centers $\boldsymbol{\mu}_k$, i.e. $\mathbf{D} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K]$, explains why the GMM belongs to *latent linear models*. The GMM is illustrated using the standard in the graphical modeling literature plate notation [Buntine, 1994, Comon and Jutten, 2010, see also Notation Section] in Figure 1-1b. By choosing different distributions as the base distributions, one can obtain mixtures of other distributions with topic models as an example (see Section 1.2).

The estimation for Gaussian mixture models is a difficult task [see, e.g., Dasgupta, 1999, Arora and Kannan, 2001, Anandkumar et al., 2012b].



(a) A GMM graphical model. (b) A GMM plate diagram.

FIGURE 1-1 – The Gaussian mixture model (GMM).

1.1.2 Factor Analysis

One problem with mixture models is that they only use a single state to generate observations, i.e. each observation can only come from one of K base distributions. Indeed, the latent variable in mixture models is represented using one-hot encoding and only one state is sampled at a time. An alternative is to use a real valued vector $\boldsymbol{\alpha} \in \mathbb{R}^K$ to represent the latent variable. The simplest choice of the prior is again a Gaussian :[1]

$$\boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{1.4}$$

It is also natural to choose a Gaussian for the conditional distribution of the continuous observation vector $\mathbf{x} \in \mathbb{R}^M$:

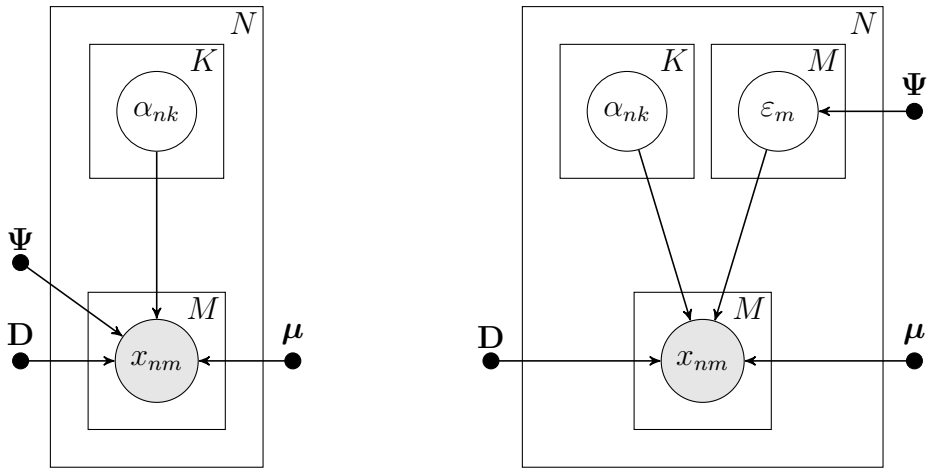
$$\mathbf{x}|\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu} + \mathbf{D}\boldsymbol{\alpha}, \boldsymbol{\Psi}), \tag{1.5}$$

where the vector $\boldsymbol{\mu} \in \mathbb{R}^M$, the matrix $\mathbf{D} \in \mathbb{R}^{M \times K}$ is called the *factor loading matrix*, and $\boldsymbol{\Psi} \in \mathbb{R}^{M \times M}$ is the covariance matrix. The elements of the latent variable are also called *factors* while the columns of $\mathbf{D}$ are called *factor loadings*. This model is known under the name of *factor analysis* [Bartholomew, 1987, Basilevsky, 1994, Bartholomew et al., 2011] and it makes the *conditional independence* assumption that the elements $x_1, x_2, \ldots x_M$ of the observed variable $\mathbf{x}$ are conditionally independent given the latent variable $\boldsymbol{\alpha}$. Therefore, the covariance matrix $\boldsymbol{\Psi}$ is *diagonal*.

It is not difficult to show that the marginal distribution of the observed variable is also a Gaussian :

$$p(\mathbf{x}) = \int p(\mathbf{x}|\boldsymbol{\alpha})\, p(\boldsymbol{\alpha})\, d\boldsymbol{\alpha} = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{D}\mathbf{D}^\top + \boldsymbol{\Psi}). \tag{1.6}$$

1. Note that the zero mean and unit covariance for the factor analysis latent variable can be chosen without loss of generality [see, e.g., Murphy, 2012].



(a) A FA plate diagram. (b) An FA generative process diagram.

FIGURE 1-2 – The factor analysis (FA) model.

Intuitively, this means that the factor analysis model explains the covariance of the observed data as a combination of two terms : (a) the independent variance associated with each coordinate (in the matrix $\boldsymbol{\Psi}$) and (b) the covariance between coordinates (captured in the matrix $\mathbf{D}$). Moreover, this representation of the covariance uses a low-rank decomposition (if $K < M$) and only $O(MK)$ parameters instead of a full covariance Gaussian with $O(M^2)$ parameters. Note, however, that if $\boldsymbol{\Psi}$ is not restricted to be diagonal, it can be trivially set to a full matrix and $\mathbf{D}$ to zero, in which case the latent factors would not be required. The factor analysis model is illustrated using the plate notation in Figure 1-2a.

One can view the factor analysis model from the generative point of view. In this case, the observed variable $\mathbf{x}$ is sampled by (a) first sampling the latent factors $\boldsymbol{\alpha}$, then (b) applying the linear transformation $\mathbf{D}$ to this sampled latent factors and the linear shift [2] $\boldsymbol{\mu}$, and finally (c) adding the Gaussian noise :

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{D}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}, \tag{1.7}$$

where the $\mathbb{R}^M$-valued additive Gaussian noise is $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$ (see an illustration using the plate notation in Figure (1-2b)). This point of view explains why factor analysis is a *latent linear model* : it is essentially a linear transformation of latent factors.

Although inference in the factor analysis model is an easy task, the model is *unidentifiable*. Indeed, the covariance of the observed variable under the factor analysis model in (1.6) has the term $\mathbf{D}\mathbf{D}^\top$. Let $\mathbf{Q}$ be an arbitrary $M \times M$ orthogonal matrix. Then right-multiplying $\mathbf{D}$ by this orthogonal matrix, i.e. $\widetilde{\mathbf{D}} = \mathbf{D}\mathbf{Q}$, does not change the distribution : $\widetilde{\mathbf{D}}\widetilde{\mathbf{D}}^\top = \mathbf{D}\,\mathbf{Q}\mathbf{Q}^\top\mathbf{D}^\top = \mathbf{D}\mathbf{D}^\top$. Thus a whole family of matrices $\widetilde{\mathbf{D}}$ gives rise

2. Note that the linear shift $\boldsymbol{\mu}$ can be omitted in practice if one preliminary centers observations by subtracting the empirical mean. Therefore, this variable is often ignored.

to the same likelihood (1.6). Geometrically, multiplying $\mathbf{D}$ by an orthogonal matrix can be seen as a rotation of the latent factors $\boldsymbol{\alpha}$ before generating the observations $\mathbf{x}$. However, since $\boldsymbol{\alpha}$ is drawn from an isotropic Gaussian, this does not influence the likelihood. Consequently, one can not uniquely identify the parameter $\mathbf{D}$, nor can one identify the latent factors $\boldsymbol{\alpha}$, independently of the type of estimation and inference methods used.

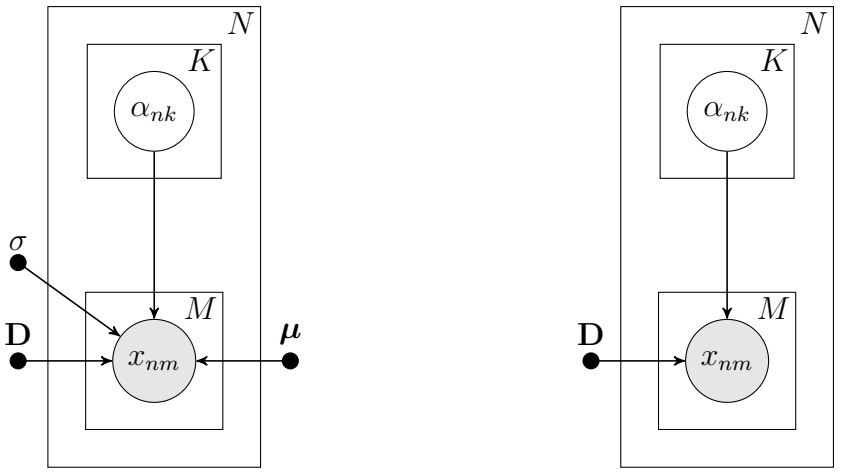
This unidentifiability does not influence the predictive performance of the factor analysis model, since the likelihood does not change. However, it does affect the factor loading matrix, and, therefore, the interpretation of the latent factors. Since factor analysis is often used to uncover the latent structure in the data, this issue causes serious problems. Numerous attempts were made to address this problem by adding additional assumptions on the model. This includes some heuristic methods for choosing a "meaningful" rotation of the latent factors, e.g., the *varimax* approach [Kaiser, 1958], which maximizes the variance of the squared loadings of a factor on all the variables. More rigorous approaches are based on adding supplementary constraints on the factor loading matrix, the most noticeable one is perhaps *sparse principal component analysis* [Zou et al., 2006], which is a separate field of research on its own [see, e.g., Archambeau and Bach, 2008, d'Aspremont et al., 2008, Journée et al., 2010, d'Aspremont et al., 2014]. An alternative approach is to use *non-Gaussian* priors for the latent factors, which is well known under the name of *independent component analysis* (see Section 1.1.4).

Factor analysis was also extended to multiway data [3] as *parallel factor analysis (Parafac)* [Harshman and Lundy, 1994], or three-mode principal component analysis [Kroonenberg, 1983]. Interestingly, Parafac is also the tensor decomposition which is used in the algorithmic framework of this thesis (see Section 2.1.2).

1.1.3 Probabilistic Principal Component Analysis

Standard *principal component analysis (PCA)* [Pearson, 1901, Jolliffe, 2002] is an algebraic tool that finds a low-dimensional subspace such that if the original data is projected onto this subspace then the variance of the projected data is maximized. It is well known that this subspace can be defined by the empirical mean of the data sample and the eigenvectors of the empirical covariance matrix. The eigenvectors of this covariance matrix, sorted in the decreasing order of the eigenvalues, are called *principal directions*. Although this PCA solution is uniquely defined (given all eigenvalues are distinct), principal component form a possible basis of the "best" low-dimensional subspace; any other basis, e.g., obtained with any orthogonal transformation of principal components, would be a solution as well. As we shall see shortly, this solution is directly related to a special case of the factor analysis model. Therefore, standard PCA partially resolves the unidentifiability of factor analysis. However, since each principal component is a linear combination of the original

3. By multiway data we mean data presented in the form of a multidimensional (i.e. dimension 3 or higher; dimension 2 corresponds to a matrix) array.



(a) A PPCA plate diagram.

(b) An ICA plate diagram.

FIGURE 1-3 – The probabilistic PCA and ICA models.

variables, the PCA solution is still difficult to interpret.

Although PCA is not necessarily considered to be a method based on Gaussian distributions, it can be justified using Gaussians. Indeed, a particular case of the factor analysis model when the covariance is isotropic, i.e. $\boldsymbol{\Psi} = \sigma^2\mathbf{I}$:

$$\begin{aligned} \boldsymbol{\alpha} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ \mathbf{x}|\boldsymbol{\alpha} &\sim \mathcal{N}(\boldsymbol{\mu} + \mathbf{D}\boldsymbol{\alpha}, \sigma^2\mathbf{I}), \end{aligned} \tag{1.8}$$

is known under the name of *probabilistic principal component analysis* (see an illustration using the plate notation in Figure 1-3a).

[Roweis, 1998, Tipping and Bishop, 1999] show a *probabilistic* interpretation of PCA : the PCA solution can be expressed as a maximum likelihood solution of the probabilistic principal component analysis model when $\sigma \to 0$. In particular, the factor loading matrix of probabilistic PCA is equal to $\mathbf{D} = \mathbf{V}(\boldsymbol{\Lambda} - \sigma^2\mathbf{I})^{1/2}\mathbf{Q}$, where $\mathbf{V}$ is the matrix with principal directions in the columns, $\boldsymbol{\Lambda}$ is the diagonal matrix with respective eigenvalues of the empirical covariance matrix on the diagonal, and $\mathbf{Q}$ is an arbitrary orthogonal matrix. This unidentifiability of probabilistic PCA is inherited from factor analysis. Therefore, PCA is unidentifiable as well. Despite the fact that the standard PCA solution is unique, PCA is defined as a subspace and the principal directions are a basis of this subspace. An arbitrary rotation of this basis does not change the subspace.

1.1.4 Independent Component Analysis

Independent component analysis (ICA) [Jutten, 1987, Jutten and Hérault, 1991, Hyvärinen et al., 2001, Comon, 1994, Comon and Jutten, 2010] was originally developed in the *blind source separation (BSS)* context. A typical BSS problem is the so called *cocktail party problem* : we are given several speakers (*sources*) and several

microphones (*sensors*), detecting a linear combination of the mixed noisy signal. The task is to *separate* the individual *sources* from the mixed *signal*.

Noisy ICA. ICA models this problem in a natural way as follows

$$\mathbf{x} = \mathbf{D}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}, \tag{1.9}$$

where the vector $\mathbf{x} \in \mathbb{R}^M$ represents the observed *signal*, the vector $\boldsymbol{\alpha} \in \mathbb{R}^K$ with *mutually independent* components stands for latent *sources*, the vector $\boldsymbol{\varepsilon} \in \mathbb{R}^M$ is the *additive noise*, and the matrix $\mathbf{D} \in \mathbb{R}^{M \times K}$ is the *mixing matrix*.

Noiseless ICA. Often, to simplify the estimation and inference in the ICA model, it is common to assume that the noise level is zero, in which case one rewrites the ICA model (1.9) as :

$$\mathbf{x} = \mathbf{D}\boldsymbol{\alpha}. \tag{1.10}$$

Alternatively, another simplifying assumption for the noisy ICA model in (1.9) is that the noise $\boldsymbol{\varepsilon}$ is Gaussian (see, e.g., Section 2.4).

Identifiability. It is straightforward to see the connection between the factor analysis formulation in (1.7) and the ICA model in (1.9). In fact, factor analysis is a special case of the ICA model where the sources and additive noise are constrained to be independent Gaussians (one can ignore the shift vector $\boldsymbol{\mu}$ since observations can be centered to have zero-mean). However, ICA generally relaxes the Gaussianity assumption, preserving only the *independence* of sources, although assumptions on the additive noise may vary. The Gaussianity assumption on the sources can be too restrictive and considering other priors can lead to models with higher expressive power. Moreover, as we mentioned in Section 1.1.2, the Gaussian latent factors (sources) are actually the reason of the unidentifiability of factor analysis. Indeed, a well known result [4] says that the mixing matrix and the latent sources of ICA are *essentially identifiable* (see below) if *at most one* source is Gaussian [Comon, 1994]. Hence, one can see ICA as an identifiable version of factor analysis.

In any case, the permutation and scaling of the mixing matrix and sources in the ICA model (as well as all other latent linear models) can never be identified. Indeed, the product $\mathbf{d}_k \alpha_k$ does not change if one simultaneously rescales (including the sign change) the terms by some non-zero constant $c \neq 0 : (c\,\mathbf{d}_k)(c^{-1}\alpha_k) = \mathbf{d}_k \alpha_k$; nor does the product $\mathbf{D}\boldsymbol{\alpha}$ change if one consistently permutes both the columns of $\mathbf{D}$ and the elements of $\boldsymbol{\alpha}$. Therefore, it only makes sense to talk about the identifiability up to permutation and scaling, which is sometimes referred to as *essential identifiability* [see, e.g., Comon and Jutten, 2010]. One can also define a *canonical form* where, e.g., the columns of the mixing matrix are constrained to have the unit ℓ_1-norm.

Independent Subspace Analysis (ISA). An interesting geometric interpretation of the permutation and scaling unidentifiability was provided by Cardoso [1998], where

4. Given the number of latent sources does not exceed the number of observations, $K \leq M$, and the mixing matrix is full rank. In Section 1.4, we briefly discuss the other case.

ICA is equivalently interpreted as the sum of vectors $\mathbf{w}_k \in \mathcal{S}_k$:

$$\mathbf{x} = \sum_{k=1}^{K} \mathbf{w}_k + \boldsymbol{\varepsilon}, \tag{1.11}$$

from one-dimensional subspaces $\mathcal{S}_k := \left\{\mathbf{w} \in \mathbb{R}^M : \mathbf{w} = \alpha \mathbf{d}_k, \alpha \in \mathbb{R}\right\}$ determined by the vectors $\mathbf{d}_k$. Each such subspace can actually be identified, given the vectors $\mathbf{d}_k$ are linearly independent, but the representation of every such subspace is clearly not unique. This gives rise to *multidimensional independent component analysis (MICA)*, which looks for orthogonal projections on (not necessary one-dimensional) subspaces $\mathcal{S}_r := \left\{\mathbf{w} \in \mathbb{R}^M : \mathbf{w} = \sum_{s=1}^{S_r} \alpha_s^{(r)} \mathbf{d}_s^{(r)}, \forall \alpha_s^{(r)} \in \mathbb{R}\right\}$, where S_r is the dimension of the r-th subspace, rather than looking for the linear transformation $\mathbf{D}\boldsymbol{\alpha}$. In such a model, the source vector consists of blocks, $\boldsymbol{\alpha} = \left(\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}, \ldots, \boldsymbol{\alpha}^{(R)}\right)$, where each block is $\boldsymbol{\alpha}^{(r)} = \left(\alpha_1^{(r)}, \alpha_2^{(r)}, \ldots, \alpha_{S_r}^{(r)}\right)$ and the total number of sources is preserved $\sum_{r=1}^{R} S_r = K$. For such sources, the independence assumption is replaced with the following : the tuples inside of one block $\alpha_1^{(r)}, \alpha_2^{(r)}, \ldots, \alpha_{S_r}^{(r)}$ can be dependent, however, the blocks $\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}, \ldots, \boldsymbol{\alpha}^{(R)}$ are mutually independent.

This model is also known under the name of *independent subspace analysis (ISA)*[5] [Hyvärinen and Hoyer, 2000]. Cardoso [1998] conjectured that the ISA problem can be solved by first solving the ICA task and then clustering the ICA elements into statistically independent groups. Szaboó et al. [2007] prove that this is indeed the case : under some additional conditions, the solution of the ISA task reduces to a permutation of the ICA task [see also Szaboó et al., 2012].

A special case of ICA estimation and inference algorithms—known as algebraic cumulant-based algorithms—are of central importance in this thesis. We describe these algorithms in Section 2.2.2 and 2.4.2. and use them to develop fast and efficient algorithms for topic models through a close connection to ICA (see Chapter 3).

1.1.5 Dictionary Learning

Another class of latent linear models is the signal processing tool called *dictionary learning* [Olshausen and Field, 1996, 1997], which targets approximation of the observed signal $\mathbf{x} \in \mathbb{R}^M$ with the linear combination of the dictionary atoms, which are columns of the matrix $\mathbf{D} \in \mathbb{R}^{M \times K}$. A special case of dictionary learning is the ℓ_1*-sparse coding* problem. It aims at minimizing $\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1$, which is well known to enforce sparsity on $\boldsymbol{\alpha}$ given the regularization parameter is chosen appropriately [Tibshirani, 1996, Chen et al., 1999]. This minimization problem is equivalent to the maximum a posteriori estimator of the noisy ICA model (1.9) where the additive noise is Gaussian and the sources are independent Laplace variables. The Laplace distribution is often considered as sparsity inducing prior since it has (slightly) heavier tails than Gaussian. Another way to see the connection between the two models

5. Note that although Hyvärinen and Hoyer [2000] make additional assumptions on the density of the source tuples, the name ISA is used in the literature in the more general setting.

is replacing the ℓ_2-distance with the KL-divergence and looking for the so-called demixing matrix which minimizes the mutual information between the demixed signals [this is one of approaches to ICA ; see, e.g., Comon and Jutten, 2010]. However, these topics are outside of the scope of this thesis.

1.2 Latent Linear Models for Count Data

1.2.1 Admixture and Topic Models

The models described in Section 1.1 are designed for *continuous* data. Similar techniques are often desirable for *count* data, i.e. *non-negative* and *discrete*, which often appears when working, e.g., with text or images. Directly applying these models to count data does not work in practice (in the noiseless setting) : the equality $\mathbf{x} = \mathbf{D}\boldsymbol{\alpha}$ is only possible when both $\mathbf{D}$ and $\boldsymbol{\alpha}$ are discrete and usually non-negative. Moreover, *negative* values, that can appear in the latent factor vectors or factor loading matrix, create interpretation problems [Buntine and Jakulin, 2006]. To fix this, one could turn count data into continuous, e.g., using the *term frequency–inverse document frequency (tf-idf)* values for text documents [Baeza-Yates and Ribeiro-Neto, 1999]. However, this does not solve the interpretability issues.

Topic Models. An algebraic adaptation of PCA to discrete data is well known as *non-negative matrix factorization* [Lee and Seung, 1999, 2001]. NMF with the KL-divergence as the objective function is equivalent to *probabilistic latent semantic indexing (pLSI)* [see Section 1.2.3 ; Hofmann, 1999a,b], which is probably the simplest and historically one of the first *probabilistic topic model*. Topic models can be seen as probabilistic latent (linear) models adapted to count data. *Latent Dirichlet allocation (LDA)* [Blei et al., 2003] is probably the most widely used topic model and extends pLSI from a discrete mixture model to an *admixture* model (see below). In fact, it was shown that pLSI is a special case [6] of LDA [Girolami and Kabán, 2003]. Buntine and Jakulin [2006] propose to use an umbrella term *discrete component analysis* for these and related models. Indeed, all these models enforce constraints on the latent variables and linear transformation matrix to preserve *non-negativity* and *discreteness*, which are intrinsic to count data.

Admixture Model for Count Data. A natural extension of the models from Section 1.1 to count data is to replace the equality $\mathbf{x} = \mathbf{D}\boldsymbol{\alpha}$ with the equality in expectation $\mathbb{E}(\mathbf{x}|\boldsymbol{\alpha}) = \mathbf{D}\boldsymbol{\alpha}$, which gives an *admixture* model [Pritchard et al., 2000] :

$$\begin{aligned} \boldsymbol{\alpha} &\sim \mathrm{PD}_{\boldsymbol{\alpha}}(\mathbf{c}_1), \\ \mathbf{x}|\boldsymbol{\alpha} &\sim \mathrm{PD}_{\mathbf{x}}(\mathbf{D}\boldsymbol{\alpha}, \mathbf{c}_2), \quad \text{such that} \quad \mathbb{E}(\mathbf{x}|\boldsymbol{\alpha}) = \mathbf{D}\boldsymbol{\alpha}, \end{aligned} \tag{1.12}$$

where $\mathrm{PD}_{\boldsymbol{\alpha}}(\cdot)$ is a continuous vector valued probability density function of the latent vectors $\boldsymbol{\alpha}$ with a hyper-parameter vector $\mathbf{c}_1$; and $\mathrm{PD}_{\mathbf{x}}(\cdot)$ is a discrete-valued probability distribution of the observation vector $\mathbf{x}$ conditioned on the latent vector $\boldsymbol{\alpha}$

6. Roughly speaking, pLSI is an ML/MAP estimate of LDA under the uniform prior on $\boldsymbol{\theta}$.

with a hyper-parameter vector $\mathbf{c}_2$ [Buntine and Jakulin, 2005]. Admixture models are *latent linear models* since the expectation of the observation vector is equal to a linear transformation of the latent vector.

1.2.2 Topic Models Terminology

Topic models [Steyvers and Griffiths, 2007, Blei and Lafferty, 2009, Blei, 2012] are probabilistic models that allow to discover thematic information in text corpora and annotate the documents using this information.

"Genetics"	**"Evolution"**	**"Disease"**	**"Computers"**
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

TABLE 1.1 – An example of topics obtained by fitting the 100-topic LDA model to 17,000 articles from the journal *Science*. Each topic is represented by the top 15 most frequent words. An example is due to Blei [2012].

Although it is common to describe topic models using the text modeling terminology, applications of topic models go far beyond information retrieval applications. For example, topic models were successfully applied in computer vision using the notion of *visual words* and the computer vision bag-of-words model [Sivic and Zisserman, 2003, Wang and Grimson, 2008, Sivic and Zisserman, 2009]. However, we restrict ourselves to the standard text corpora terminology, which we summarize below.

The *vocabulary* is a set $\mathcal{W} := \{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_M\}$ of all the words in the language. The number M of words in the vocabulary is called the *vocabulary size*. Each *word* $\boldsymbol{\omega}_m$ is represented using the one-hot encoding over M words. In the literature, the name *term* is also used to refer to a word.

The *document* is a set $\mathcal{D} := \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L\}$ of tokens $\mathbf{w}_\ell \in \mathbb{R}^M$, for $\ell \in [L]$, where a *token* is some word, i.e. $\mathbf{w}_\ell = \boldsymbol{\omega}_m$, and L is the *length* of a document. Two tokens in a document can be equal to the same word from the vocabulary, but words are

unique. The *bag-of-words* model [Baeza-Yates and Ribeiro-Neto, 1999] assumes that the order of tokens in a document does not matter. The *count vector* $\mathbf{x} \in \mathbb{R}^M$ of a document $\mathcal{D}$ is a vector with the m-th element x_m equal to the number of times the m-th word from the vocabulary appears in this document, i.e. $\mathbf{x} = \sum_{\ell=1}^{L} \mathbf{w}_\ell$.

The *corpus* is a set $\mathcal{C} := \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N\}$ of N documents. The *count matrix* $\mathbf{X}$ of this corpus is the $M \times N$ matrix with the n-th column equal to the count vector $\mathbf{x}_n$ of the n-th document. The matrix $\mathbf{X}$ is sometimes also called *(word-document) co-occurrence matrix*.

There are K topics in a model, where the k-th *topic* $\mathbf{d}_k$ is a parameter vector of a discrete distribution over the words in the vocabulary, i.e. $\mathbf{d}_k \in \boldsymbol{\Delta}_M$ (see also Figure 1.1 for an example of topics displayed as the most probable words). The m-th element of such a vector indicates the probability with which the m-th word from the vocabulary appears in the k-th topic. The matrix $\mathbf{D} \in \mathbb{R}^{M \times K}$ obtained by stacking the K topics together, $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K]$, is called the *topic matrix*. Note that in our notation $D_{mk} = d_{km}$, i.e. the index order is reverted.

We will always use the index $k \in [K]$ to refer to topics, the index $n \in [N]$ to refer to documents, the index $m \in [M]$ to refer to words from the vocabulary, and the index $\ell \in [L_n]$ to refer to tokens of the n-th document.

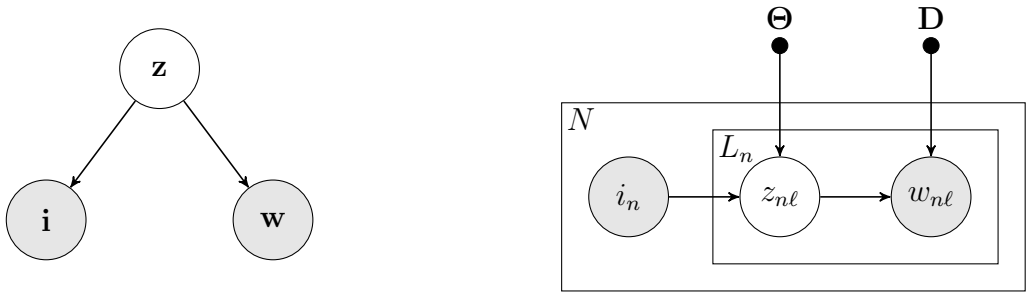
1.2.3 Probabilistic Latent Semantic Indexing

Latent Semantic Indexing. LSI [Deerwester et al., 1990] is a linear algebra tool for mapping documents to a vector space of reduced dimensionality, the *latent semantic space*. LSI is obtained as a low-rank-K approximation (see Section 2.1.3) of the (word-document) co-occurrence matrix. LSI is nearly equivalent to standard PCA : the only difference is that in LSI the documents are not centered (the mean is not subtracted) prior to computing the SVD of the co-occurrence matrix, which is normally done to preserve sparsity. The hope behind LSI is that words with the same common meaning are mapped to roughly the same direction in the latent space, which allows to compute meaningful association values between pairs of documents, even if the documents do not have any terms in common. However, LSI does not guarantee non-negative values in the latent space, which is undesirable for the interpretation purposes of non-negative count data.

Probabilistic Latent Semantic Indexing. A direct probabilistic extension of LSI is *probabilistic latent semantic indexing (pLSI)* [Hofmann, 1999a,b]. The pLSI model is a discrete mixture model and, similarly to the Gaussian mixture model (see Section 1.1.1), the latent variable z of the pLSI model can take one of K states, modeled as before by the K-vector $\mathbf{z}$ with the one-hot encoding. The observed variables are documents, modeled as the N-vector $\mathbf{i}$ with the one-hot encoding, and tokens, modeled as the M-vectors $\mathbf{w}$ with the one-hot encoding.

The generative pLSI model of a token in the so-called *symmetric*[7] parametrization (a)

7. A more common *asymmetric* parametrization of pLSI is described below.



(a) Symmetric pLSI.

(b) Asymmetric pLSI.

FIGURE 1-4 – The probabilistic latent semantic indexing model (pLSI).

first picks the topic $\mathbf{z}$ and then, given this topic, (b) picks the document $\mathbf{i}$ and (c) picks the token $\mathbf{w}$ for the picked document form the discrete distribution characterized by the k-th topic $\mathbf{d}_k$, where k is such that $z_k = 1$. This gives the following joint probability model :

$$p(\mathbf{i}, \mathbf{w}) = \sum_{\mathbf{z}} p(\mathbf{z})\, p(\mathbf{i}|\mathbf{z})\, p(\mathbf{w}|\mathbf{z}). \tag{1.13}$$

The respective graphical model is illustrated in Figure 1-4a. It is interesting to notice that in this formulation pLSI can be seen as a model for working with multi-view data (see Section 1.3 and Chapter 4) and directly admits extension to more than two or three views (see the explanation under the probabilistic interpretation of the non-negative CP decomposition of tensors in Section 2.1.2). Therefore, pLSI easily extends to model co-occurrence of three and more variables. It is also well known [Gaussier and Goutte, 2005, Ding et al., 2006, 2008] that pLSI can be seen as probabilistic interpretation [8] of *non-negative matrix factorization (NMF)* [Lee and Seung, 1999, 2001]. Therefore, the mentioned multi-view extension of pLSI can be seen as a probabilistic interpretation of the *non-negative canonical polyadic (NCP) decomposition* of tensors (see Section 2.1.2).

The symmetric pLSI model makes the following two *independence* assumptions : (a) the *bag-of-words* assumption, i.e. the observation pairs $(\mathbf{i}, \mathbf{w})$ are assumed to be generated independently and (b) tokens $\mathbf{w}$ are generated *conditionally independent* of the specific document identity $\mathbf{i}$ given the latent class (topic) $\mathbf{z}$.

Such model allows to handle (a) *polysemous* words, i.e. words that may have multiple senses and multiple types of usage in a different context, and (b) *synonyms*, i.e. different words that may have similar meaning or denote the same context.

It is not difficult to show with the help of the Bayes rule applied to $p(\mathbf{i}|\mathbf{z})$, that the joint density in (1.13) can be equivalently represented as

$$p(\mathbf{i}, \mathbf{w}) = p(\mathbf{i})\, p(\mathbf{w}|\mathbf{i}), \quad p(\mathbf{w}|\mathbf{i}) = \sum_{\mathbf{z}} p(\mathbf{w}|\mathbf{z})\, p(\mathbf{z}|\mathbf{i}), \tag{1.14}$$

8. The maximum likelihood formulation of pLSI is equivalent to the minimization of KL divergence for NMF.

which is known under the name of the *asymmetric* parametrization (see an illustration using the plate notation in Figure 1-4b).

The conditional distribution $p(\mathbf{z}|\mathbf{i})$ is the discrete distribution with the parameter $\boldsymbol{\theta}_n \in \boldsymbol{\Delta}_K$, where n is such that $i_n = 1$:

$$p(\mathbf{z}|\mathbf{i}) = \prod_{k=1}^{K} \theta_{nk}^{z_k}.$$

Note that for each document, there is one parameter $\boldsymbol{\theta}_n$; one can form a matrix $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N]$ of these parameters. Substituting $p(\mathbf{z}|\mathbf{i})$ into the expression for $p(\mathbf{w}|\mathbf{i})$, we obtain :

$$p(\mathbf{w}|\mathbf{i}) = \sum_{k=1}^{K} \theta_{nk} p(\mathbf{w}|\mathbf{z}_k), \tag{1.15}$$

where we used $\mathbf{z}_k$ to emphasize that this is the vector $\mathbf{z}$ with the k-th element equal to 1. Therefore, the conditional distribution of the tokens in a document is the *mixture of discrete distributions* $p(\mathbf{w}|\mathbf{z}_k)$ over the vocabulary of M words with K latent topics. Substituting the discrete distributions $p(\mathbf{w}|\mathbf{z}_k)$ with the parameters $\mathbf{d}_k$ into the conditional distribution (1.15), we obtain :

$$p(\mathbf{w}|\mathbf{i}) = \sum_{k=1}^{K} \theta_{nk} \prod_{m=1}^{M} d_{km}^{w_m} \overset{(a)}{=} \sum_{k=1}^{K} \prod_{m=1}^{M} (\theta_{nk} d_{km})^{w_m} = \prod_{m=1}^{M} \left[\sum_{k=1}^{K} \theta_{nk} d_{km} \right]^{w_m},$$

where, in the second equality (a), we could exchange the sum and the product due to the special form of the one-hot encoded vector $\mathbf{w}$. The sum in the brackets is actually the m-th element of the vector $\mathbf{D}\boldsymbol{\theta}_n$ and the conditional distribution $p(\mathbf{w}|\mathbf{i}_n)$ of tokens in the n-th document is the discrete distribution with the parameter $\mathbf{D}\boldsymbol{\theta}_n$:

$$\mathbf{w}|\mathbf{i}_n \sim \text{Mult}(1, [\mathbf{D}\boldsymbol{\theta}_n]),$$

where we used $\mathbf{i}_n$ to emphasize that this is the n-th document. This demonstrates that pLSI is a *latent linear model* : indeed, $\mathbb{E}(\mathbf{w}|\mathbf{i}_n) = \mathbf{D}\boldsymbol{\theta}_n$.

A significant drawback of pLSI is the *number of parameters* : the matrix $\boldsymbol{\Theta} \in \mathbb{R}^{K \times N}$ and the topic matrix $\mathbf{D} \in \mathbb{R}^{M \times K}$ give in total $O(NK + MK)$ parameters, i.e. the number of parameters grows linearly not only in the number of topics K and the number of words in the vocabulary M, but also in the number of documents N. Therefore, pLSI is prone to *overfitting* [Blei et al., 2003].

1.2.4 Latent Dirichlet Allocation

Instead of defining a parameter for each document as $\boldsymbol{\theta}_n$ in pLSI, *latent Dirichlet allocation (LDA)* [Blei et al., 2003] defines a single parameter $\mathbf{c} \in \mathbb{R}_{++}^K$ for a corpus and the *topic intensities* or *topic proportions* $\boldsymbol{\theta}$ for each document are modeled as another latent variable. This $\boldsymbol{\Delta}_K$-valued random variable $\boldsymbol{\theta}$ has the Dirichlet distribution

(hence, the name) since it is the *conjugate* prior to the multinomial distribution, which simplifies the estimation and inference procedure in the variational inference framework (see Section 2.4). However, note that although formally the number of parameters in the LDA model, $O(KM)$, is lower than in pLSI, in the variational inference procedure for LDA a vector of topic intensities has to be estimated for every document and in this sense there is not much difference between the two models and LDA just does kind of "smoothing" to avoid overfitting.

Therefore, in the LDA model, all documents in a corpus are drawn independent and identically distributed in accordance with the following generative process of a document (see the definitions of the Poisson and Dirichlet distributions in Appendix A.1) :

(0. Draw the document length $L \sim \text{Poisson}(\lambda)$.)

1. Draw the topic proportions $\boldsymbol{\theta} \sim \text{Dirichlet}(\mathbf{c})$.
2. For each of the L tokens $\mathbf{w}_\ell$:
   (a) Draw the topic $\mathbf{z}_\ell \sim \text{Mult}(1, \boldsymbol{\theta})$.
   (b) Draw the token $\mathbf{w}_\ell | \mathbf{z}_\ell \sim \text{Mult}\left(1, \mathbf{d}_{\mathbf{z}_\ell}\right)$.

Note that $\mathbf{d}_{\mathbf{z}_\ell}$ denotes the k-th topic where k corresponds to the non-zero entry of the vector $\mathbf{z}_\ell$. The number of topics K is assumed to be known and fixed. Although, the Poisson assumption on L is normally ignored in practice, this assumption will be important to show the connection [9] of LDA with the gamma-Poisson model (see Chapter 3). We concisely summarize this generative process as follows :

$$\begin{aligned} \big(L \sim \text{Poisson}(\lambda)\big), \\ \boldsymbol{\theta} \sim \text{Dirichlet}(\mathbf{c}), \\ \mathbf{z}_\ell | \boldsymbol{\theta} \sim \text{Mult}(1, \boldsymbol{\theta}), \\ \mathbf{w}_\ell | \mathbf{z}_\ell \sim \text{Mult}\left(1, \mathbf{d}_{\mathbf{z}_\ell}\right). \end{aligned} \tag{1.16}$$

We emphasize that it is formulated using tokens. This model (1.16) is illustrated with a plate diagram in Figure (1-5).

One can think of the latent variables $\mathbf{z}_\ell$ in the model 1.16 as auxiliary variables which were introduced for the convenience of inference. In fact, they can be marginalized out [Buntine, 2002, Buntine and Jakulin, 2004, 2006] to obtain an equivalent and more compact representation of the model. For completeness, we rigorously demonstrate the equivalence.

Marginalizing Out the Latent Variable z. The conditional distribution of the topic state vector $\mathbf{z}_\ell$ of the ℓ-th token in a document given the latent vector of the topic intensities $\boldsymbol{\theta}$ is

$$p(\mathbf{z}_\ell | \boldsymbol{\theta}) = \prod_{k=1}^{K} \theta_k^{z_{\ell k}}$$

9. The result is basically that the two models are equivalent given the Poisson assumption on L for the LDA model.

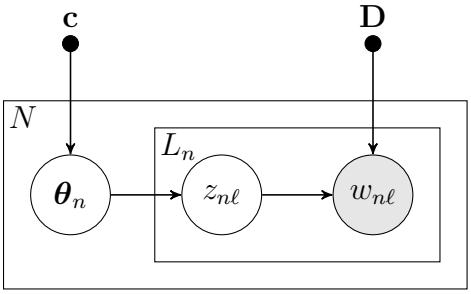


FIGURE 1-5 – The latent Dirichlet allocation (LDA) model.

and the conditional distribution of the ℓ-th token, given the respective latent topic state vector $\mathbf{z}_\ell$ and the latent topic intensities vector $\boldsymbol{\theta}$ has the form

$$p(\mathbf{w}_\ell|\mathbf{z}_\ell, \boldsymbol{\theta}) = \prod_{k=1}^{K}\prod_{m=1}^{M} d_{mk}^{w_{\ell m} z_{\ell k}}.$$

The joint distribution of all the tokens, $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L$, in a document and the respective latent states, $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_L$, take the form

$$p(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L, \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_L|\boldsymbol{\theta}) = \prod_{\ell=1}^{L} p(\mathbf{w}_\ell, \mathbf{z}_\ell|\boldsymbol{\theta}),$$

where $p(\mathbf{w}_\ell, \mathbf{z}_\ell|\boldsymbol{\theta}) = p(\mathbf{w}_\ell, \mathbf{z}_\ell|\boldsymbol{\theta})p(\mathbf{z}_\ell|\boldsymbol{\theta})$. Substituting the expressions from above to the marginal distribution of all the tokens we get

$$\begin{aligned}
p(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L|\boldsymbol{\theta}) &= \prod_{\ell=1}^{L}\sum_{\mathbf{z}_\ell\in\mathcal{Z}} p(\mathbf{w}_\ell, \mathbf{z}_\ell|\boldsymbol{\theta}) \\
&\overset{(a)}{=} \prod_{\ell=1}^{L}\sum_{\mathbf{z}_\ell\in\mathcal{Z}}\prod_{k=1}^{K}\prod_{m=1}^{M}[\theta_k d_{mk}]^{w_{\ell m} z_{\ell k}} \\
&= \prod_{\ell=1}^{L}\sum_{k=1}^{K}\prod_{m=1}^{M}[\theta_k d_{km}]^{w_{\ell m}} \\
&\overset{(b)}{=} \prod_{\ell=1}^{L}\prod_{m=1}^{M}\left([\mathbf{D}\boldsymbol{\theta}]_m\right)^{w_{\ell m}},
\end{aligned}$$

where $\mathcal{Z}$ stands for the set of all possible values of the one-hot encoded K-vector; in the equality (a) we used that $\prod_{k=1}^{K}\theta_k^{z_{\ell k}} = \prod_{k=1}^{K}(\theta_k^{z_{\ell k}})^{\sum_{m=1}^{M} w_{\ell m}} = \prod_{k=1}^{K}\prod_{m=1}^{M}\theta_k^{z_{\ell k} w_{\ell m}}$ since $\sum_{\ell=1}^{L} w_\ell = 1$; and in the equality (b) we used that we can exchange the summation and the product due to the one-hot encoding of the vector $\mathbf{w}_\ell$. Finally, we can

further rewrite this marginal distribution as

$$p(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L|\boldsymbol{\theta}) = \prod_{m=1}^{M} ([\mathbf{D}\boldsymbol{\theta}]_m)^{\sum_{\ell=1}^{L} w_{\ell m}}.$$

Recall that the count vector of a document $\mathbf{x}$ is equal to the sum of all tokens $\mathbf{w}_\ell$, i.e. $\mathbf{x} = \sum_{\ell=1}^{L} \mathbf{w}_\ell$. Let C_m denote the number of times the m-th word from the vocabulary appears in a document, then the vector $C = [C_1, C_2, \ldots, C_M]^\top$ represents the counts of all words in this document. Given $\boldsymbol{\theta}$, the event that the random variable $\mathbf{x}$ is equal to the vector C is then equal to the event that among the tokens $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L$ of this document there are C_m tokens corresponding to the m-th word. The second event can be represented in $L!/(C_1!C_2!\ldots C_M!)$ different ways, where L is the document length. Therefore, the count vector $\mathbf{x}$ is distributed in accordance with the multinomial distribution with the parameter $\mathbf{D}\boldsymbol{\theta}$ and L trials :

$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{L!}{x_1!x_2!\ldots x_M!} \prod_{m=1}^{M} [\mathbf{D}\boldsymbol{\theta}]_m^{x_m}.$$

This gives an equivalent formulation of the generative process (1.16) of a document under the LDA model :

$$\begin{aligned} \big(L &\sim \text{Poisson}(\lambda)\big), \\ \boldsymbol{\theta} &\sim \text{Dirichlet}(\mathbf{c}), \\ \mathbf{x}|\boldsymbol{\theta} &\sim \text{Mult}(L, \mathbf{D}\boldsymbol{\theta}). \end{aligned} \qquad \text{LDA model} \quad (1.17)$$

In this formulation, LDA is a special case of the *admixture* model (1.12), which justifies the place of the LDA model in the list of *latent linear models*. In formulation (3.2), the LDA model is also known under the names of *multinomial PCA* or the *Dirichlet-multinomial* model [Buntine, 2002, Buntine and Jakulin, 2004, 2006]. This formulation will prove useful in Chapter 3 for the formulation and motivation of the novel *discrete ICA* model.

1.2.5 Other Topic Models

LDA and pLSI do not model correlations between topics (e.g., that a topic about *geology* is more likely to appear together with a topic about *chemistry* than about *sport*). To address this, several classes of topic models were proposed, such as the *correlated topic model* [Blei and Lafferty, 2007] and *pachinko allocation model* [Li and McCallum, 2006]. However, the estimation and inference in these models is a more challenging task. Arabshahi and Anandkumar [2016] proposed recently *latent normalized infinitely divisible topic models* with an estimation method based on the method of moments, which goes along the lines of the methods developed in Chapter 3. Some models from this class allow modeling both positive and negative correlations among topics (see also Section 3.2). Moreover, we assumed that the number of topics K is fixed and known, which is rarely the case in practice. The *Bayesian nonparametric*

topic model [Teh et al., 2006] addresses this issue : the number of topics is determined by the collection during posterior inference. However, these and many other topic models are outside the scope of this thesis [for more details see, e.g., Blei, 2012].

1.3 Latent Linear Models for Multi-View Data

1.3.1 Probabilistic Canonical Correlation Analysis

Aligned or *multi-view* data often naturally arise in applications. For two views, such data is represented as two data sets such that each observation from one view is aligned with one observation in the other view and other way round. For instance, one view can consist of sentences in one language and the other can contain translations of these sentences into another language such that the same sentences in two different languages are aligned. Formally, one is given two data sets (finite samples) :

$$\mathbf{X}^{(1)} = \left\{\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}, \ldots, \mathbf{x}_N^{(1)}\right\} \subset \mathbb{R}^{M_1}, \quad \mathbf{X}^{(2)} = \left\{\mathbf{x}_1^{(2)}, \mathbf{x}_2^{(2)}, \ldots, \mathbf{x}_N^{(2)}\right\} \subset \mathbb{R}^{M_2}, \tag{1.18}$$

such that each pair $\left(\mathbf{x}_n^{(1)}, \mathbf{x}_n^{(2)}\right)$ is aligned. Naturally, the number of views can be larger than two with tuples of aligned observations. Such data is also known in the literature under the names of *dyadic*, *coupled*, or *paired* data.

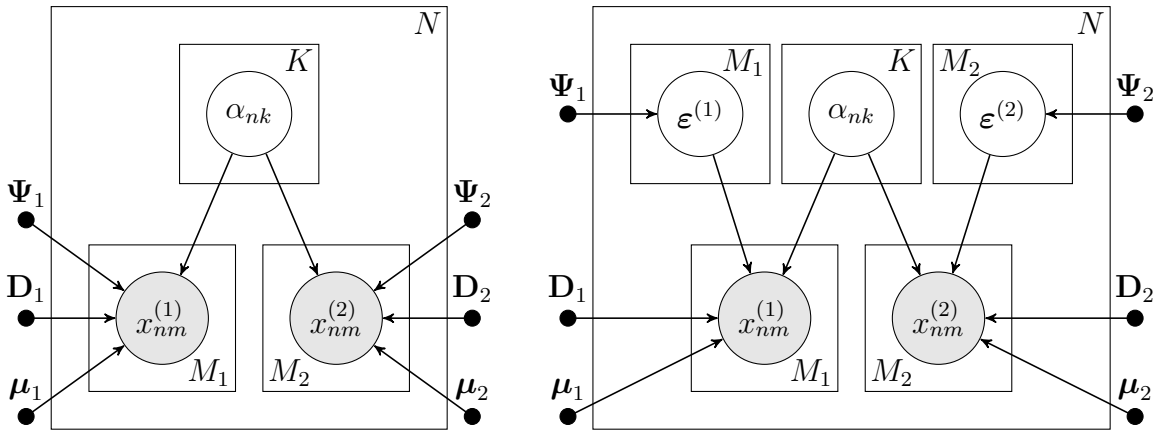
Canonical Correlation Analysis. Classical *canonical correlation analysis (CCA)* [Hotelling, 1936] aims to find a pair of linear transformations $\mathbf{D}_1 \in \mathbb{R}^{M_1 \times K}$ and $\mathbf{D}_2 \in \mathbb{R}^{M_2 \times K}$ of two observation vectors $\mathbf{x}^{(1)} \in \mathbb{R}^{M_1}$ and $\mathbf{x}^{(2)} \in \mathbb{R}^{M_2}$ each representing a data-view, such that each component of transformed variables in one data set, $\mathbf{D}_1^\top \mathbf{x}^{(1)}$, is correlated with a single component in the other set, $\mathbf{D}_2^\top \mathbf{x}^{(2)}$, i.e. the correlation is *maximized*. The columns of the matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ are called *canonical correlation directions*. Likewise classical PCA, the CCA solution boils down to solving a generalized SVD problem.

Probabilistic Canonical Correlation Analysis. The following *probabilistic* interpretation of CCA was proposed by Browne [1979], Bach and Jordan [2005], Klami et al. [2013]. Given that K sources are independent and identically distributed standard normal random variables, $\boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$, such probabilistic CCA model, which we will refer to as the *Gaussian CCA* model, is given by

$$\begin{aligned} \mathbf{x}^{(1)} | \boldsymbol{\alpha} &\sim \mathcal{N}(\boldsymbol{\mu}_1 + \mathbf{D}_1 \boldsymbol{\alpha}, \boldsymbol{\Psi}_1), \\ \mathbf{x}^{(2)} | \boldsymbol{\alpha} &\sim \mathcal{N}(\boldsymbol{\mu}_2 + \mathbf{D}_2 \boldsymbol{\alpha}, \boldsymbol{\Psi}_2), \end{aligned} \tag{1.19}$$

where the covariance matrices $\boldsymbol{\Psi}_1 \in \mathbb{R}^{M_1 \times M_1}$ and $\boldsymbol{\Psi}_2 \in \mathbb{R}^{M_2 \times M_2}$ are positive semi-definite. The maximum likelihood estimators of the parameters $\mathbf{D}_1$ and $\mathbf{D}_2$ coincide with canonical correlation directions, up to permutation, scaling, and left-multiplication by *any invertible* matrix. Therefore, likewise factor analysis and probabilistic PCA, probabilistic CCA is *unidentifiable*.

By analogy with factor analysis, the Gaussian CCA model (1.19) can be equivalently



(a) A PCCA plate diagram. (b) A PCCA generative process diagram.

FIGURE 1-6 – The probabilistic CCA model.

represented through the following generative process

$$\begin{aligned} \mathbf{x}^{(1)} &= \boldsymbol{\mu}_1 + \mathbf{D}_1 \boldsymbol{\alpha} + \boldsymbol{\varepsilon}^{(1)}, \\ \mathbf{x}^{(2)} &= \boldsymbol{\mu}_2 + \mathbf{D}_2 \boldsymbol{\alpha} + \boldsymbol{\varepsilon}^{(2)}, \end{aligned} \tag{1.20}$$

where the additive noise vectors are multivariate normal random variables :

$$\boldsymbol{\varepsilon}^{(1)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}_1), \qquad \boldsymbol{\varepsilon}^{(2)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}_2), \tag{1.21}$$

and the following *independence* assumptions are made

$$\begin{aligned} &\alpha_1, \ldots, \alpha_K \text{ are mutually independent,} \\ &\boldsymbol{\alpha} \perp\!\!\!\perp \boldsymbol{\varepsilon}^{(1)}, \boldsymbol{\varepsilon}^{(2)} \quad \text{and} \quad \boldsymbol{\varepsilon}^{(1)} \perp\!\!\!\perp \boldsymbol{\varepsilon}^{(2)}. \end{aligned} \tag{1.22}$$

Therefore, Gaussian CCA is nearly an extension of factor analysis to two views. The difference is that the covariance matrices of the noise, $\boldsymbol{\Psi}_1$ and $\boldsymbol{\Psi}_2$, are not restricted to be diagonal and the view-specific noise may be arbitrary correlated. The only requirement is that there are no correlations across the views. More specifically, to see how Gaussian CCA is related to factor analysis, one can stack the view specific vectors (matrices) into a single vector (matrix) :

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \boldsymbol{\varepsilon}^{(1)} \\ \boldsymbol{\varepsilon}^{(2)} \end{pmatrix}, \tag{1.23}$$

which leads exactly to the generative model of factor analysis (1.7) with a single difference of the assumptions on the additive noise. Indeed, in factor analysis the additive noise is zero-mean Gaussian variable with the diagonal covariance matrix $\boldsymbol{\Psi}$, while in Gaussian CCA, the covariance of the zero-mean Gaussian additive noise has

the block diagonal structure :

$$\boldsymbol{\Psi} = \begin{pmatrix} \boldsymbol{\Psi}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_2 \end{pmatrix}, \tag{1.24}$$

Using such *stacking trick*, one can see that the marginal distribution of the observations under the Gaussian CCA model is the same as the one for factor analysis (1.6) with the covariance matrix of the noise in the form (1.24). Extension to more than two views are possible [see, e.g., Kettenring, 1971, Bach and Jordan, 2002], but are not considered in this thesis.

By relaxing the Gaussianity assumption, ICA is transformed into an identifiable version of factor analysis. Similarly, by relaxing the Gaussianity assumption on the independent latent sources in Gaussian CCA, we obtain an identifiable version of CCA (see Chapter 4). Some other multi-view models are outlined in Section 4.2.

1.4 Overcomplete Latent Linear Models

It is common to distinguish the undercomplete and overcomplete cases of latent linear models. The *undercomplete* case assumes that there are less latent variables than observations in a model, i.e. $K < M$ (in the ICA literature, this case is also known as *overdetermined*). On the contrary, the *overcomplete* case assumes that there are more latent variables than observations in a model, i.e. $K > M$ (in the ICA literature, this case is also known as *underdetermined*). The equality case, $M = K$ corresponds to the *complete* or *determined* case if the linear transformation matrix $\mathbf{D}$ is full rank.

In the undercomplete case, the problem $\mathbf{D}\boldsymbol{\alpha}$ is well-posed if the matrix $\mathbf{D}$ has full column rank in a sense that the latent sources and the matrix $\mathbf{D}$ can be identified up to permutation and scaling. This is an easier problem and such setting is assumed all over this thesis.

On the contrary, in the overcomplete case at least some columns of the linear transformation matrix are linearly dependent and the recovery problem of the matrix $\mathbf{D}$ is a more difficult problem. Nevertheless, (essentially) unique recovery of the matrix $\mathbf{D}$ is still possible without any additional assumptions [Lee et al., 1999, Lewicki and Sejnowski, 2000, Comon, 2004, De Lathauwer et al., 2007, De Lathauwer and Castaing, 2008, Comon, 1998] or in the topic modeling case [Anandkumar et al., 2015c]. However, recovery of the latent sources $\boldsymbol{\alpha}$ requires additional assumptions in such case.

Chapitre 2

Tensors and Estimation in Latent Linear Models

Abstract

In accordance with the *method of moments*, parameters of a model are estimated by matching the *population* moments (or other statistics such as cumulants) of this model with respective *sample estimates* of this moments (or other statistics). When the method of moments is used for the estimation in latent linear models, higher-order statistics, which happen to be *tensors*, have often to be taken into account. Therefore, we first provide an overview of tensors and their decompositions in this chapter, with the emphasis on the *canonical polyadic (CP)* decomposition (in Section 2.2), and recall different higher-order statistics (moments and cumulants ; in Section 2.2). We then connect this CP decomposition of tensors with higher-order statistics of latent linear models, using as example such models as *independent component analysis* (in Section 2.2.2) and *latent Dirichlet allocation* (in Section 2.2.3), and then make an overview of algorithms for the CP decomposition (in Section 2.3). This eventually allows us to perform the estimation in some latent linear models.

2.1 Tensors, Higher Order Statistics, and CPD

2.1.1 Tensors

Let $\mathbb{V}^{(1)}, \mathbb{V}^{(2)}, \ldots, \mathbb{V}^{(S)}$ be S vector spaces of dimension $M_1, M_2, \ldots, M_S$, respectively. An order-S *tensor* $\boldsymbol{\mathcal{T}}$ is an element of the vector space obtained as a tensor product of these S vector spaces, $\mathbb{V} := \mathbb{V}^{(1)} \otimes \mathbb{V}^{(2)} \otimes \cdots \otimes \mathbb{V}^{(S)}$, where $\otimes$ stands for the tensor (outer) product. The *order* S of a tensor is the number of dimensions, also known as *ways* or *modes*. Thus, an order-1 tensor is a vector, an order-2 tensor is a matrix, and order-3 or higher tensors are referred to as *higher-order* tensors. For a detailed overview of tensors see, e.g., McCullagh [1987], Comon [2002], Kolda and Bader [2009], Comon [2009], Landsberg [2012], Comon [2014], and references therein.

Let $\mathbf{E}^{(s)} := \left\{\mathbf{e}_1^{(s)}, \mathbf{e}_2^{(s)}, \ldots, \mathbf{e}_{M_s}^{(s)}\right\}$ denote a basis of the vector space $\mathbb{V}^{(s)}$. Then the coordinates $\mathcal{T}_{m_1 m_2 \ldots m_S}$ of any tensor $\boldsymbol{\mathcal{T}} \in \mathbb{V}$ correspond to

$$\boldsymbol{\mathcal{T}} = \sum_{s=1}^{S} \sum_{m_s=1}^{M_s} \mathcal{T}_{m_1 m_2 \ldots m_S}\, \mathbf{e}_{m_1}^{(1)} \otimes \mathbf{e}_{m_2}^{(2)} \otimes \cdots \otimes \mathbf{e}_{m_S}^{(S)}, \tag{2.1}$$

where the summation is performed along all modes. Unless otherwise specified, we restrict ourselves to the vector spaces over the field $\mathbb{R}$ of real numbers, in which case we write $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_S}$. Moreover, a basis $\mathbf{E}^{(s)}$ is often chosen to be the canonical basis of $\mathbb{R}^{M_s}$, i.e. the vector $\mathbf{e}_m^{(s)}$ is the m-th column of the $M_s \times M_s$ identity matrix, for all $m \in [M_s]$. In this case, a tensor $\boldsymbol{\mathcal{T}}$ defined in (2.1) can be seen as an S-way array with the $(m_1, m_2, \ldots, m_S)$-th element equal to $\mathcal{T}_{m_1 m_2 \ldots m_S}$. The latter representation, however, is basis dependent.

The *s-mode product* of a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_S}$ with a matrix $\mathbf{A} \in \mathbb{R}^{K \times M_s}$ is denoted as $\boldsymbol{\mathcal{T}} \times_s \mathbf{A}$ and is a tensor of size $M_1 \times \cdots \times M_{s-1} \times K \times M_{s+1} \times \cdots \times M_S$ with the $(m_1, \ldots, m_{s-1}, k, m_{s+1}, \ldots, m_S)$-th element equal to

$$[\boldsymbol{\mathcal{T}} \times_s \mathbf{A}]_{m_1 \ldots m_{s-1}\, k\, m_{s+1} \ldots m_S} := \sum_{m_s=1}^{M_s} \mathcal{T}_{m_1 \ldots m_{s-1}\, m_s\, m_{s+1} \ldots m_S}\, A_{k m_s}. \tag{2.2}$$

The s-mode product is directly related to a change of basis and the so-called multilinearity property. Indeed, let $\mathbf{e}_m^{(s)} = \mathbf{A}^{(s)} \mathbf{e}'^{(s)}_m$, for $m = 1, 2, \ldots, M_s$, denote a change of basis in $\mathbb{R}^{M_s}$ for some given matrices $\mathbf{A}^{(s)} \in \mathbb{R}^{M_s \times M_s}$, for all $s \in [S]$. Then the new coordinates $\mathcal{T}'_{k_1 k_2 \ldots k_S}$ of the tensor $\boldsymbol{\mathcal{T}}$ are expressed as a function of the original ones [Comon et al., 2009a] as

$$\mathcal{T}'_{k_1 k_2 \ldots k_S} = \sum_{s=1}^{S} \sum_{m_s=1}^{M_s} A^{(1)}_{k_1 m_1} A^{(2)}_{k_2 m_2} \ldots A^{(S)}_{k_S m_S}\, \mathcal{T}_{m_1 m_2 \ldots m_S}. \tag{2.3}$$

This, essentially, can be denoted as $\boldsymbol{\mathcal{T}}' = \boldsymbol{\mathcal{T}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \cdots \times_S \mathbf{A}^{(S)}$. The property (2.3) of tensors is referred to as *multilinearity* and, in particular, it allows

to treat higher-order statistics (moments and cumulants) as tensors (see Property 2 in Section 2.2).

Let us also define several special types of tensors. A tensor $\boldsymbol{\mathcal{T}}$ is called *diagonal* if the only non-zero elements are the ones on the diagonal, i.e. $\mathcal{T}_{m_1 m_2 \dots m_S} \neq 0$ only if $m_1 = m_2 = \cdots = m_S$. A tensor $\boldsymbol{\mathcal{T}}$ is called *cubical* if the dimensions along all its modes coincide, i.e. $M_1 = M_2 = \cdots = M_S = M$. A cubical tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{M \times M \times \cdots \times M}$ is called *symmetric* if it remains unchanged under any permutation $\boldsymbol{\pi}$ of its indices, i.e.

$$\mathcal{T}_{\boldsymbol{\pi}(m_1)\boldsymbol{\pi}(m_2)\dots\boldsymbol{\pi}(m_S)} = \mathcal{T}_{m_1 m_2 \dots m_S}. \tag{2.4}$$

Otherwise, a tensor $\boldsymbol{\mathcal{T}}$ is called *non-symmetric*.

A *fiber* of a tensor is a vector obtained by fixing all but one indices in this tensor. Fibers are extensions of the matrix rows and columns to tensors. A fiber is called *mode-s fiber*, if the unfixed index is along the s-th dimension (or mode). A *slice* is a two-dimensional section of a tensor obtained by fixing all but two indices. *Unfolding* (a.k.a. *matricization*) refers to the process of rearranging the elements of a tensor into a matrix. *Vectorization* is the process of rearranging the elements of a tensor into a vector. For both unfolding and vectorization, the ordering of the elements is not important as long as it is consistent.

2.1.2 The Canonical Polyadic Decomposition

It turns out that working with tensors is much more challenging than working with matrices. To emphasize the difference, let us first briefly recall an important result from linear algebra—the singular value decomposition—and its possible extensions to tensors.

The Singular Value Decomposition. Matrices — order-2 tensors — are well studied and their properties are well understood [see, e.g., Horn and Johnson, 2013]. One noticeable property of matrices is the *singular value decomposition (SVD)*. It says that given a real or complex $M_1 \times M_2$ matrix $\mathbf{A}$ of rank R, there exist unitary matrices $\mathbf{U}$ and $\mathbf{V}$, of size $M_1 \times M_1$ and $M_2 \times M_2$ respectively, and values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_R > 0 = \sigma_{R+1} = \cdots = \sigma_F$, where $F = \min[M_1, M_2]$, such that

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*, \tag{2.5}$$

where $\boldsymbol{\Sigma}$ is an $M_1 \times M_2$ diagonal matrix with the diagonal elements equal to $\sigma_1, \dots, \sigma_F$, and $\mathbf{V}^*$ stands for the conjugate transpose of $\mathbf{V}$. The columns of the matrix $\mathbf{U}$ are called the *left singular vectors*, the columns of the matrix $\mathbf{V}$ are called the *right singular vectors*, and the values σ_m are referred to as the *singular values*.

The SVD formulation (2.5) can be equivalently rewritten as the sum of rank-1 terms :

$$\mathbf{A} = \sum_{r=1}^{R} \sigma_r \, \mathbf{u}_r \mathbf{v}_r^*, \tag{2.6}$$

where the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_R$ and $\mathbf{v}_1, \ldots, \mathbf{v}_R$ are the first R left and right singular vectors, respectively, and the superscript $(\cdot)^*$ stands for the conjugate transpose. Here the terms $\mathbf{u}_r\mathbf{v}_r^*$ are the *rank-1 matrices*. Any rank-R matrix admits the singular value decomposition and it is unique (up to scaling by -1) given all singular values are different.

An extension of the SVD to tensors is not straightforward. Consider the following decomposition of an order-3 tensor introduced by Tucker [1966]

$$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{G}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}, \tag{2.7}$$

where $\boldsymbol{\mathcal{G}}$ is an order-3 tensor called the *core tensor* and the matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ are called the *factor matrices*. The decomposition problem would be to estimate the core tensor $\boldsymbol{\mathcal{G}}$ and factor matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ given the tensor $\boldsymbol{\mathcal{T}}$. To obtain an equivalent of the SVD for tensors, we would have to make two assumptions in (2.7) : (a) the tensor $\boldsymbol{\mathcal{G}}$ is diagonal and (b) the matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ are orthonormal. In general, if the number of free parameters in the right-hand side of (2.7) is smaller than the number of equations, then one can not solve the problem. This happens to be the case for an order-3 tensor under the mentioned diagonality and orthogonality assumptions. Therefore, it is impossible to define an extension of the SVD to higher-order tensors without relaxing some assumptions.

By relaxing one or the other assumption, two main tensor equivalents of the SVD for tensors can be found in the literature. One of these decompositions, a tensor equivalent of the formulation in (2.5), is the *Tucker decomposition* [Tucker, 1966] where the tensor $\boldsymbol{\mathcal{G}}$ in (2.7) is allowed to have non-zero non-diagonal entries, but the orthonormality constraint is preserved. Note that in this case the dimensions of the core tensor $\boldsymbol{\mathcal{G}}$ are normally significantly smaller than the dimensions of the tensor $\boldsymbol{\mathcal{T}}$. In the following, we use equation (2.7) to refer to the Tucker decomposition. Another decomposition, a tensor equivalent of the formulation in (2.6), is the so-called *canonical polyadic decomposition*, or the *CP decomposition*, where the tensor $\boldsymbol{\mathcal{G}}$ is constrained to be diagonal, but the orthonormality constraint on the factor matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ is relaxed, allowing, in particular, factor matrices with more columns than rows. The CP decomposition is of central importance in this thesis and will be discussed in detail in this and other sections.

Not only the mentioned extensions of the SVD for tensors differ, but computing these decompositions is also a challenging task. In fact, most of the tensor problems are NP-hard [Hillar and Lim, 2013].

The Canonical Polyadic Decomposition. An order-S tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_S}$ is called a *rank-1 tensor* if it can be written as the tensor product of S vectors, i.e. $\boldsymbol{\mathcal{T}} = \mathbf{u}^{(1)} \otimes \mathbf{u}^{(2)} \otimes \cdots \otimes \mathbf{u}^{(S)}$, where $\mathbf{u}^{(s)} \in \mathbb{R}^{M_s}$ for $s \in [S]$. Any tensor $\boldsymbol{\mathcal{T}}$ admits a decomposition into a sum of rank-1 tensors as follows

$$\boldsymbol{\mathcal{T}} = \sum_{j=1}^{F} \mathbf{u}_j^{(1)} \otimes \mathbf{u}_j^{(2)} \otimes \cdots \otimes \mathbf{u}_j^{(S)}, \tag{2.8}$$

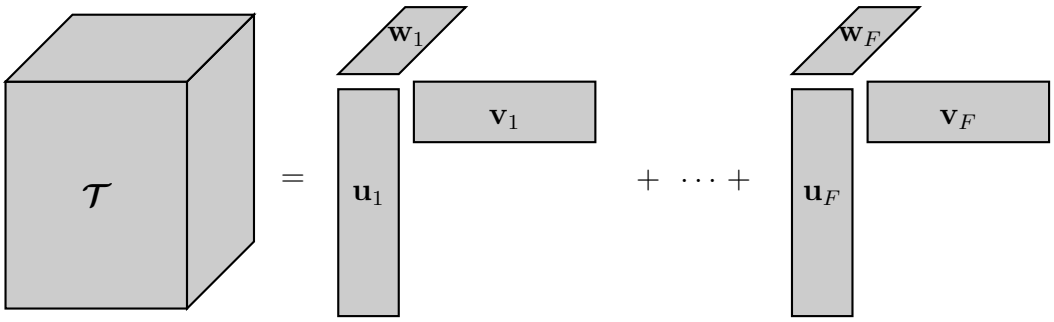


FIGURE 2-1 – A canonical polyadic decomposition of an order-3 tensor $\mathcal{T}$.

with potentially very large F, which is called the *canonical polyadic decomposition* or *CP decomposition* (see an illustration on Figure 2-1). Note that for $S = 3$ this is equivalent to the decomposition in (2.7) with the diagonal core tensor $\mathcal{G}$ and relaxed orthonormality assumption on the factor matrices (with $\mathbf{u} := \mathbf{u}^{(1)}$, $\mathbf{v} := \mathbf{u}^{(2)}$, and $\mathbf{w} := \mathbf{u}^{(3)}$), since we can eliminate the scalar factors $\mathcal{G}_{mmm}$ by respectively rescaling the factor vectors. For $s \in [S]$, the order-s *factor matrix* $\mathbf{U}^{(s)} \in \mathbb{R}^{M_s \times F}$ of the CP decomposition is obtained by stacking the *factors* $\mathbf{u}_1^{(s)}$, $\mathbf{u}_2^{(s)}$, ..., $\mathbf{u}_F^{(s)}$ columnwise into a matrix, i.e. $\mathbf{U}^{(s)} := \left[\mathbf{u}_1^{(s)}\ \mathbf{u}_2^{(s)}\ \ldots\ \mathbf{u}_F^{(s)}\right]$. We emphasize one more time that the matrices $\mathbf{U}^{(s)}$ are not assumed to be orthonormal. Moreover, the number of factors F does not have to be equal to any of dimensions and can, in particular, be larger than any of M_1, M_2, ..., M_S.

It is worth noting that the naming convention between the CP decomposition and factor analysis is confusing (see Section 1.1.2). The factors of the CP decomposition correspond to the factor loadings in factor analysis. The factors in factor analysis refer to the latent variables and, therefore, mean something different from the factors in the CP decomposition. Overall, the factor matrix of the CP decomposition is an equivalent of the factor loading matrix in factor analysis (see also a probabilistic interpretation of the non-negative CP decomposition below).

The canonical polyadic decomposition was originally introduced by Hitchcock [1927a,b]. The name *canonical decomposition* was given by Carroll and Chang [1970] and the name *Parafac* was introduced by Harshman [1970], Harshman and Lundy [1994], both in the psychometrics literature. For a detailed introduction to the CP decomposition and a literature overview see, e.g., Kolda and Bader [2009], Comon et al. [2009a].

The coordinate representation of the CP decomposition is as follows. Let $u_{m_s r}^{(s)}$ denote the m_s-th coordinates of the vector $\mathbf{u}_r^{(s)}$ in a basis $\mathbf{E}^{(s)}$, then the CP decomposi-

tion (2.8) can be rewritten as

$$\begin{aligned}\boldsymbol{\mathcal{T}} &= \sum_{j=1}^{F} \left(\sum_{m_1=1}^{M_1} u_{m_1 j} \mathbf{e}^{(1)}_{m_1} \right) \otimes \left(\sum_{m_2=1}^{M_2} u_{m_2 j} \mathbf{e}^{(2)}_{m_2} \right) \otimes \cdots \otimes \left(\sum_{m_S=1}^{M_S} u_{m_S j} \mathbf{e}^{(S)}_{m_S} \right) \\ &= \sum_{s=1}^{S} \sum_{m_s=1}^{M_s} \left(\sum_{j=1}^{F} u^{(1)}_{m_1 j} u^{(2)}_{m_2 j} \dots u^{(S)}_{m_S j} \right) \mathbf{e}^{(1)}_{m_1} \otimes \mathbf{e}^{(2)}_{m_2} \otimes \cdots \otimes \mathbf{e}^{(S)}_{m_S},\end{aligned} \tag{2.9}$$

which shows that the coordinates are related as $\mathcal{T}_{m_1 m_2 \dots m_S} = \sum_{j=1}^{F} u^{(1)}_{m_1 j} u^{(2)}_{m_2 j} \dots u^{(S)}_{m_S j}$. However, by analogy with the coordinate representation of a tensor in (2.1), the representation in (2.9) is basis dependent as opposed to the representation in (2.8).

Symmetric CP Decomposition. It is common to distinguish special types of CP decompositions. One of them is the *symmetric CP decomposition* [Comon et al., 2008] : for a symmetric tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{M \times M \times \cdots \times M}$ the symmetric CP decomposition is defined as

$$\boldsymbol{\mathcal{T}} = \sum_{j=1}^{F} \lambda_j \, \mathbf{u}_j \otimes \mathbf{u}_j \otimes \cdots \otimes \mathbf{u}_j. \tag{2.10}$$

Likewise the symmetric case, the matrix $\mathbf{U} := [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_F] \in \mathbb{R}^{M \times F}$ is not assumed to be orthogonal and F can be larger than M. As opposed to the non-symmetric CP decomposition (2.8), introduction of the scalar factors λ_r is essential. Indeed, in the non-symmetric decomposition (2.8), we can simply rescale the factors $\mathbf{u}^{(1)}_j$, $\mathbf{u}^{(2)}_j$, …, $\mathbf{u}^{(S)}_j$ by λ_j without changing the result. However, in the symmetric case such rescaling is not always possible : take for example an even order tensor and $\lambda_j = -1$.

Nonnegative CP Decomposition. Another type of the CP decomposition is the *nonnegative canonical polyadic (NCP) decomposition* of a nonnegative tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_S}$ which is defined as

$$\boldsymbol{\mathcal{T}} = \sum_{j=1}^{F} \lambda_j \, \mathbf{u}^{(1)}_j \otimes \mathbf{u}^{(2)}_j \otimes \cdots \otimes \mathbf{u}^{(S)}_j, \tag{2.11}$$

where all factors are nonnegative, i.e. $\mathbf{u}^{(s)}_j \geq 0$ and $\lambda_j \geq 0$, for all $j \in [F]$ and $s \in [S]$ [Carroll et al., 1989, Krijnen and Ten Berge, 1991, Paatero, 1997, Shashua and Hazan, 2005, Lim and Comon, 2009]. Note that, without loss of generality, the vectors $\mathbf{u}^{(s)}_j$ can be restricted to belong to the probability simplex, i.e. $\mathbf{u}^{(s)}_j \in \boldsymbol{\Delta}_{M_s}$, for all $s \in [S]$. The nonnegative CP decomposition (2.11) is a straightforward extension of nonnegative matrix factorization [Paatero and Tapper, 1994, Lee and Seung, 1999] to tensors. Moreover, since NMF is equivalent [1] to probabilistic latent semantic indexing (pLSI ; see Section 1.2.3) [Gaussier and Goutte, 2005, Ding et al., 2006, 2008], and since the NCP decomposition is an extension of NMF to tensors, the nonnegative CP

1. The KL-divergence based NMF objective is equivalent to the maximum likelihood objective of pLSI.

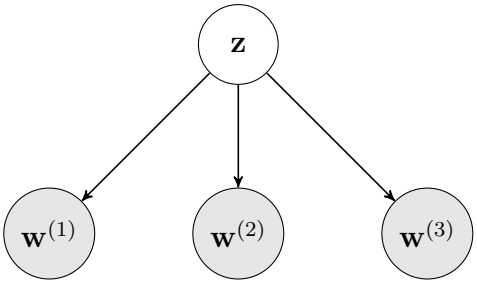


FIGURE 2-2 – The probabilistic interpretation of NCP (multi-view pLSI).

decomposition (2.11) has a probabilistic interpretation as extension of pLSI to more than two random variables [Lim and Comon, 2009]. Importantly, the best low-rank approximation problem of the nonnegative CP decomposition is *well-posed* [De Silva and Lim, 2008, Comon et al., 2008, Lim and Comon, 2009], i.e. a solution always exists. Some CP uniqueness results also extend to nonnegative tensors [Qi et al., 2016].

Probabilistic Interpretation of the NCP Decomposition. We discussed pLSI in Section 1.2.3, where we mentioned that pLSI admits straightforward extension to the multi-view case via its symmetric parametrization (see also Figure 1-4a). Indeed, one just needs to add more observed variables (views), which are conditionally independent of other variables given the latent variable (i.e., the naïve Bayes hypothesis) Lim and Comon [2009]. We illustrate such extension to the case of three observed variables in Figure 2-2.

Extending the pLSI generative model to this case of three views, we directly obtain the following marginal distribution

$$p\left(\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \mathbf{w}^{(3)}\right) = \sum_{\mathbf{z}} p(\mathbf{z}) p\left(\mathbf{w}^{(1)}|\mathbf{z}\right) p\left(\mathbf{w}^{(2)}|\mathbf{z}\right) p\left(\mathbf{w}^{(3)}|\mathbf{z}\right),$$

where $p(\mathbf{z})$ and $p\left(\mathbf{w}^{(j)}|\mathbf{z}\right)$, for $j = 1, 2, 3$, are some discrete distributions. Therefore, this probability mass function can be rewritten as a non-negative CP decomposition, provided we "store" the distributions $p(\mathbf{w}^{(j)}|\mathbf{z}_k)$, for all j and k, in matrices $\mathbf{D}^{(j)} \in \mathbb{R}^{M_j \times K}$. Likewise the minimization of the KL divergence in NMF is equivalent to the maximization of the maximum likelihood in pLSI, this equivalence of NCP and extended to three views pLSI is the same : although the formulations of the problems are equivalent, different algorithms can be used for solving the factorization or estimation and inference problems. This probabilistic interpretation of NCP clearly has straightforward extensions to more than three views (to tensors of order higher than three).

2.1.3 Tensor Rank and Low-Rank Approximation

Tensor Rank. The *tensor rank* is defined as the minimal number R of rank-1 terms in the CP decomposition (2.8) of a tensor [Hitchcock, 1927a, Kruskal, 1977]. The tensor

rank always exists and is well defined [see, e.g., Comon et al., 2009a]. Although the definition of the tensor rank is an exact extension of the definition of the matrix rank, the properties are quite different. For example, the rank of a real valued tensor may actually be different over $\mathbb{R}$ and $\mathbb{C}$ [see, e.g., Kruskal, 1989, Ten Berge, 1991, Kolda and Bader, 2009]. Moreover, there does not exist a straightforward algorithm to determine the rank of a specific given tensor; in fact, the problem is NP-hard [Håstad, 1990, Hillar and Lim, 2013]. Due to this complexity of the tensor rank, other types of tensor ranks are also introduced in the literature.

The *symmetric tensor rank* is defined as the minimal number R_{sym} of rank-1 terms in the symmetric CP decomposition (2.10) of a tensor [Comon et al., 2008]. It is clear that $R_{sym} \geq R$ for any symmetric tensor $\boldsymbol{\mathcal{T}}$, since any constraint on decomposable tensor may only increase the rank. However, it has been conjectured by Comon et al. [2008] that the rank and the symmetric rank are always equal, but this has not yet been proved in the general case.

The *nonnegative tensor rank* is defined as the minimal number R_n of nonnegative rank-1 terms in the nonnegative CP decomposition (2.11) of a tensor [Lim and Comon, 2009]. The nonnegative rank is generally strictly larger than the rank. This is already the case for matrices, order-2 tensors [Comon, 2014].

A natural generalization of the matrix row and column rank to tensors is the mode-s rank of a tensor. The *mode-s rank* of a tensor $\boldsymbol{\mathcal{T}}$ is the dimension of the space spanned by the mode-s fibers [2] of this tensor. We denote the mode-s rank of a tensor $\boldsymbol{\mathcal{T}}$ as R_s. For matrices, the row rank is equal to the column rank and is equal to the number of the rank-1 terms in the SVD, where the latter is the equivalent of the tensor rank R. For tensors, however, the numbers R, R_1, R_2, ..., and R_S can all be different. Therefore, the S-tuple $(R_1, R_2, \ldots, R_S)$ of all the mode-s ranks of a tensor, which is called the *multirank* of this tensor, is of interest as well.

Generic (resp. *typical*) *ranks* are the ranks that one encounters with probability one (resp. nonzero probability), when their entries are drawn independently according to some continuous distribution [Lickteig, 1985, Comon, 2002, Comon and Ten Berge, 2008, Comon et al., 2009b, Comon, 2014]. For matrices, the generic and typical rank are equal to $\min(M_1, M_2)$. For a higher-order tensor, a generic rank over the real numbers does not necessary exist (although, over the complex numbers it always exists) and both generic and typical rank can be strictly greater than $\min(R_1, R_2, \ldots, R_S)$ and, in general, are hard to compute. This is another striking difference of the tensor rank from the matrix rank. The generic and typical rank for some order-3 tensors of small dimensions have been determined by Comon and Ten Berge [2008], Comon et al. [2009b]. The probabilities for the typical rank of some tensors have been studied with numerical simulations. For example, the rank of a $2 \times 2 \times 2$ tensor with elements drawn as independent and identically distributed standard normal random variables $\mathcal{N}(0, 1)$ is equal to 2 with probability 0.25π, and to 3 with probability 0.75π

2. Recall that an mode-s *fiber* of a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{M_1 \times \cdots \times M_S}$ is the M_s-dimensional vector obtained from $\boldsymbol{\mathcal{T}}$ by changing the index $m_s \in [M_s]$ while keeping all other indices fixed.

[Kruskal, 1989]. For this particular setting, the exact values of the probabilities were also derived by Bergqvist [2013].

Finally, another useful notion of tensor rank is the *border rank* which is defined as the minimum number of rank-1 terms that are sufficient to approximate the given tensor with arbitrary small nonzero error. This notion of border rank arises when considering the best low-rank approximation of tensors (see below) and provides an explanation of the ill-posedness of this problem. The idea behind this problem is that a series of rank-S tensors can converge to a rank-$(S+1)$ tensor, i.e. the space of rank-S tensors is not closed [Kruskal et al., 1989]. Once more, the rank and border rank always coincide for matrices, but they do not generally coincide for higher-order tensors. An exception here is the nonnegative CP decomposition, where the nonnegative border rank coincides with the nonnegative rank [Lim and Comon, 2009] and, therefore, the low-rank approximation problem (see below) is well-posed in this case.

Tensor Low-rank Approximation. In practice, matrices or tensors are often expected to have *approximate* low rank structure. For example, as we will see in Sections 2.2.2 and 2.2.3, population higher-order statistics of some models are tensors with a low-rank CP structure (given the undercomplete setting). However, their sample estimators are perturbed by noise, which poses the question of the approximation rather than exact estimation. The low-rank approximation is of special interest because, as we will see later, having high rank in the approximation often does not guarantee the uniqueness of the decomposition [see, e.g., Comon et al., 2009a].

The best low-rank approximation problem for matrices is well-posed and the solution can be easily obtained through the SVD. Indeed, recall that the best low-rank approximation of a rank-R matrix $\mathbf{A} \in \mathbb{R}^{M_1 \times M_2}$ by a matrix $\hat{\mathbf{A}} \in \mathbb{R}^{M_1 \times M_2}$ of the rank $K < R$ is defined as

$$\hat{\mathbf{A}} = \underset{\mathbf{B} \in \mathbb{R}^{M_1 \times M_2}}{\arg\min} \|\mathbf{A} - \mathbf{B}\|_F^2 \quad \text{subject to} \quad \operatorname{rank}[\mathbf{B}] \leq K. \tag{2.12}$$

By the renowned Eckart-Young theorem [Eckart and Young, 1936], the solution to this problem is directly obtained from the *truncated singular value decomposition*, that is $\hat{\mathbf{A}} = \sum_{k=1}^{K} \sigma_k \ \mathbf{u}_k \otimes \mathbf{v}_k$, where $\sigma_1, \ldots, \sigma_K$ are the first K largest singular values of the matrix $\mathbf{A}$ and $\mathbf{u}_k$ and $\mathbf{v}_k$, for $k \in [K]$, are the respective left and right singular vectors.

Similarly to the matrix case, given a fixed value $K < R = \operatorname{rank}[\boldsymbol{\mathcal{T}}]$, one can be interested in approximating this tensor $\boldsymbol{\mathcal{T}}$ by a low rank-K tensor. A natural way to define this problem for a given $\boldsymbol{\mathcal{T}}$ is as follows :

$$\inf \left\| \boldsymbol{\mathcal{T}} - \sum_{k=1}^{K} \mathbf{u}_k^{(1)} \otimes \mathbf{u}_k^{(2)} \otimes \cdots \otimes \mathbf{u}_k^{(S)} \right\|^2, \tag{2.13}$$

where the *(Frobenius) norm* $\|\boldsymbol{\mathcal{T}}\|$ of a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_S}$ is defined as

$$\|\boldsymbol{\mathcal{T}}\|^2 = \sum_{s=1}^{S} \sum_{m_s=1}^{M_s} \mathcal{T}^2_{m_1 m_2 \ldots m_S}, \tag{2.14}$$

which is essentially an equivalent of the matrix Frobenius (or Euclidean) norm. [3] The major difficulty with the problem (2.13) is that there exists a tensor $\boldsymbol{\mathcal{T}}$ such that this infimum may not be attained. This failure can occur with positive probability over a wide range of dimensions, orders, ranks, and for any continuous measure of proximity (this includes all norms and Bregman divergences) used instead of the Frobenius norm [Kruskal et al., 1989, De Silva and Lim, 2008, Comon et al., 2009a]. Moreover, such failures can occur with positive probability and in some cases with certainty, i.e. where the infimum in (2.13) is never attained. From the algorithmic point of view, the ill-posedness of the low-rank tensor decomposition is a serious issue, since what can we compute when a problem does not have a solution ? In practice, this ill-conditioning comes out in a form of rank-1 summands that grew unbounded in magnitude but their sum remains bounded [Bini et al., 1979, 1980, Paatero, 2000, Kruskal et al., 1989, Comon et al., 2009a, Lim and Comon, 2009]. This phenomenon is known in the literature under the name of *CP degeneracy*.

However, there are several exceptional cases, where the low-rank tensor approximation problem is known to be well-posed. The first case is the case where $R = 1$ which is the best rank-1 tensor approximation problem [De Lathauwer et al., 2000]. This is also true in the case of symmetric tensors : the best symmetric rank-1 tensor approximation problem is well posed [Kofidis and Regalia, 2002]. Note that the best rank-1 approximation of a symmetric tensor is symmetric [Comon et al., 2008]. The second case where the CP decomposition is well posed is the case of the matrices—order-2 tensors—where the low-rank approximation can always (given all singular values are distinct) be found by the Eckart-Young theorem [Eckart and Young, 1936]. The third case is the case of the nonnegative CP decomposition which is always well posed [Lim and Comon, 2009]. However, in general the best rank approximation problem is ill-posed for tensors, including the symmetric case [Comon et al., 2008, De Silva and Lim, 2008]. Moreover, computing the best rank approximation for tensors, even the best rank-1 approximation, is NP-hard in general [Hillar and Lim, 2013].

2.1.4 CP Uniqueness and Identifiability

As we have already mentioned, each rank-1 term in the CP decomposition is not uniquely represented by an outer product of vectors ; moreover the order of the rank-one terms can be arbitrary changed. Therefore, one is more interested in the uniqueness of the rank-1 terms, and scalar factors λ_r (if any), in the CP decomposition, which is sometimes referred to as the *essential uniqueness* or *essential identifiability* (up to permutation and scaling) [Comon, 2014].

3. In fact, the norm in (2.14) is the Euclidean norm of any unfolding of the tensor $\boldsymbol{\mathcal{T}}$.

Sufficient Condition. The essential uniqueness of the CP-decomposition (2.8) of a tensor $\boldsymbol{\mathcal{T}}$ is ensured if the sufficient condition below is satisfied [Kruskal, 1977, Sidiropoulos and Bro, 2000, Stegeman and Sidiropoulos, 2007, Comon et al., 2009a] :

$$\sum_{s=1}^{S} \text{rank}_k\big(\mathbf{U}^{(s)}\big) \geq 2R + S - 1,$$

where R is the rank of a tensor and the *Kruskal rank* or *k-rank* of a matrix $\mathbf{A}$, denoted $\text{rank}_k(\mathbf{A})$, is the maximal number r such that any set of r columns of $\mathbf{A}$ is linearly independent [Kruskal, 1977]. With probability one, this condition is equivalent to

$$\sum_{s=1}^{S} \min\left[M_s, F\right] \geq 2F + S - 1,$$

where M_s denotes the s-th dimension of $\boldsymbol{\mathcal{T}}$ and F is the number of rank-1 terms in the CPD of $\boldsymbol{\mathcal{T}}$. This condition is *not necessary*. Some attempts to relax this condition for special types of tensors or in general were made by De Lathauwer [2006], Comon et al. [2009a], Domanov and De Lathauwer [2013a,b, 2014, 2016]. In the symmetric tensor case, the symmetric CP decomposition is essentially unique with probability 1 if the dimension does not exceed the order [Mella, 2006], however, this condition is quite restrictive as well.

Consequences. This condition basically says that if one tries to approximate a tensor by a sum of rank-1 terms, there is an upper limit on the number of terms in such approximation which guarantees uniqueness. This limit can be increased in some special cases (e.g. with additional sparsity constraints), however, in general, this result is a motivation for a low-rank tensor approximation. We very briefly discuss some of the algorithms for the computation the CP decomposition in Section 2.3.

2.2 Higher Order Statistics

In this section, we demonstrate how the CP decomposition of tensor is related to the estimation and inference in latent linear models. In fact, population higher-order statistics of some latent linear models (see, e.g., Sections 2.2.2 and 2.2.3) are tensors in the CP form with the factors of this CP decomposition being the linear transformation matrix(ces). Approximating such decomposition of sample estimators of these higher-order statistics (see Section 2.3 for some algorithms), thus allows to perform the estimation in these models (see Sections 2.4.2, 3.4, and 4.5).

2.2.1 Moments, Cumulants, and Generating Functions

Higher-order moments and higher-order cumulants are examples of higher-order statistics. There is a one-to-one correspondence between moments and cumulants. Moreover, cumulants are often introduced as functions of moments. Despite a simpler definition of higher-order moments, most of statistical calculations using cumulants

are easier than most of statistical computations using moments, especially, when dealing with independent random variables. Hence, the importance of cumulants.

In this section, we recall the definitions of moments, cumulants and closely related concepts, such as the moment- and cumulant-generating functions as well as the first and second characteristic functions. Higher-order cumulants of ICA (see Section 2.2.2), higher-order moment-based statistics of LDA (see Section 2.2.3), as well as higher-order cumulant-based statistics of other models (see Sections 3.3.2 and 4.4.1) are tensors in the form of (sometimes symmetric and sometimes non-negative) CP decomposition. The cumulant generating function is used for defining the generalized cumulants (in Section 4.4) and the proof of the identifiability results for the models introduced in Chapter 4 is based on the second characteristic function and its properties (see Section 4.3.2). For details on the topics covered in this section, see, e.g., McCullagh [1987], Stuart and Ord [1994], De Lathauwer [2010].

Moments. The *order-S moment* tensor $\boldsymbol{\mu}_{\mathbf{x}}^{(S)} \in \mathbb{R}^{M \times M \times \cdots \times M}$ of an $\mathbb{R}^M$-valued random variable $\mathbf{x}$ is defined element-wise as

$$\left[\boldsymbol{\mu}_{\mathbf{x}}^{(S)}\right]_{m_1 m_2 \ldots m_S} := \mathbb{E}\left(x_{m_1} x_{m_2} \ldots x_{m_S}\right), \tag{2.15}$$

for all $m_1, m_2, \ldots, m_S \in [M]$. Note that the order-1 moment of a random vector $\mathbf{x}$ is equal to its mean, i.e. $\boldsymbol{\mu}_{\mathbf{x}}^{(1)} = \mathbb{E}(\mathbf{x})$. Just as the expectation of a random variable, some random variables do not have finite moments and sometimes moments are not defined at all. For example, no mean or higher moments exist for the Cauchy distribution. Another example is the student's t-distribution with infinite moments of the order-3 or higher.

The Moment-Generating Function. The *moment-generating function (MGF)* of an $\mathbb{R}^M$-valued random variable $\mathbf{x}$ is

$$\mathbf{M}_{\mathbf{x}}(\mathbf{t}) := \mathbb{E}(e^{\mathbf{t}^\top \mathbf{x}}), \tag{2.16}$$

for any $\mathbf{t} \in \mathbb{R}^M$. The MGF of a random variable does not always exist (since the expectation may not be finite). When the MGF is finite in the neighborhood of zero, it uniquely determines the probability distribution, i.e. if two random variables have the same MGF, they have the same distribution (except possibly at a countable number of points having 0 probability).

The Taylor series expansion of the MGF at zero has the following form

$$\begin{aligned} \mathbf{M}_{\mathbf{x}}(\mathbf{t}) &= \sum_{j=0}^{\infty} \frac{1}{j!} \nabla^j \mathbf{M}_{\mathbf{x}}(\mathbf{0}) \times_1 \mathbf{t} \times_2 \mathbf{t} \cdots \times_j \mathbf{t} \\ &= 1 + \langle \mathbf{t}, \nabla \mathbf{M}_{\mathbf{x}}(\mathbf{0}) \rangle + \frac{1}{2!} \left\langle \mathbf{t}, \nabla^2 \mathbf{M}_{\mathbf{x}}(\mathbf{0})\, \mathbf{t} \right\rangle + \ldots \end{aligned} \tag{2.17}$$

where $\langle \cdot, \cdot \rangle$ is the inner product, i.e. $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^\top \mathbf{b}$, the mode-$s$ product $\times_s$ of a tensor with a vector is defined in (2.2) (note that $K = 1$ for a vector), and the derivatives

of the MGF are the moments, i.e. $\nabla^S \mathbf{M}_{\mathbf{x}}(\mathbf{0}) = \boldsymbol{\mu}_{\mathbf{x}}^{(S)}$ for any $S = 1, 2, 3, \ldots$. Many distributions, e.g., normal, exponential, Poisson, etc., are uniquely defined by their moments. Such distributions are called M-determinate. However, unlike the MGF, two (or even infinitely many) distinct distributions may have the same moments. Such distributions are called M-equivalent and each of them is M-indeterminate. One of the best known M-indeterminate distributions is the log-normal distribution [see, e.g., Heyde, 1963]. Indeed, the log-normal distribution

$$p(x) := \frac{1}{x\sqrt{2\pi}} \exp\left(-\frac{(\ln x)^2}{2}\right), \quad x > 0$$

and the "perturbed" log-normal distribution

$$q(x) := p(x)(1 + \sin(2\pi \log x))), \quad x > 0$$

have the same moments [see also Durrett, 2013, Chapter 3]. Another example of M-indeterminate distributions is the cube of the Laplace distribution [Stoyanov, 2006]. For more details and examples see, e.g., Stoyanov [2006], Lin and Stoyanov [2009], Durrett [2013] and references therein. Note however that non-uniqueness occurs only when the MGF $\mathbf{M}_{\mathbf{x}}(\mathbf{t})$ is not analytic at zero [McCullagh, 1987, Section 2.2.1].

The First Characteristic Function. The *first characteristic function* of an $\mathbb{R}^M$-valued random variable $\mathbf{x}$ is defined as

$$\boldsymbol{\phi}_{\mathbf{x}}(\mathbf{t}) := \mathbb{E}(e^{i\mathbf{t}^\top \mathbf{x}}), \tag{2.18}$$

for any $\mathbf{t} \in \mathbb{R}^M$, where i is the imaginary unit. From the definition in (2.18) it follows that the first characteristic function $\boldsymbol{\phi}_{\mathbf{x}}(\mathbf{t})$ is the Fourier transform of the probability measure (if the random variable $\mathbf{x}$ has a probability density). Unlike the moment-generating function, the first characteristic function always exists. Moreover, in accordance with the *uniqueness theorem* [see, e.g., Jacod and Protter, 2004, Durrett, 2013], the Fourier transform of a probability measure on $\mathbb{R}^M$ characterizes this probability measure. This means that the first characteristic function of any real-valued random variable completely defines its probability distribution. If a random variable admits a probability density function (PDF), then there is a one-to-one correspondence between this PDF and the first characteristic function : the latter is the inverse Fourier transform of the former.

Cumulants. The *order-S cumulant* tensor $\boldsymbol{\kappa}_{\mathbf{x}}^{(S)} \in \mathbb{R}^{M \times M \times \cdots \times M}$ of an $\mathbb{R}^M$-valued random variable $\mathbf{x}$ is defined element-wise as

$$\left[\boldsymbol{\kappa}_{\mathbf{x}}^{(S)}\right]_{m_1 m_2 \ldots m_S} := \sum (-1)^{s-1}(s-1)! \mathbb{E}\left[\prod_{j \in P_1} x_j\right] \mathbb{E}\left[\prod_{j \in P_2} x_j\right] \ldots \mathbb{E}\left[\prod_{j \in P_s} x_j\right], \tag{2.19}$$

where the summation involves all possible partitions $[P_1, P_2, \ldots, P_s]$, for $1 \leq s \leq S$, of the integers $\{m_1, m_2, \ldots, m_S\}$. Note that the first cumulant is equal to the mean; the second cumulant is equal to the covariance matrix; the third cumulant is equal

to the third central moment :

$$\begin{aligned}\boldsymbol{\kappa}_{\mathbf{x}}^{(1)} &= \mathbb{E}(\mathbf{x}),\\ \boldsymbol{\kappa}_{\mathbf{x}}^{(2)} &= \mathrm{cov}(\mathbf{x}),\\ \boldsymbol{\kappa}_{\mathbf{x}}^{(3)} &= \mathbb{E}\left[(\mathbf{x}-\mathbb{E}(\mathbf{x}))\otimes(\mathbf{x}-\mathbb{E}(\mathbf{x}))\otimes(\mathbf{x}-\mathbb{E}(\mathbf{x}))\right].\end{aligned} \tag{2.20}$$

However, for order-4 and higher, the cumulants are different from the central moments and the interpretation of cumulants of order higher than 3 is not straightforward. For example, the order-4 cumulant is equal to

$$\begin{aligned}\left[\boldsymbol{\kappa}_{\mathbf{x}}^{(4)}\right]_{m_1m_2m_3m_4} &= \mathbb{E}[\bar{x}_{m_1}\bar{x}_{m_2}\bar{x}_{m_3}\bar{x}_{m_4}] - \mathbb{E}[\bar{x}_{m_1}\bar{x}_{m_2}]\mathbb{E}[\bar{x}_{m_3}\bar{x}_{m_4}]\\ &\quad - \mathbb{E}[\bar{x}_{m_1}\bar{x}_{m_3}]\mathbb{E}[\bar{x}_{m_2}\bar{x}_{m_4}] - \mathbb{E}[\bar{x}_{m_1}\bar{x}_{m_4}]\mathbb{E}[\bar{x}_{m_2}\bar{x}_{m_3}],\end{aligned} \tag{2.21}$$

where $\bar{x}_m$ stands for the centered m-th element of the random vector $\mathbf{x}$, i.e. $\bar{x}_m = x_m - \mathbb{E}[x_m]$. The cumulants of a set of random variables give an indication of their mutual statistical dependence (as we will see below, completely independent variables have zero cross-cumulants) and the higher-order cumulants of a single random variable are some measure of its non-Gaussianity since the cumulants of a Gaussian random variable are all equal to zero for $S > 2$.

The (m, m')-th element of a covariance matrix $\mathrm{cov}(\mathbf{x})$, which is an order-2 cumulant, is sometimes called *cross-covariance* if $m \neq m'$. By analogy, we sometimes refer to non-diagonal elements of higher-order cumulants as *cross-cumulants*.

The Cumulant-Generating Function. The *cumulant-generating function (CGF)* of an $\mathbb{R}^M$-valued random variable $\mathbf{x}$ is defined as

$$\mathbf{K}_{\mathbf{x}}(\mathbf{t}) := \log \mathbb{E}(e^{\mathbf{t}^\top \mathbf{x}}), \tag{2.22}$$

for any $\mathbf{t} \in \mathbb{R}^M$. That is the CGF is the logarithm of the MGF defined in (2.16). Likewise moments are the coefficients of the Taylor series of the moment-generating function evaluated at zero, cumulants are the coefficients of the Taylor series of the cumulant-generating function evaluated at zero :

$$\boldsymbol{\kappa}_{\mathbf{x}}^{(s)} = \nabla^s \mathbf{K}_{\mathbf{x}}(\mathbf{0}), \quad s = 1, 2, \ldots \tag{2.23}$$

Note that this way to define cumulants will be important in Chapter 4, where we introduce the so-called *generalized cumulants*. Since there is a one-to-one correspondence between moments and cumulants, the discussion related to the uniqueness of moments extends directly to cumulants. Marcinkiewicz [1939] showed that the normal distribution is the only distribution whose cumulant generating function is a polynomial, i.e. the only distribution having a finite number of non-zero cumulants (we will use this property in the proof on the identifiability theorem from Section 4.3.2 in Appendix 4.3.3). The MGF of a Poisson random variable with mean λ is $\mathbf{M}_x(t) = \exp[\lambda(e^t - 1)]$; and the CGF of this random variable is $\mathbf{K}_x(t) = \lambda(e^t - 1)$. Consequently, all the cumulants of a Poisson random variable are

equal to the mean (we will use this property for the derivation of the DICA cumulants in Section 3.3.2).

The Second Characteristic Function. The *second characteristic function* of an $\mathbb{R}^M$-valued random variable $\mathbf{x}$ is defined as

$$\boldsymbol{\psi}_{\mathbf{x}}(\mathbf{t}) := \log \mathbb{E}(e^{i\,\mathbf{t}^\top \mathbf{x}}), \tag{2.24}$$

for any $\mathbf{t} \in \mathbb{R}^M$. Hence, the second characteristic function is the logarithm [4] of the first characteristic function $\boldsymbol{\psi}_{\mathbf{x}}(\mathbf{t})$ defined in (2.18) and, therefore, it also uniquely defines a distribution. The second characteristic function is the key tool for the identifiability proof in Chapter 4.

Properties of Cumulants. Since the definition of higher-order moments is somewhat simpler, they might seem more interesting at the first sight than higher-order cumulants. However, cumulants have a number of important properties which they do not share with moments. Some of these properties are as follows [Nikias and Mendel, 1993, Nikias and Petropulu, 1993, De Lathauwer, 2010] :

1. *Symmetry :* real moments $\boldsymbol{\mu}_{\mathbf{x}}^{(S)}$ and cumulants $\boldsymbol{\kappa}_{\mathbf{x}}^{(S)}$ are symmetric tensors (see (2.4) for definition), i.e. they remain unchanged under any permutation of their indices.

2. *Multilinearity :* if an $\mathbb{R}^M$-valued random variable $\mathbf{x}$ and an $\mathbb{R}^K$-valued random variable $\boldsymbol{\alpha}$ are linearly dependent, i.e. $\mathbf{x} = \mathbf{D}\boldsymbol{\alpha}$ for some $\mathbf{D} \in R^{M\times K}$, then moments $\boldsymbol{\mu}_{\mathbf{x}}^{(S)}$ and cumulants $\boldsymbol{\kappa}_{\mathbf{x}}^{(S)}$, for all $S = 1, 2, \ldots$, are multilinear :

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{x}}^{(S)} &= \boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(S)} \times_1 \mathbf{D}^\top \times_2 \mathbf{D}^\top \cdots \times_S \mathbf{D}^\top, \\ \boldsymbol{\kappa}_{\mathbf{x}}^{(S)} &= \boldsymbol{\kappa}_{\boldsymbol{\alpha}}^{(S)} \times_1 \mathbf{D}^\top \times_2 \mathbf{D}^\top \cdots \times_S \mathbf{D}^\top. \end{aligned} \tag{2.25}$$

   Let $\mathbf{D}$ be a basis transformation ($M = K$). Then (2.25) is the rule in accordance to which moments and cumulants change under basis transformations. This is exactly the multilinearity property of tensors in (2.3) and therefore is the reason we can refer to moments and cumulants as *tensors*.

3. *Even distribution :* if a real random variable x has an even probability density function $p_x(x)$, i.e., $p_x(x)$ is symmetric about the origin, then the odd moments, $\boldsymbol{\mu}_x^{(S)}$, and cumulants, $\boldsymbol{\kappa}_x^{(S)}$ for $S = 1, 3, 5, \ldots$, of x vanish.

4. *Independence :* if all the elements of an $\mathbb{R}^K$-valued random variable $\boldsymbol{\alpha}$ are mutually independent, then the order-S cumulant $\boldsymbol{\kappa}_{\boldsymbol{\alpha}}^{(S)}$ of this variable is diagonal for all $S = 1, 2, 3, \ldots$. For example, for order-2 cumulant (i.e. the covariance matrix) it holds

$$\boldsymbol{\kappa}_{\boldsymbol{\alpha}}^{(2)} = \mathrm{cov}(\boldsymbol{\alpha}) = \mathrm{Diag}[\mathrm{var}(\boldsymbol{\alpha})] \tag{2.26}$$

4. Note that the complex logarithm in general is not *uniquely* defined. It is common to choose the *principal* value of $\log(z)$, where $z = x + iy$ is a complex number, as the logarithm whose imaginary part lies in the interval $[-\pi, \pi]$.

and in general for order-S cumulant (element-wise) one has

$$\left[\boldsymbol{\kappa}_{\boldsymbol{\alpha}}^{(S)}\right]_{m_1 m_2 \ldots m_S} = \delta(m_1, m_2, \ldots, m_S)\mathbb{E}[(x_{m_1} - \mathbb{E}(x_{m_1}))^S], \tag{2.27}$$

where δ is the Kronecker delta, i.e. $\delta(m_1, m_2, \ldots, m_S) = 1$ if all indices are equal $m_1 = m_2 = \cdots = m_S$ and 0 otherwise. Such property in general does not hold for moments.

5. *Sum of independent variables :* if an $\mathbb{R}^M$-valued random variable $\mathbf{x}$ and an $\mathbb{R}^M$-valued random variable $\mathbf{y}$ are independent, i.e. $\mathbf{x} \perp\!\!\!\perp \mathbf{y}$ then the cumulant of their sum is equal to the sum of cumulants :

$$\boldsymbol{\kappa}_{\mathbf{x}+\mathbf{y}}^{(S)} = \boldsymbol{\kappa}_{\mathbf{x}}^{(S)} + \boldsymbol{\kappa}_{\mathbf{y}}^{(S)}, \quad S = 1, 2, 3, \ldots \tag{2.28}$$

This property does not hold for moments either.

6. *non-Gaussianity :* if y is a Gaussian variable with the same mean and variance as a given random variable x, then for $S \geq 3$ it holds :

$$\boldsymbol{\kappa}_x^{(S)} = \boldsymbol{\mu}_x^{(S)} - \boldsymbol{\mu}_y^{(S)}. \tag{2.29}$$

As a consequence, higher-order cumulants of a Gaussian random variable are all zero. Therefore, in combination with the multilinearity property, this leads to an interesting property that higher-order cumulants do not change under additive Gaussian noise. That is, if z is a Gaussian random variable, it holds that $\boldsymbol{\kappa}_{x+z}^{(S)} = \boldsymbol{\kappa}_x^{(S)}$ for all $S \geq 3$.

7. *The law of total cumulance :* for two $\mathbb{R}^M$-valued random variables $\mathbf{x}$ and $\mathbf{y}$ the following laws hold. For $S = 1$, the *law of total expectation*, also known under the name of the *tower property* :

$$\left[\boldsymbol{\kappa}_{\mathbf{x}}^{(1)}\right]_m = \mathbb{E}[x_m] = \mathbb{E}[\mathbb{E}(x_m|\mathbf{y})]. \tag{2.30}$$

For $S = 2$, the *law of total covariance* :

$$\begin{aligned} \left[\boldsymbol{\kappa}_{\mathbf{x}}^{(2)}\right]_{m_1 m_2} &= \operatorname{cov}(x_{m_1}, x_{m_2}) \\ &= \mathbb{E}[\operatorname{cov}(x_{m_1}, x_{m_2}|\mathbf{y})] + \operatorname{cov}[\mathbb{E}(x_{m_1}|\mathbf{y}), \mathbb{E}(x_{m_2}|\mathbf{y})]. \end{aligned} \tag{2.31}$$

For $S = 3$, the *law of total cumulance* :

$$\begin{aligned} \left[\boldsymbol{\kappa}_{\mathbf{x}}^{(3)}\right]_{m_1 m_2 m_3} &= \mathbb{E}[\operatorname{cum}(x_{m_1}, x_{m_2}, x_{m_3}|\mathbf{y})] \\ &+ \operatorname{cum}[\mathbb{E}(x_{m_1}|\mathbf{y}), \mathbb{E}(x_{m_2}|\mathbf{y}), \mathbb{E}(x_{m_3}|\mathbf{y})] \\ &+ \operatorname{cov}[\mathbb{E}(x_{m_1}|\mathbf{y}), \operatorname{cov}(x_{m_2}, x_{m_3}|\mathbf{y})] \\ &+ \operatorname{cov}[\mathbb{E}(x_{m_2}|\mathbf{y}), \operatorname{cov}(x_{m_1}, x_{m_3}|\mathbf{y})] \\ &+ \operatorname{cov}[\mathbb{E}(x_{m_3}|\mathbf{y}), \operatorname{cov}(x_{m_1}, x_{m_2}|\mathbf{y})]. \end{aligned} \tag{2.32}$$

One of the key differences between the MGF and CGF (similar expressions can be

written for the first and second characteristic functions) is as follows. The MGF of a sum of two independent random variables α_1 and α_2 is equal to the product of their MGFs : $\mathbf{M}_{\alpha_1+\alpha_2}(t) = \mathbf{M}_{\alpha_1}(t)\mathbf{M}_{\alpha_2}(t)$. The CGF of a sum of two independent random variables X and Y, however, is equal to the sum of their CGFs : $\mathbf{K}_{\alpha_1+\alpha_2}(t) = \mathbf{K}_{\alpha_1}(t) + \mathbf{K}_{\alpha_2}(t)$. This explains why cumulants, but not moments, of independent variables are diagonal. The CGF is separable and taking derivatives with respect to α_1 and α_2 enforces zero cross-terms ; this does not hold for the MGF (we will make extensive use of this property in Chapter 4, e.g., when deriving the CP form of the so-called generalized cumulants for multi-view linear models in Section 4.4.2).

2.2.2 CPD of ICA Cumulants

In this section, we present a well known result that higher-order population cumulants of independent component analysis (ICA) are tensors in the form of symmetric CP decomposition (plus potentially some noise). The CP factors in this decomposition are the mixing matrix and, therefore, this structure can be used for the estimation in the ICA model. Some references to the ICA methods based on this idea include but are not limited to Cardoso [1989, 1990], Cardoso and Souloumiac [1993], Souloumiac [1993], Cardoso [1999], De Lathauwer [2010]. Note that this is only one of many other possible approaches to the estimation in the ICA model [see, e.g., Comon and Jutten, 2010] ; the details are beyond the scope of this thesis.

ICA cumulants. By the multilinearity (Property 2) and the sum of independent variables (Property 5) properties of cumulants from Section 2.2, the covariance and higher-order cumulants of the noisy ICA model (1.9) take the form :

$$\operatorname{cov}(\mathbf{x}) = \mathbf{D}\operatorname{cov}(\boldsymbol{\alpha})\mathbf{D}^\top + \operatorname{cov}(\boldsymbol{\varepsilon}), \tag{2.33}$$

$$\boldsymbol{\kappa}_{\mathbf{x}}^{(S)} = \boldsymbol{\kappa}_{\boldsymbol{\alpha}}^{(S)} \times_1 \mathbf{D}^\top \times_2 \mathbf{D}^\top \cdots \times_S \mathbf{D}^\top + \boldsymbol{\kappa}_{\boldsymbol{\varepsilon}}^{(S)}, \qquad S = 3, 4, \ldots, \tag{2.34}$$

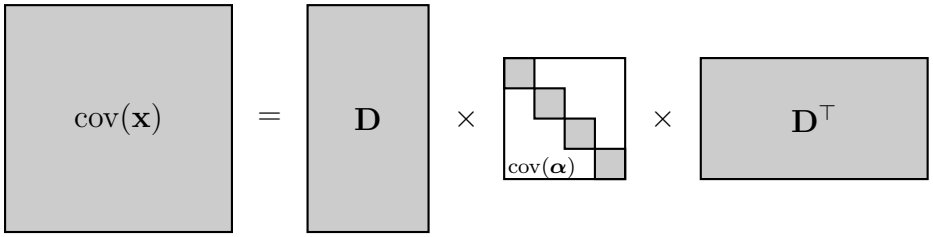
where the covariance $\operatorname{cov}(\boldsymbol{\alpha})$ and higher-order cumulants $\boldsymbol{\kappa}_{\boldsymbol{\alpha}}^{(S)}$ of the sources are diagonal by the independence property (Property 4) of cumulants since the sources are independent. Furthermore, a higher-order cumulant $\boldsymbol{\kappa}_{\boldsymbol{\varepsilon}}^{(S)}$, for $S = 3, 4, \ldots$, of the additive noise vanishes whenever the noise is Gaussian by the non-Gaussianity property (Property 6).

In the noiseless version (1.10) of ICA the noise is assumed to be zero and the covariance and higher-order cumulants take the form

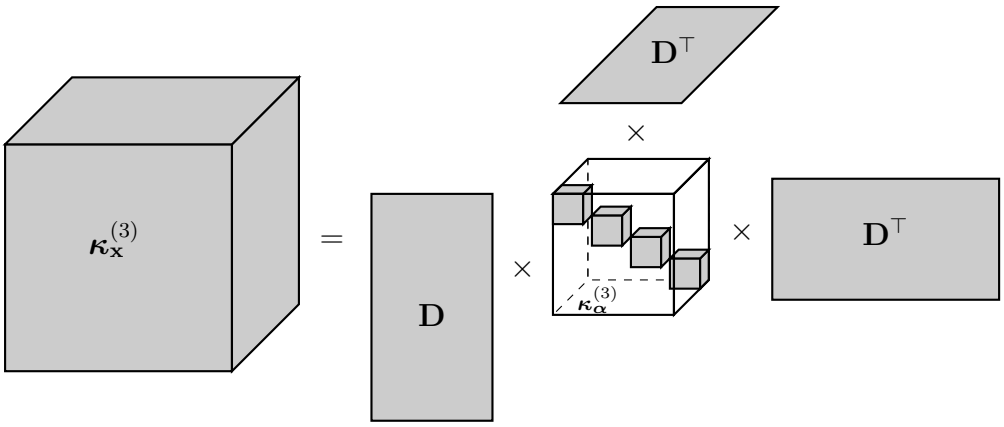
$$\operatorname{cov}(\mathbf{x}) = \mathbf{D}\operatorname{cov}(\boldsymbol{\alpha})\mathbf{D}^\top, \tag{2.35}$$

$$\boldsymbol{\kappa}_{\mathbf{x}}^{(S)} = \boldsymbol{\kappa}_{\boldsymbol{\alpha}}^{(S)} \times_1 \mathbf{D}^\top \times_2 \mathbf{D}^\top \cdots \times_S \mathbf{D}^\top, \qquad S = 3, 4, \ldots, \tag{2.36}$$

where $\operatorname{cov}(\boldsymbol{\alpha})$ and $\boldsymbol{\kappa}_{\boldsymbol{\alpha}}^{(S)}$ are diagonal. Therefore, these population ICA covariance (2.35) and cumulants (2.36) are represented as a (weighted) sum of rank-1 terms, which, in fact, is the symmetric CP decomposition (2.10) described in Section 2.1.2 (see illustration in Figure 2-3). In Section 2.4.2, we discuss how to use this decomposition for



(a) The ICA population covariance matrix (2.33).



(b) The order-3 ICA population cumulant (2.34).

FIGURE 2-3 – The ICA population covariance and order-3 cumulant.

the estimation in the ICA model.

2.2.3 CPD of LDA Moments

In this section, we present a known result that a well defined combination of higher-order population moments [5] of the LDA model (1.16), likewise ICA cumulants, are tensors in the form of the symmetric CP decomposition [Anandkumar et al., 2012a, 2013a, 2015a]. In addition to being symmetric, the CP decomposition of these *LDA moments* is also *non-negative*.

LDA Moments. For deriving the LDA moments, a document is assumed to be composed of at least three tokens : $L \geq 3$. As the LDA generative model (1.16) is only defined *conditional* on the length L, this is not too problematic. However, given that we present (see Chapter 3) models which also model L, we mention for clarity that we can suppose that all expectations and probabilities defined below are implicitly conditioning on $L \geq 3$.

The order-3 LDA population moment is defined [see, e.g., Anandkumar et al., 2012a] as the outer product of the first three tokens $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$ of a document :

$$\boldsymbol{\mu}_{\mathbf{x}}^{(3)} := \mathbb{E}(\mathbf{w}_1 \otimes \mathbf{w}_2 \otimes \mathbf{w}_3), \tag{2.37}$$

5. To be more exact, some higher-order statistics closely related to moments.

where we used the count vector $\mathbf{x}$ of a document as a general reference to these three tokens. Indeed, the tokens $\mathbf{w}_\ell$ in the LDA generative model (1.16) are conditionally independent and identically distributed given the topic intensities $\boldsymbol{\theta}$ of this document or equivalently the order of the tokens in a document does not matter [Blei et al., 2003]. Therefore, we can use any three tokens from this document to define the order-3 LDA moment :

$$\boldsymbol{\mu}_{\mathbf{x}}^{(3)} = \mathbb{E}(\mathbf{w}_{\ell_1}, \mathbf{w}_{\ell_2}, \mathbf{w}_{\ell_3}),$$

for any $1 \leq \ell_1 \neq \ell_2 \neq \ell_3 \leq L$. To highlight this arbitrary choice of tokens and to make the links with the U-statistics estimator presented later in (2.49), we thus use generic distinct ℓ_1, ℓ_2, and ℓ_3 in the definition of the LDA moments, instead of $\ell_1 = 1$, $\ell_2 = 2$, and $\ell_3 = 3$ as done by Anandkumar et al. [2012a].

Using this notation, by the law of total expectation and the properties of the Dirichlet distribution, the moments of the LDA model (1.16) take the form [Anandkumar et al., 2012a] :

$$\boldsymbol{\mu}_{\mathbf{x}}^{(1)} = \mathbb{E}(\mathbf{w}_{\ell_1}) = \mathbf{D}\frac{\mathbf{c}}{c_0}, \tag{2.38}$$

$$\boldsymbol{\mu}_{\mathbf{x}}^{(2)} = \mathbb{E}(\mathbf{w}_{\ell_1} \otimes \mathbf{w}_{\ell_2}) = \frac{c_0}{c_0+1}\boldsymbol{\mu}_{\mathbf{x}}^{(1)} \otimes \boldsymbol{\mu}_{\mathbf{x}}^{(1)} + \frac{1}{c_0(c_0+1)}\mathbf{D}\,\mathrm{Diag}\,(\mathbf{c})\,\mathbf{D}^\top, \tag{2.39}$$

$$\begin{aligned}\boldsymbol{\mu}_{\mathbf{x}}^{(3)} &= \mathbb{E}(\mathbf{w}_{\ell_1} \otimes \mathbf{w}_{\ell_2} \otimes \mathbf{w}_{\ell_3}) \\ &= C_1 \left[\mathbb{E}(\mathbf{w}_{\ell_1} \otimes \mathbf{w}_{\ell_2} \otimes \boldsymbol{\mu}_{\mathbf{x}}^{(1)}) + \mathbb{E}(\mathbf{w}_{\ell_1} \otimes \boldsymbol{\mu}_{\mathbf{x}}^{(1)} \otimes \mathbf{w}_{\ell_3}) + \mathbb{E}(\boldsymbol{\mu}_{\mathbf{x}}^{(1)} \otimes \mathbf{w}_{\ell_2} \otimes \mathbf{w}_{\ell_3})\right], \\ &- C_2\, \boldsymbol{\mu}_{\mathbf{x}}^{(1)} \otimes \boldsymbol{\mu}_{\mathbf{x}}^{(1)} \otimes \boldsymbol{\mu}_{\mathbf{x}}^{(1)} + C_3 \sum_{k=1}^{K} c_k\, \mathbf{d}_k \otimes \mathbf{d}_k \otimes \mathbf{d}_k.\end{aligned} \tag{2.40}$$

where $C_1 = c_0(c_0+2)^{-1}$, $C_2 = 2c_0^2\left[(c_0+1)(c_0+2)\right]^{-1}$, $C_3 = 2\left[c_0(c_0+1)(c_0+2)\right]^{-1}$, and $\otimes$ denotes the tensor product.

The last term in Equation (2.39) and the last term in Equation (2.40) are a matrix and an order-3 tensor in the form of non-negative symmetric CP decomposition. Therefore, moving all but these terms from the LHS to the RHS of these equations, we obtain :

$$(Pairs) = \mathbf{S}^{LDA} := \boldsymbol{\mu}_{\mathbf{x}}^{(2)} - \frac{c_0}{c_0+1}\boldsymbol{\mu}_{\mathbf{x}}^{(1)} \otimes \boldsymbol{\mu}_{\mathbf{x}}^{(1)}, \qquad \text{LDA } \mathbf{S}\text{-moment} \tag{2.41}$$

$$\begin{aligned}(Triples) = \boldsymbol{\mathcal{T}}^{LDA} &:= \boldsymbol{\mu}_{\mathbf{x}}^{(3)} + C_2\, \boldsymbol{\mu}_{\mathbf{x}}^{(1)} \otimes \boldsymbol{\mu}_{\mathbf{x}}^{(1)} \otimes \boldsymbol{\mu}_{\mathbf{x}}^{(1)}. \qquad \text{LDA } \boldsymbol{\mathcal{T}}\text{-moment} \\ &- \frac{c_0}{c_0+2}\left[\mathbb{E}(\mathbf{w}_{\ell_1} \otimes \mathbf{w}_{\ell_2} \otimes \boldsymbol{\mu}_{\mathbf{x}}^{(1)}) + \mathbb{E}(\mathbf{w}_{\ell_1} \otimes \boldsymbol{\mu}_{\mathbf{x}}^{(1)} \otimes \mathbf{w}_{\ell_3}) + \mathbb{E}(\boldsymbol{\mu}_{\mathbf{x}}^{(1)} \otimes \mathbf{w}_{\ell_2} \otimes \mathbf{w}_{\ell_3})\right],\end{aligned} \tag{2.42}$$

where $C_2 = 2c_0^2\left[(c_0+1)(c_0+2)\right]^{-1}$. Slightly abusing terminology, we refer to the matrix $\mathbf{S}^{LDA}$ and tensor $\boldsymbol{\mathcal{T}}^{LDA}$ as the *LDA moments*. They have the following *diagonal*

structure

$$\mathbf{S}^{LDA} = \frac{1}{c_0(c_0+1)} \sum_{k=1}^{K} c_k \, \mathbf{d}_k \otimes \mathbf{d}_k, \tag{2.43}$$

$$\boldsymbol{\mathcal{T}}^{LDA} = \frac{2}{c_0(c_0+1)(c_0+2)} \sum_{k=1}^{K} c_k \, \mathbf{d}_k \otimes \mathbf{d}_k \otimes \mathbf{d}_k, \tag{2.44}$$

which is the symmetric non-negative CP decomposition (2.10).

Asymptotically Unbiased Finite Sample Estimators for the LDA Moments. In the rest of this section, we present asymptotically unbiased finite sample estimators for the LDA moments. In Appendix C.1.1, we also derive expressions for fast implementation of these estimators.

Given realizations $\mathbf{w}_{n\ell}$, $n \in [N]$, $\ell \in [L_n]$, of the token random variable $\mathbf{w}_\ell$, we now give the expressions for the finite sample estimates [6] of the LDA moments $\mathbf{S}^{LDA}$ and $\boldsymbol{\mathcal{T}}^{LDA}$. We use the notation $\widehat{\mathbb{E}}$ below to express a U-statistics empirical expectation over the token within a documents, uniformly averaged over the whole corpus. For example,

$$\widehat{\mathbb{E}}(\mathbf{w}_{\ell_1} \otimes \mathbf{w}_{\ell_2} \otimes \widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}) := \frac{1}{N} \sum_{n=1}^{N} \frac{1}{L_n(L_n-1)} \sum_{\ell_1=1}^{L_n} \sum_{\substack{\ell_2=1 \\ \ell_2 \neq \ell_1}}^{L_n} \mathbf{w}_{\ell_1} \otimes \mathbf{w}_{\ell_2} \otimes \widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)},$$

where L_n stands for the number of tokens in the n-th document. This gives the following expressions for the finite sample estimates of the LDA moments :

$$\widehat{\mathbf{S}}^{LDA} := \widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)} - \frac{c_0}{c_0+1} \widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)} \otimes \widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}, \tag{2.45}$$

$$\begin{aligned} \widehat{\boldsymbol{\mathcal{T}}}^{LDA} := \widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(3)} + \frac{2c_0^2}{(c_0+1)(c_0+2)} \widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)} \otimes \widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)} \otimes \widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)} \\ - \frac{c_0}{c_0+2} \left[\widehat{\mathbb{E}}(\mathbf{w}_{\ell_1} \otimes \mathbf{w}_{\ell_2} \otimes \widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}) + \widehat{\mathbb{E}}(\mathbf{w}_{\ell_1} \otimes \widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)} \otimes \mathbf{w}_{\ell_3}) + \widehat{\mathbb{E}}(\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)} \otimes \mathbf{w}_{\ell_2} \otimes \mathbf{w}_{\ell_3}) \right], \end{aligned} \tag{2.46}$$

where, as suggested by Anandkumar et al. [2014], unbiased U-statistics estimates of

6. Note that because non-linear functions of $\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}$ appear in the expression for $\widehat{\mathbf{S}}^{LDA}$ (2.45) and $\widehat{\boldsymbol{\mathcal{T}}}^{LDA}$ (2.46), the estimator is biased, i.e., $\mathbb{E}\big[\widehat{\mathbf{S}}^{LDA}\big] \neq \mathbf{S}^{LDA}$. The bias is small though : $\|\mathbb{E}(\widehat{\mathbf{S}}^{LDA}) - \widehat{\mathbf{S}}\| = O(N^{-1})$ and the estimator is asymptotically unbiased. This is in contrast with the estimator for the DICA cumulants (see Section 3.3.2) which is easily made unbiased.

$\boldsymbol{\mu}_{\mathbf{x}}^{(1)}$, $\boldsymbol{\mu}_{\mathbf{x}}^{(2)}$ and $\boldsymbol{\mu}_{\mathbf{x}}^{(3)}$ are :

$$\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)} := \widehat{\mathbb{E}}(\mathbf{w}_\ell) = N^{-1} \sum_{n=1}^{N} L_n^{-1} \sum_{\ell=1}^{L_n} \mathbf{w}_{n\ell}, \tag{2.47}$$

$$\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)} := \widehat{\mathbb{E}}(\mathbf{w}_{\ell_1} \otimes \mathbf{w}_{\ell_2}) = N^{-1} \sum_{n=1}^{N} \frac{1}{L_n(L_n-1)} \sum_{\ell_1=1}^{L_n} \sum_{\substack{\ell_2=1 \\ \ell_2 \neq \ell_1}}^{L_n} \mathbf{w}_{n\ell_1} \otimes \mathbf{w}_{n\ell_2}, \tag{2.48}$$

$$\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(3)} := \widehat{\mathbb{E}}(\mathbf{w}_{\ell_1} \otimes \mathbf{w}_{\ell_2} \otimes \mathbf{w}_{\ell_3}) = N^{-1} \sum_{n=1}^{N} \delta_{3n} \sum_{\ell_1=1}^{L_n} \sum_{\substack{\ell_2=1 \\ \ell_2 \neq \ell_1}}^{L_n} \sum_{\substack{\ell_3=1 \\ \ell_3 \neq \ell_2 \\ \ell_3 \neq \ell_1}}^{L_n} \mathbf{w}_{n\ell_1} \otimes \mathbf{w}_{n\ell_2} \otimes \mathbf{w}_{n\ell_3}. \tag{2.49}$$

Here, the vectors $\boldsymbol{\delta}_1$, $\boldsymbol{\delta}_2$ and $\boldsymbol{\delta}_3 \in \mathbb{R}^N$ are defined element-wise as $\delta_{1n} := L_n^{-1}$; $\delta_{2n} := (L_n(L_n-1))^{-1}$, i.e., $\delta_{2n} = \left[\binom{L_n}{2} 2!\right]^{-1}$ is the number of times to choose an ordered pair of tokens out of L_n tokens ; $\delta_{3n} := (L_n(L_n-1)(L_n-2))^{-1}$, i.e., $\delta_{3n} = \left[\binom{L_n}{3} 3!\right]^{-1}$ is the number of times to choose an ordered triple of tokens out of L_n tokens. Note that the vectors $\boldsymbol{\delta}_1$, $\boldsymbol{\delta}_2$, and $\boldsymbol{\delta}_3$ have nothing to do with the Kronecker delta δ.

There is a slight abuse of notation in the expressions above as $\mathbf{w}_\ell$ is sometimes treated as a random variable (i.e., in $\widehat{\mathbb{E}}(\mathbf{w}_\ell)$, $\widehat{\mathbb{E}}(\mathbf{w}_{\ell_1} \otimes \mathbf{w}_{\ell_2})$, etc.) and sometimes as its realization. However, the difference is clear from the context.

2.3 Algorithms for the CP Decomposition

In this section we review two special types of CP decompositions : the orthogonal symmetric CP decomposition (in Section 2.3.1) and non-orthogonal non-symmetric CP decomposition (in Section 2.3.2) and discuss some of the algorithms for the approximation of these decompositions. The orthogonal decomposition is further used for the prewhitening based estimation algorithms for the LDA and discrete ICA models (in Chapter 3) and the non-orthogonal decomposition is the basis for the new estimation algorithms for multi-view models (in Chapter 4).

2.3.1 Algorithms for Orthogonal Symmetric CPD

Orthogonal Symmetric CP Decomposition. Given a symmetric $K \times K \times K$ tensor $\boldsymbol{\mathcal{T}}$, the goal is to approximate this tensor as follows

$$\boldsymbol{\mathcal{T}} \approx \boldsymbol{\mathcal{G}} \times_1 \mathbf{V}^\top \times_2 \mathbf{V}^\top \times_3 \mathbf{V}^\top, \tag{2.50}$$

where the core tensor $\boldsymbol{\mathcal{G}}$ is diagonal with at most one zero diagonal element and the matrix $\mathbf{V} \in \mathbb{R}^{K \times K}$ is orthogonal, which can be equivalently rewritten as

$$\boldsymbol{\mathcal{T}} \approx \sum_{k=1}^{K} g_k \, \mathbf{v}_k \otimes \mathbf{v}_k \otimes \mathbf{v}_k = \sum_{k=1}^{K} g_k \, \mathbf{v}_k^{\otimes 3}, \tag{2.51}$$

where $g_k = \mathcal{G}_{kkk}$ and the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_K$ are the columns of $\mathbf{V}$ and are orthogonal as well. In fact, this is a special case of the symmetric orthogonal CP decomposition, since the number of factors is equal to the dimension K. However, moment-based estimation methods for latent linear models can often be reduced to such decomposition, which explains their interest. The orthogonal symmetric tensor decomposition has been first investigated by Comon [1994]. The proposed algorithm was based on the idea of Jacobi-like sweeping.

The decomposition in (2.51) can be seen as a direct extension of the symmetric matrix eigendecomposition to tensors (see also Section 2.1.2). However, the tensor decomposition in (2.50) or (2.51) is very different from its matrix predecessor. For example, the Eckart–Young SVD approximation theorem does not extend to this decomposition [Kolda, 2003] and not every symmetric tensor admits such (exact) decomposition. Contrary to the matrix case, this tensor decomposition is up to trivial indeterminacies unique if at most one diagonal entry of the core tensor $\boldsymbol{\mathcal{G}}$ is equal to zero [De Lathauwer, 2010].

Symmetric tensors which admit an exact decomposition in the form (2.51) are called *orthogonally decomposable* [Kolda, 2001]. In general, symmetric tensors *do not* necessary admit an orthogonal CP decomposition (see Robeva [2016] for a classification of tensors that have such decompositions). This leaves a room for different algorithms, which deal with the estimation error in a different manner. In this section, we discuss two types of such algorithms : one—*orthogonal joint diagonalization*—is based on the ideas of contracting a tensor to matrices and jointly diagonalizing them and another—*tensor power method*—is based on the extension of matrix power iterations to the tensor case.

Tensor Contraction or Projection. Since it is easier to work with matrices rather than with tensors, it is natural to define a transformation of a tensor to a matrix. One such transformation is the *contraction* (a.k.a. *projection*) of an order-3 tensor $\boldsymbol{\mathcal{T}}$ with (onto) a vector $\mathbf{a}$, which is defined element-wise as follows

$$[\boldsymbol{\mathcal{T}}(\mathbf{a})]_{k_1 k_2} = \sum_{k_3=1}^{K} \mathcal{T}_{k_1 k_2 k_3} a_{k_3}. \tag{2.52}$$

This definition can naturally be extended to higher-order tensors. For example, a contraction of an order-4 tensor $\boldsymbol{\mathcal{F}}$ with a matrix $\mathbf{A}$ is defined as

$$[\boldsymbol{\mathcal{F}}(\mathbf{A})]_{k_1 k_2} = \sum_{k_3=1}^{K} \sum_{k_4=1}^{K} \mathcal{F}_{k_1 k_2 k_3 k_4} A_{k_3 k_4}. \tag{2.53}$$

Such matrix $\boldsymbol{\mathcal{F}}(\mathbf{A})$ is also known in the literature under the name of a *cumulant matrix* [see, e.g. Cardoso, 1999]. Note that, although we are not going to use order-4 tensors in the algorithms described in this or other sections, this extension justifies that these algorithms can also be extended.

The Eigendecomposition Based Algorithm

For a tensor in the (nearly) orthogonal form, e.g. (2.51), the contraction with some vector $\mathbf{a}$ takes the form :

$$\boldsymbol{\mathcal{T}}(\mathbf{a}) \approx \sum_{k=1}^{K} g_k \left\langle \mathbf{v}_k, \mathbf{a} \right\rangle \mathbf{v}_k \otimes \mathbf{v}_k = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top, \tag{2.54}$$

where $\boldsymbol{\Lambda}$ is the diagonal matrix with the k-th diagonal element equal to $\Lambda_{kk} = g_k \langle \mathbf{v}_k, \mathbf{a} \rangle$ and $\otimes$ is the outer product, i.e. $\mathbf{v}_k \otimes \mathbf{v}_k = \mathbf{v}_k \mathbf{v}_k^\top$. Since $\mathbf{V}$ is an orthogonal matrix, the expression on the RHS is actually the eigenvalue decomposition of a symmetric matrix, which is uniquely defined (up to permutation) given all the diagonal elements of the matrix $\boldsymbol{\Lambda}$ are distinct.

This suggests a straightforward algorithm for the estimation of the orthogonal CP factors $\mathbf{V}$, which we will call the *eigendecomposition (ED-) based algorithm*, consisting of two simple steps : (a) contract a tensor $\boldsymbol{\mathcal{T}}$ with a vector $\mathbf{a}$ and (b) compute the eigendecomposition of the contraction $\boldsymbol{\mathcal{T}}(\mathbf{a})$. The vector $\mathbf{a}$ can be chosen, e.g., uniformly at random from the unit ℓ_2-sphere. In this case, the eigenvalues are distinct with probability 1. This turns such ED-based approach to a valid algebraic technique and, indeed, if a tensor is exactly in the form of the orthogonal symmetric CP decomposition (i.e. with the equality in (2.51)), such algorithm finds an exact solution with probability 1. This ED-based algorithm is well known in the signal processing and machine learning literature [Cardoso, 1989, 1990, Anandkumar et al., 2012a,b, Hsu and Kakade, 2013].

Orthogonal Joint Matrix Diagonalization

The things get more complicated when a tensor is not exactly in the orthogonal symmetric CPD form, which is always the case in practice. In this case, the contraction of a tensor with a vector leads to a significant loss of information since a tensor has K^3 elements, while its contraction only K^2. Therefore, in the presence of noise the estimate of $\mathbf{V}$ obtained with the ED-based algorithm is quite poor (see Section 3.5) and alternative methods are needed.

Target Matrices. One such approach is to consider several, instead of only one, contractions with random vectors [Cardoso and Souloumiac, 1993, Kuleshov et al., 2015a]. Indeed, if we project a tensor $\boldsymbol{\mathcal{T}}$ onto P vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_P$, we get :

$$\boldsymbol{\mathcal{T}}(\mathbf{a}_1) \approx \mathbf{V}\boldsymbol{\Lambda}_1\mathbf{V}^\top, \quad \boldsymbol{\mathcal{T}}(\mathbf{a}_2) \approx \mathbf{V}\boldsymbol{\Lambda}_2\mathbf{V}^\top, \quad \ldots, \quad \boldsymbol{\mathcal{T}}(\mathbf{a}_P) \approx \mathbf{V}\boldsymbol{\Lambda}_P\mathbf{V}^\top, \tag{2.55}$$

where each $\mathbf{\Lambda}_p$ is a diagonal matrix with the k-th diagonal element equal to $[\mathbf{\Lambda}_p]_{kk} = g_k\langle \mathbf{a}_p, \mathbf{v}_k\rangle$ for $p \in [P]$. In this section, we assume that the number P is fixed and known ; the choice of this number is discussed in Sections 2.4.2 and 3.4.

Let us denote each contracted tensor as a matrix $\mathbf{A}_p = \boldsymbol{\mathcal{T}}(\mathbf{a}_p)$, for $p \in [P]$. This gives the following set $\mathcal{A}$ of P matrices :

$$\mathcal{A} = \{\mathbf{A}_1,\ \mathbf{A}_2,\ \ldots,\ \mathbf{A}_P\}\,, \tag{2.56}$$

where each square $K \times K$ matrix $\mathbf{A}_p$ is expected to be approximately in the form $\mathbf{A}_p \approx \mathbf{V}\mathbf{\Lambda}_p\mathbf{V}^\top$. These matrices, which we aim to *jointly diagonalize*, are called the *target matrices*.

Orthogonal Joint Matrix Diagonalization Problem. The problem is formulated as follows : find an *orthogonal* matrix $\mathbf{Q}$ that the matrices obtained by the congruence transformation of the target matrices :

$$\mathbf{Q}^\top\mathcal{A}\mathbf{Q} = \left\{\mathbf{Q}^\top\mathbf{A}_1\mathbf{Q},\quad \mathbf{Q}^\top\mathbf{A}_2\mathbf{Q},\quad \ldots,\quad \mathbf{Q}^\top\mathbf{A}_P\mathbf{Q}\right\}, \tag{2.57}$$

are *jointly as diagonal as possible*. This can be formalized as the following optimization problem. For a square matrix $\mathbf{A}$, let $\mathrm{Off}(\mathbf{A})$ denote the sum of squared off-diagonal elements :

$$\mathrm{Off}(\mathbf{A}) = \sum_{k_1=1}^{K}\sum_{\substack{k_2=1\\k_2\neq k_1}}^{K} A_{k_1k_2}^2 = \|\mathbf{A} - \mathrm{Diag}(\mathbf{A})\|_F^2\,, \tag{2.58}$$

where $\mathrm{Diag}(\mathbf{A})$ is the diagonal matrix with the diagonal values of $\mathbf{A}$ on its diagonal, i.e. $[\mathrm{Diag}(\mathbf{A})]_{mm} = A_{mm}$. Then the *orthogonal joint matrix diagonalization (OJD)* problem is :

$$\mathbf{Q}^\star = \underset{\mathbf{Q}\in\mathcal{S}_K}{\arg\min}\sum_{p=1}^{P}\mathrm{Off}(\mathbf{Q}^\top\mathbf{A}_p\mathbf{Q}), \tag{2.59}$$

where $\mathcal{S}_K$ stands for the *Stiefel manifold*, i.e. the set of all orthogonal matrices in $\mathbb{R}^K$: $\mathcal{S}_K = \left\{\mathbf{Q} \in \mathbb{R}^K : \mathbf{Q}^\top\mathbf{Q} = \mathbf{Q}\mathbf{Q}^\top = \mathbf{I}\right\}$. Below we outline a Jacobi-like algorithm for this optimization problem.

On Gradient-Based Methods. Note that one could alternatively formulate this joint diagonalization problem by substituting the $\mathbf{V}\mathbf{\Lambda}_p\mathbf{V}^\top$, for $p \in [P]$, into the objective function (2.58) and then minimizing $\sum_{p=1}^{P}\left\|\mathbf{V}\mathbf{\Lambda}_p\mathbf{V}^\top - \mathrm{Diag}(\mathbf{V}\mathbf{\Lambda}_p\mathbf{V}^\top)\right\|_F^2$ using gradient methods, such that at the the optimum $\mathbf{A}_p \approx \mathbf{V}\mathbf{\Lambda}_p\mathbf{V}^\top$, for $p \in [P]$. However, rigorously this would not be a correct approach to solving the original problem of the CPD in (2.51), since the matrices $\mathbf{\Lambda}_p$ depend on $\mathbf{V}$ (nevertheless, some joint diagonalization algorithms are build on such idea). It is also straightforward to construct gradient based methods to optimize the problem in (2.59) over the Stiefel manifold. In fact, we compared the Jacobi-type algorithms described below with the gradient based methods implemented in the *manopt toolbox* [Boumal et al., 2014]. However, the latter was significantly slower while achieving equivalent results in terms

of the objective value and accuracy of the solution. This speed improvement of the Jacobi-like algorithm described below can be explained by the fact that this algorithm admits a closed form solution for the optimal Jacobi angle at every iteration.

Jacobi Rotation Matrix. The *Jacobi (a.k.a. Givens) rotation* matrix is the $K \times K$ matrix $\mathbf{G}(r, q, \theta)$ defined as :

$$\mathbf{G}(r,q,\theta) = \begin{array}{c} \\ r \\ \\ q \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & & r & & q & & \end{array} \\ \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \cos(\theta) & \cdots & \sin(\theta) & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -\sin(\theta) & \cdots & \cos(\theta) & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \end{array} \tag{2.60}$$

where the angle θ is called the *Jacobi (Givens) angle*. A Jacobi rotation matrix is orthogonal and corresponds to a rotation transformation in the (r, q)-plane when applied to a vector or matrix. The update $\mathbf{G}(r, q, \theta)^\top \mathbf{A}$ affects only two rows (r-th and q-th) of $\mathbf{A}$. Likewise, the update $\mathbf{A}\mathbf{G}(r, q, \theta)$ affects just two columns (r-th and q-th) of $\mathbf{A}$.

The Jacobi-Like Algorithm for Several Matrices. This algorithm is a direct extension of the *Jacobi algorithm* [Golub and Van Loan, 2013] for the computation of the eigendecomposition of a *normal* matrix. [7]

In the single matrix case, any normal matrix can be diagonalized through a *congruence* transformation by an orthogonal matrix, i.e. there exits an orthogonal $\mathbf{Q}$ such that $\mathbf{\Lambda} = \mathbf{Q}^\top \mathbf{A}\mathbf{Q}$, where $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues of $\mathbf{A}$ on the diagonal and the columns of $\mathbf{Q}$ contain the respective eigenvectors of $\mathbf{A}$. In the multiple matrix case (2.56), however, similar decomposition is in general not possible, unless the matrices are *jointly diagonalizable* [Horn and Johnson, 2013]. The problem of *approximate joint diagonalization* through a congruence transformation by an orthogonal matrix can be formulated as the optimization problem in (2.59) and below we describe an extension of the Jacobi algorithm to this problem [Bunse-Gerstner et al., 1993, Cardoso and Souloumiac, 1993, 1996].

The algorithm iteratively constructs the sequence of matrices $\mathcal{A}^{(0)}, \mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \ldots, \mathcal{A}^{(J)}$, where J is the number of iterations of the algorithm until the convergence, such that $\mathcal{A}^{(0)} = \mathcal{A}$ and

$$\mathcal{A}^{(j+1)} = \mathbf{G}(r, q, \theta^{(j)})^\top \mathcal{A}^{(j)}\ \mathbf{G}(r, q, \theta^{(j)}), \qquad j = 1, 2, 3, \ldots, J, \tag{2.61}$$

where r and q are chosen in accordance with some rule (see below) for each j. At

7. A real square matrix $\mathbf{A}$ is called *normal* if $\mathbf{A}^\top \mathbf{A} = \mathbf{A}\mathbf{A}^\top$. A symmetric matrix is a special case of normal matrices.

Algorithm 1 Jacobi-Like Orthogonal Joint Diagonalization of Several Matrices

Initialize : $\boldsymbol{\mathcal{A}}^{(0)} \leftarrow \boldsymbol{\mathcal{A}}$ and $\mathbf{Q}^{(0)} \leftarrow \mathbf{I}_M$ and iterations $j = 0$
for sweeps $1, 2, \ldots$ **do**
  for $r = 1, \ldots, K-1$ **do**
    for $q = r+1, \ldots, K$ **do**
      Compute the optimal Jacobi angle $\theta^{(j)}$ (closed form)
      Update $\mathbf{Q}^{(j+1)} \leftarrow \mathbf{Q}^{(j)}\mathbf{G}\big(r, q, \theta^{(j)}\big)$
      Update $\boldsymbol{\mathcal{A}}^{(j+1)} \leftarrow {\mathbf{G}^{(j)}}^{\top}\boldsymbol{\mathcal{A}}^{(j)}\mathbf{G}^{(j)}$
      Increase $j \leftarrow j + 1$
    end for
  end for
end for
Output : $\hat{\boldsymbol{\Lambda}}_p = {\mathbf{Q}^{(j)}}^{\top}\mathbf{A}_p^{(0)}\mathbf{Q}^{(j)}$ and $\widehat{\mathbf{V}} = [\mathbf{Q}^{(j)}]^{\top}$

every iteration, the *optimal Jacobi angle* $\theta^{(j)}$ is found as

$$\theta^{(j)} = \underset{|\theta| \le \pi/4}{\arg\min} \sum_{p=1}^{P} \operatorname{Off}\big[\mathbf{G}(r, q, \theta)^{\top}\mathbf{A}_p^{(j)}\mathbf{G}(r, q, \theta)\big]. \tag{2.62}$$

This is a linearly constrained [8] quadratic optimization problem in a single dimension and a solution always exists and a closed form expression for the optimal Jacobi angle can be obtained [Cardoso and Souloumiac, 1996, Fu and Gao, 2006, Iferroudjene et al., 2009]. This iterative procedure is summarized in Algorithm 1. We will refer to this algorithm as *orthogonal joint diagonalization* or simply *OJD*. In the ICA literature, the algorithm is widely known under the name of *joint approximate diagonalization of eigen-matrices (JADE)* [Cardoso and Souloumiac, 1993].

In the case of a single matrix $P = 1$, this algorithm is exactly equivalent to the Jacobi algorithm and the update (2.61) with the optimal Jacobi angle satisfying (2.62) has a property that each new matrix $\mathbf{A}^{(j+1)}$ is "more diagonal" than its predecessor $\mathbf{A}^{(j)}$. The algorithm converges when the off-diagonal entries, respectively, the objective $\operatorname{Off}\big(\mathbf{A}^{(j+1)}\big)$, are small enough to be declared zero.

Convergence Rate in the Single Matrix Case. In the single matrix case, each Jacobi update (2.61) involves $O(M)$ flops plus the cost of finding optimal indices $r = r(j)$ and $q = q(j)$. In the *classical Jacobi algorithm* for a single matrix, one targets to maximize the reduction of non-diagonal elements of $\mathbf{A}^{(j)}$, and r and q are chosen such that A_{rq}^2 is maximal. In this case, one can show that $\operatorname{Off}\big(\mathbf{A}^{(j)}\big) \le (1 - N^{-1})^j \operatorname{Off}\big(\mathbf{A}^{(0)}\big)$, $N = K(K-1)/2$, which implies a *linear convergence rate*. Moreover, the asymptotic convergence rate of the classical Jacobi algorithm is considerably better [see, e.g., Golub and Van Loan, 2013] : for j large enough there is a constant c such that

8. Note that the choice $\pi/4$, and not $\pi/2$, in the constraint is conventional, since the objective is $\pi/2$-periodic (it is quadratic in $\sin(\theta)$ and $\cos(\theta)$) and, in the convergence theory of the classical Jacobi iteration, it is critical that $|\theta| < \pi/4$ [Golub and Van Loan, 2013].

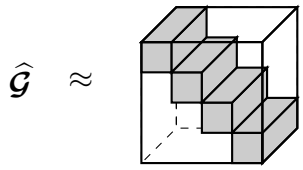


FIGURE 2-4 – The form of the core tensor of (2.50) approximated with OJD.

$\sqrt{\text{Off}\left(\mathbf{A}^{(j+N)}\right)} \leq c \cdot \text{Off}\left(\mathbf{A}^{(j)}\right)$, which implies a (local) *quadratic convergence rate*. Independently of the pivot r and q choice, the Jacobi algorithm converges to the global optimum.

Convergence Rate in the Case of Several Matrices. Bunse-Gerstner et al. [1993] analyzed the convergence properties of Algorithm 1 in the case of two matrices and proven the local quadratic convergence rate. They conjectured the global convergence properties of the algorithm in the two matrix case, which they confirmed with simulation experiments. However, no global convergence results are known in the multiple matrix case for Algorithm 1.

The Cyclic Order. The problem of the classical Jacobi algorithm is that the optimal index choice requires $O(M^2)$ flops and thus is more expensive than the cost of the Jacobi update. Therefore, in practice it is common to follow a *lexicographic* rule for the choice the indices r and q in the row-by-row and column-by-column fashion, which is known as the *cyclic Jacobi algorithm*. In this case, the linear and (locally) quadratic convergence rates are not applicable anymore, although the algorithm is still guaranteed to converge to the global optimum.

Sweep. It is common to refer to $K(K-1)/2$ consecutive iterations of any Jacobi-like algorithm as a *sweep*. In the case of the cyclic order, this refers to one pass over all elements of (symmetric) matrices.

Perturbation Analysis. Cardoso [1994a] provided first order perturbation analysis of joint diagonalizers $\mathbf{Q}$ in the case when matrices $\mathbf{A}_p$ are perturbed with the additive noise.

Relation to the CP Decomposition of Tensors. We motivated the OJD Algorithm 1 as a method for approximating the orthogonal symmetric CPD (2.50) or (2.51). The approximation comes from the fact that an approximated core tensor $\widehat{\boldsymbol{\mathcal{G}}}$ is not diagonal, since the matrices $\widehat{\mathbf{V}}\mathcal{A}\widehat{\mathbf{V}}^\top$ are not exactly diagonal. Moreover, even if these matrices were very close to diagonal, the core tensor $\widehat{\boldsymbol{\mathcal{G}}}$ would still have different from diagonal structure : all the fibers $\widehat{\mathcal{G}}_{kk:}$ are not guaranteed to be zero due to the contraction along the 3-rd dimension of this tensor (see Figure 2-4).

Tensor Orthogonal Diagonalization (COM2). Another Jacobi-like algorithm for approximately finding the orthogonal symmetric CPD (2.50) does not project this tensor onto vectors but instead works with the tensor directly [*COM2* ; Comon, 1994]. Instead of problem (2.59), it is formulated via the following optimization problem $\mathbf{Q}^\star = \arg\min_{\mathbf{Q}\in\mathcal{S}_K} \text{Off}(\boldsymbol{\mathcal{T}} \times_1 \mathbf{Q}^\top \times_2 \mathbf{Q}^\top \times_3 \mathbf{Q}^\top)$, where the objective is again the

sum of squared non-diagonal elements, i.e. $\text{Off}(\boldsymbol{\mathcal{T}}) = \sum_{m_1 \neq m_2 \neq m_3} \mathcal{T}^2_{m_1 m_2 m_3}$, and $\mathcal{S}_K$ is the Stiefel manifold. This algorithm directly extends the idea of iterative Jacobi updates.

It has been proved by Souloumiac and Cardoso [1993] that the asymptotical accuracy (for infinitesimal errors in the cumulant estimate) of COM2 and JADE are the same, when these algorithms are applied to the ICA cumulant tensors. By Chevalier [1995], an experimental comparison of these algorithms was performed, which indicates that both methods seem to have the same accuracy in the (under)complete case.

Tensor Power Method

Another algorithm for computing the orthogonal symmetric CP decomposition in (2.51) is the *tensor power method (TPM)* [Anandkumar et al., 2014, and references therein]. The idea behind the algorithm is to extend the matrix power method. Indeed, similarly to matrices, one can define eigenvalues and eigenvectors of a tensor [Lim, 2005, Qi, 2005] : a vector $\mathbf{u}$ that satisfy $\boldsymbol{\mathcal{T}}(\mathbf{I}, \mathbf{u}, \mathbf{u}) = \lambda \mathbf{u}$ is called an *eigenvector* of the tensor $\boldsymbol{\mathcal{T}}$ and λ is called its *eigenvalue*. Note that $\boldsymbol{\mathcal{T}}(\mathbf{I}, \mathbf{u}, \mathbf{u}) := \boldsymbol{\mathcal{T}} \times_1 \mathbf{I} \times_2 \mathbf{u} \times_3 \mathbf{u}$. It is straightforward to show that the vectors $\mathbf{v}_k$ and scalars λ_k in (2.51) are the eigenvectors and eigenvalues of $\boldsymbol{\mathcal{T}}$ respectively. Contrary to the matrix case, these are not the only eigenvalues and eigenvectors. For example, if $(\lambda_1, \mathbf{v}_1)$ and $(\lambda_2, \mathbf{v}_2)$ are two eigenpairs of $\boldsymbol{\mathcal{T}}$, then any vector $\mathbf{u} := (1/\lambda_1)\mathbf{v}_1 + (1/\lambda_2)\mathbf{v}_2$ is an eigenvector as well [Anandkumar et al., 2014]. However, one can define the so-called *robust eigenvectors* of a tensor $\boldsymbol{\mathcal{T}}$: if there exists an $\varepsilon > 0$ such that for all $\mathbf{u} \in \left\{\mathbf{v}' \in \mathbb{R}^K \ : \ \|\mathbf{v}' - \mathbf{v}\|_2 \leq \varepsilon\right\}$, repeated iteration of the map $\bar{\mathbf{v}} \mapsto \frac{\boldsymbol{\mathcal{T}}(\mathbf{I}, \bar{\mathbf{v}}, \bar{\mathbf{v}})}{\|\boldsymbol{\mathcal{T}}(\mathbf{I}, \bar{\mathbf{v}}, \bar{\mathbf{v}})\|_2}$ starting from $\mathbf{u}$ converges to $\mathbf{v}$, then $\mathbf{v}$ is called a robust eigenvector. If a tensor $\boldsymbol{\mathcal{T}}$ admits the decomposition in (2.51) then (a) the set of $\mathbf{u} \in \mathbb{R}^K$ which do not converge to some $\mathbf{v}_k$ under the repeated iteration from above has measure zero and (b) the set of robust eigenvectors of $\boldsymbol{\mathcal{T}}$ is equal to $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K\}$ (*fixed-point characterization*). Therefore, the orthogonal decomposition in (2.51) can be obtained with the tensor power method, which is summarized in Algorithm 2. Note that $\boldsymbol{\mathcal{T}}(\mathbf{u}, \mathbf{u}, \mathbf{u}) := \boldsymbol{\mathcal{T}} \times_1 \mathbf{u} \times_2 \mathbf{u} \times_3 \mathbf{u}$ and this version of the algorithm recovers the vectors $\mathbf{v}_k$ one by one through the *deflation* principle [see, e.g., Mackey, 2009, Comon and Jutten, 2010, Chapter 6]. This is also known in the literature as the *analysis view*. See also Wang et al. [2014] for an implementation and experimental comparison of the TPM for LDA.

Convergence Rate and Perturbation Analysis. When a tensor is orthogonally decomposable, the TPM guarantees the global recovery of the decomposition in (2.51) up to trivial indeterminancies at the quadratic convergence rate [Anandkumar et al., 2014]. However, in practice tensors are only approximately orthogonally decomposable (e.g., due to the finite sample noise). In this case, the TPM presented in Algorithm 2 is still able to approximately recover the decomposition (2.51). In particular, Anandkumar et al. [2014] provide perturbation analysis similar to the Wedin's perturbation theorem for singular vectors of matrices [Wedin, 1972] that bound the error of approximate decomposition in the case of additive noise. This analysis allows to set values L_{TPM} and N_{TPM} for the TPM Algorithm 2, which guarantee robust and fast

Algorithm 2 Robust Tensor Power Method

Initialize : $\boldsymbol{\mathcal{T}}^{(0)} \leftarrow \boldsymbol{\mathcal{T}}$
for deflation step $j = 1, 2, \ldots, K$ **do**
  for random restarts $\tau = 1, 2, \ldots, L_{TPM}$ **do**
    Draw $\mathbf{u}_0^{(j,\tau)} \in \mathbb{R}^K$ uniformly at random from the ℓ_2-unit sphere
    for power iterations $t = 1, 2, \ldots, N_{TPM}$ **do**
      Update $\mathbf{u}_t^{(j,\tau)} = \frac{\boldsymbol{\mathcal{T}}^{(j)}\left(\mathbf{I}, \mathbf{u}_{t-1}^{(j,\tau)}, \mathbf{u}_{t-1}^{(j,\tau)}\right)}{\left\|\boldsymbol{\mathcal{T}}^{(j)}\left(\mathbf{I}, \mathbf{u}_{t-1}^{(j,\tau)}, \mathbf{u}_{t-1}^{(j,\tau)}\right)\right\|_2}$
    end for
    Set $\mathbf{v}^{(j,\tau)} \leftarrow \mathbf{u}_t^{(j,\tau)}$
    Compute $\lambda^{(j,\tau)} = \frac{\boldsymbol{\mathcal{T}}^{(j)}\left(\mathbf{v}^{(j,\tau)}, \mathbf{v}^{(j,\tau)}, \mathbf{v}^{(j,\tau)}\right)}{\left\|\boldsymbol{\mathcal{T}}^{(j)}\left(\mathbf{v}^{(j,\tau)}, \mathbf{v}^{(j,\tau)}, \mathbf{v}^{(j,\tau)}\right)\right\|_2}$
  end for
  Find $\lambda^{(j)} \leftarrow \lambda^{(j,\tau_\star)}$ and $\mathbf{v}^{(j)}./q \leftarrow \mathbf{v}^{(j,\tau_\star)}$, where $\tau_\star := \arg\max_\tau \left\{\lambda^{(j,\tau)}\right\}_{\tau=1}^{L_{TPM}}$
  Perform deflation : $\boldsymbol{\mathcal{T}}^{(j)} \leftarrow \boldsymbol{\mathcal{T}}^{(j-1)} - \lambda^{(j)}[\mathbf{v}^{(j)}]^{\otimes 3}$
end for
Output : eigenvectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(K)}$ and eigenvalues $\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(K)}$

recovery of the decomposition. Intuitively, the meaning behind these numbers is the following. At every deflation step, L_{TPM} defines the number of random restarts which is necessary to estimate a vector $\mathbf{v}_k$ and, at every restart, while N_{TPM} defines the maximal number of tensor power update for every restart. If the number of power iterations t exceeds N_{TPM}, then the starting point was "bad" and such restart is not converging to any of vectors $\mathbf{v}_k$. If an initialization for a random restart was "good," the power iterations converge quite fast (quadratically) to one of vectors $\mathbf{v}_k$ and in this case the number of iterations t at convergence is below N_{TPM}. Hence, L_{TPM} and N_{TPM} have to be large enough for some restarts to converge, but not too large to avoid unnecessary computations. On the contrary, an advantage of the tensor power method over the orthogonal joint diagonalization algorithms is that the TPM admits a direct extension to the overcomplete regime [Anandkumar et al., 2015b].

The tensor power method is very similar [9] to such ICA methods as, e.g., FastICA [Hyvärinen, 1999], which is also a deflation-based algorithm in the orthogonal (prewhitened) subspace, but its variations depend on the choice of the contrast function. In general, an important benefit of such approach is, under ideal conditions, the absence of spurious local extrema and global convergence [Hyvärinen, 1999, Papadias, 2000, Anandkumar et al., 2014].

9. FastICA with the cubic contrast is very close to the TPM for ICA cumulants, but the algorithms are not exactly equivalent.

2.3.2 Algorithms for Non-Orthogonal Non-Symmetric CPD

Non-Orthogonal Non-Symmetric CPD. The problem is as follows (see also Section 2.1.2). Given a non-symmetric $M_1 \times M_2 \times M_3$ tensor $\boldsymbol{\mathcal{T}}$, the goal is to approximate this tensor as

$$\boldsymbol{\mathcal{T}} \approx \boldsymbol{\mathcal{G}} \times_1 \mathbf{V}^{(1)\top} \times_2 \mathbf{V}^{(2)\top} \times_3 \mathbf{V}^{(3)\top}, \tag{2.63}$$

where the $K \times K \times K$ core tensor $\boldsymbol{\mathcal{G}}$ is assumed to be diagonal with non-zero diagonal elements, but the matrices $\mathbf{V}^{(s)} \in \mathbb{R}^{M_s \times K}$, for $s = 1, 2, 3$, do not have to be all the same and are not assumed to be orthogonal as opposed to the orthogonal symmetric case (2.50) (see also Section 2.1.2). An equivalent vector representation of (2.63) is :

$$\boldsymbol{\mathcal{T}} \approx \sum_{k=1}^{K} g_k \, \mathbf{v}_k^{(1)} \otimes \mathbf{v}_k^{(2)} \otimes \mathbf{v}_k^{(3)}, \tag{2.64}$$

where $g_k = \mathcal{G}_{kkk}$ are again non-zero and the vectors $\mathbf{v}_k^{(s)}$ are the columns of the matrix $\mathbf{V}^{(s)}$, for $s = 1, 2, 3$ and are not assumed to be orthogonal. Note that we assume the number K of the CP factors in (2.64) known and fixed; we discuss the choice of K in Section 4.5. We also assume that $K < \min(M_1, M_2)$ and the matrices $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$ have a full column rank. [10]

In this section, we show that the approximation problem in (2.63) or (2.64) can be reduced to the so-called *non-orthogonal joint diagonalization by similarity* and describe Jacobi-like algorithms for this problem [Fu and Gao, 2006, Iferroudjene et al., 2009, Luciani and Albera, 2010].

Non-Orthogonal Joint Diagonalization by Similarity

Target Matrices. Let us contract both sides of the expression (2.64) with a vector $\mathbf{a}_0$ (see Equation (2.52) for the definition) :

$$\boldsymbol{\mathcal{T}}(\mathbf{a}_0) \approx \mathbf{V}^{(1)} \boldsymbol{\Lambda}_0 \mathbf{V}^{(2)\top},$$

where the $K \times K$ matrix $\boldsymbol{\Lambda}_0$ is diagonal with a diagonal element $\Lambda_{0,kk} := g_k \langle \mathbf{v}_k^{(3)}, \mathbf{a}_0 \rangle$. Assuming that the matrix $\boldsymbol{\mathcal{T}}(\mathbf{a}_1)$ has rank F (if the vector $\mathbf{a}_1$ is picked uniformly at random from the unit ℓ_2-sphere, this is true with probability 1) and *diagonalizable*, one can then construct matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ that $\mathbf{W}_1 \boldsymbol{\mathcal{T}}(\mathbf{a}_0) \mathbf{W}_2^\top = \mathbf{I}$. Note that since the decomposition in (2.64) is only identifiable up to permutation and scaling we can rescale the matrices $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$ in $\boldsymbol{\mathcal{T}}(\mathbf{a}_0)$ such that $\boldsymbol{\mathcal{T}}(\mathbf{a}_0) = \mathbf{V}^{(1)} \mathbf{V}^{(2)\top}$, without loss of generality. This means that matrices $\mathbf{W}_1 \mathbf{V}_1$ and $\mathbf{W}_2 \mathbf{V}_2$ are invertible and

$$\mathbf{W}_1 \mathbf{V}_1 = (\mathbf{W}_2 \mathbf{V}_2)^{-\top}.$$

Let us denote $\mathbf{V} := \mathbf{W}_1 \mathbf{V}_1$, then it immediately follows from the expression above that $(\mathbf{W}_2 \mathbf{V}_2)^\top = \mathbf{V}^{-1}$. This essentially leads to the following set $\mathcal{A}$ of *target ma-*

10. This corresponds to the (under)complete case of Chapter 4.

trices :

$$\mathcal{A} = \{\mathbf{A}_1,\ \mathbf{A}_2,\ \ldots,\ \mathbf{A}_P\}\,, \tag{2.65}$$

where the $K \times K$ square matrices $\mathbf{A}_p := \mathbf{W}_1 \boldsymbol{\mathcal{T}}(\mathbf{a}_p)\mathbf{W}_2^\top$, for $p \in [P]$, are all expected to be approximately in the form $\mathbf{A}_p \approx \mathbf{V}\boldsymbol{\Lambda}_p\mathbf{V}^{-1}$.

Non-Orthogonal Joint Diagonalization by Similarity Problem. This leads to the problem of approximate *non-orthogonal joint diagonalization (NOJD) by similarity* : find a matrix $\mathbf{Q} \in \mathbb{R}^{K\times K}$ such that the (not necessary normal [11]) matrices $\mathcal{A}$ are (jointly) *as diagonal as possible* when subject to the *similarity* transformation with $\mathbf{Q}$:

$$\mathbf{Q}^{-1}\mathcal{A}\mathbf{Q} = \left\{\mathbf{Q}^{-1}\mathbf{A}_1\mathbf{Q},\quad \mathbf{Q}^{-1}\mathbf{A}_2\mathbf{Q},\quad \ldots,\quad \mathbf{Q}^{-1}\mathbf{A}_P\mathbf{Q}\right\}.$$

Note that in general this problem is different from the so-called problem of *non-negative joint diagonalization by congruence* [see e.g. Afsari, 2006, 2008, Souloumiac, 2009a], where the inverse of $\mathbf{Q}$ is replaced by the transpose. The two problems are only equivalent in the case when $\mathbf{Q}$ is orthogonal.

Jacobi-like algorithms [Fu and Gao, 2006, Iferroudjene et al., 2009, Luciani and Albera, 2010] for this problem extend both the orthogonal joint diagonalization algorithm from Section 2.3.1 and Jacobi-like algorithms for the eigendecomposition of non-normal matrices [see, e.g., Ruhe, 1968, Eberlein, 1962].

The Shear Matrix. Similar to the Jacobi rotation matrix (2.60), the *shear matrix* is the $K \times K$ matrix $\mathbf{S}(r, q, y)$ defined as :

$$\mathbf{S}(r,q,y) = \begin{array}{c} \\ r \\ \\ \\ q \\ \\ \end{array}\!\!\begin{array}{c} \begin{array}{ccccccc} & & r & & q & & \end{array} \\ \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \cosh(y) & \cdots & \sinh(y) & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & \sinh(y) & \cdots & \cosh(y) & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \end{array} \tag{2.66}$$

where $y \in \mathbb{R}$ is called the *shear parameter*. A shear matrix $\mathbf{S}(r, q, y)$ is non-orthogonal and corresponds to a skew transformation in the (r, q)-plane when applied to a vector or matrix. The update $\mathbf{S}(r, q, y)^\top\mathbf{A}$ affects only two rows (r-th and q-th) of $\mathbf{A}$. Likewise, the update $\mathbf{A}\mathbf{S}(r, q, y)$ affects just two columns (r-th and q-th) of $\mathbf{A}$.

Non-Orthogonal Joint Diagonalization by Similarity Algorithms. The main idea behind all the NOJD by similarity algorithms is to construct a sequence of matrices $\mathcal{A}^{(0)}$, $\mathcal{A}^{(1)}$, $\mathcal{A}^{(2)}$, $\ldots$, $\mathcal{A}^{(J)}$:

$$\mathcal{A}^{(j)} = \left\{\mathbf{A}_1^{(j)},\ \mathbf{A}_2^{(j)},\ \ldots,\ \mathbf{A}_P^{(j)}\right\},$$

11. Recall that a matrix $\mathbf{A}$ is normal if $\mathbf{A}\mathbf{A}^\top = \mathbf{A}^\top\mathbf{A}$.

such that $\mathcal{A}^{(0)} = \mathcal{A}$ and at every iteration j, the matrices are constructed by consecutive similarity transformations first with the *optimal shear matrix* $\mathbf{S}^{(j)} = \mathbf{S}\big(r, q, y^{(j)}\big)$ and then with the *optimal Jacobi rotation* $\mathbf{G}^{(j)} = \mathbf{G}\big(r, q, \theta^{(j)}\big)$:

$$\mathcal{A}^{(j)} = \mathbf{G}^{(j)\top}\mathbf{S}^{(j)-1}\mathcal{A}^{(j-1)}\mathbf{S}^{(j)}\mathbf{G}^{(j)}, \tag{2.67}$$

where the indices $r = r(j)$ and $q = q(j)$ are chosen in accordance with some rule (see below) and we discuss below the choice of the optimal shear parameter $y^{(j)}$ and Jacobi angle $\theta^{(j)}$. Note that in expression (2.67), the matrix $\mathbf{S}^{(j)-1}$ denotes the inverse of the optimal shear matrix $\mathbf{S}^{(j)}$.

For convenience, let us separate the similarity transformations with a shear and Jacobi rotation matrices. Given matrices $\mathcal{A}^{(j-1)}$ after the $(j-1)$-th iteration, define :

$$\mathcal{A}'(y) = \mathbf{S}^{-1}(r, q, y)\, \mathcal{A}^{(j-1)}\, \mathbf{S}(r, q, y), \qquad \mathcal{A}'^{(j)} = \mathcal{A}'\big(y^{(j)}\big), \tag{2.68}$$

$$\mathcal{A}(\theta) = \mathcal{A}''(\theta) = \mathbf{G}^{\top}(r, q, \theta)\, \mathcal{A}'^{(j)}\, \mathbf{G}(r, q, \theta), \qquad \mathcal{A}^{(j)} = \mathcal{A}\big(\theta^{(j)}\big). \tag{2.69}$$

Considering each of these transformations separately at each iteration of the algorithm leads to two consecutive updates : the *shear* and *Jacobi* updates.

The Shear Update. The shear update is defined as such shear transformation from (2.68) that the respective shear parameter y is optimal in some way. In fact, the algorithms by Fu and Gao [2006], Iferroudjene et al. [2009], Luciani and Albera [2010] differ essentially in the way this optimality of the shear transformation is defined. The sh-rt algorithm [Fu and Gao, 2006] and the JUST algorithm [Luciani and Albera, 2010] define the optimal shear parameter as :

$$y^{(j)} = \arg\min_{y \in \mathbb{R}} \sum_{p=1}^{P} \left\| \mathbf{A}'_p(y) \right\|_F^2, \tag{2.70}$$

and propose some heuristics to approximate this optimal shear parameter since a closed form solution to this problem can not be easily represented as opposed to the Jacobi update case (see Section 2.3.1). Note that the Frobenius norm $\|\mathbf{A}\|_F^2$ appearing in the objective is called the *normality measure* (see below). The JUST algorithm formulates the optimal shear parameter as $y^{(j)} = \arg\min_{y \in \mathbb{R}} \sum_{p=1}^{P} \mathrm{Off}(\mathbf{A}'_p(y))$, where the sum of squared off-diagonal elements $\mathrm{Off}(\cdot)$ is defined in (2.58), and proposes complex expressions for a closed form solution. Note that in both cases the objectives tend to infinitely whenever $y \to \pm\infty$. Therefore, one can rewrite these optimization problems with the constraint $y \in [y_L, y_R]$ for y_L and y_R sufficiently large. This allows to compute the optimal solution of (2.70) with an exhaustive search approach. We compared experimentally all four approaches to the NOJD problem and observed that although the convergence properties of the algorithms can slightly differ, the difference in the accuracy of the obtained solutions can be barely noticed.

The Jacobi Update. The Jacobi update (2.69) is exactly equivalent to the Jacobi update of the orthogonal joint diagonalization algorithm from Section 2.3.1 applied

Algorithm 3 Non-Orthogonal Joint Diagonalization (NOJD) by Similarity

Initialize : $\mathcal{A}^{(0)} \leftarrow \mathcal{A}$ and $\mathbf{Q}^{(0)} \leftarrow \mathbf{I}_M$ and iterations $j \leftarrow 0$
for sweeps $1, 2, \ldots$ **do**
  for $r = 1, \ldots, K-1$ **do**
    for $q = r+1, \ldots, K$ **do**
      Increase $j \leftarrow j + 1$
      Compute or approximate the optimal shear parameter $y^{(j)}$
      Compute the optimal Jacobi angle $\theta^{(j)}$ (closed form)
      Update $\mathbf{Q}^{(j)} \leftarrow \mathbf{Q}^{(j-1)}\mathbf{S}^{(j)}\mathbf{G}^{(j)}$
      Update $\mathcal{A}^{(j)} \leftarrow \mathbf{G}^{(j)\top}\mathbf{S}^{(j)-1}\mathcal{A}^{(j-1)}\mathbf{S}^{(j)}\mathbf{G}^{(j)}$
    end for
  end for
end for
Output : $\mathbf{Q}^{(j)}$

to the matrices $\mathcal{A}'^{(j)}$ and therefore the description is omitted. Note that although Fu and Gao [2006], Iferroudjene et al. [2009] propose a different approach from Cardoso and Souloumiac [1996] to obtain a closed form solution for the optimal Jacobi angle, one can show that the solutions are equivalent. The NOJD by similarity algorithms are summarized in Algorithm 3.

Convergence. To the best of our knowledge, no theoretical analysis of the NOJD by similarity algorithms [Fu and Gao, 2006, Iferroudjene et al., 2009, Luciani and Albera, 2010] is available in the literature, except for the single matrix case when they boil down to the (non-normal or non-symmetric) eigendecomposition [Eberlein, 1962, Ruhe, 1968]. In the latter case, the algorithms converge globally at a (locally) quadratic convergence rate given some sophisticated rule for the order of the indices r and q. A quadratic convergence rate is conjectured in the multiple matrix case, but has not been proved yet. In practice, we always use the lexicographical order for the choice of the indices r and q.

Perturbation Analysis. To the best of our knowledge, no perturbation analysis of the NOJD by similarity algorithms is available in the literature. Potential extension of the results for the NOJD by congruence algorithms [Afsari, 2008, Kuleshov et al., 2015b,a] could be of interest.

Intuition Behind the Algorithms. The *Schur decomposition* says that for any diagonalizable matrix $\mathbf{A}$ there exists an orthogonal matrix $\mathbf{Q}$ that $\mathbf{Q}^\top\mathbf{A}\mathbf{Q} = \mathbf{\Lambda} + \mathbf{N}$, where the matrix $\mathbf{\Lambda}$ is diagonal with the eigenvalues of $\mathbf{A}$ on the diagonal and the matrix $\mathbf{N}$ is strictly upper triangular [Golub and Van Loan, 2013, Chapter 7] . Moreover, for any non-singular matrix $\mathbf{M}$ it holds that $\inf_{\mathbf{M}} \|\mathbf{M}^{-1}\mathbf{A}\mathbf{M}\|_F^2 = \|\mathbf{\Lambda}\|_F^2$, and, therefore, a diagonalized version of the matrix $\mathbf{A}$ must have the smallest Frobenius norm [Ruhe, 1968]. Since the Jacobi transformation $\mathbf{G}(r, q, \theta)^\top\mathbf{A}\mathbf{G}(r, q, \theta)$ does not change the Frobenius norm, the Frobenius norm of $\|\mathbf{M}^{-1}\mathbf{A}\mathbf{M}\|_F^2$ can only be minimized with a shear transformation, i.e. when $\mathbf{M}$ is equal to (a multiplication of)

shear matrices. This explains why a shear transformation is necessary for the NOJD-type algorithms. If a matrix is normal, the strictly upper triangular matrix $\mathbf{N}$ in its Schur decomposition vanishes and the matrix $\mathbf{M}$ contains the eigen vectors of $\mathbf{A}$ in its columns. This explains why the squared Frobenius norm is called the *normality measure* : decreasing the normality measure of a matrix "moves" this matrix closer to a normal matrix ; this also explains why a normal diagonalizable matrix can be diagonalized by an orthogonal matrix, such as a Jacobi rotation matrix, which preserves the Frobenius norm. Hence, the optimal shear transformation by minimizing the normality measure decreases the deviation from normality and then the optimal Jacobi transformation by minimizing the diagonality measure (the sum of squared off-diagonal elements) decreases the deviation from diagonality.

2.4 Latent Linear Models : Estimation and Inference

In this thesis, we refer to as *estimation* the process of estimating model parameters, while we use the term *inference* for the process of inferring the latent variables given an observation, which is a more standard terminology in the frequentist literature. For most probabilistic models of practical interest, the estimation and inference are intractable : for example, this is the case of almost all the models from Chapter 1, with the exception of principal and canonical correlation analyses. For example, it was shown that the maximum a posteriori (MAP) based inference for LDA is NP-hard if the effective number of topics per document is large [Sontag and Roy, 2011]. Moreover, e.g., the maximum likelihood-based estimation of the topic matrix in topic models is also NP-hard [Arora et al., 2012]. Therefore, one is interested in approximation methods. Below we briefly recall some of them : the expectation maximization algorithm-based methods as well as some moment matching-based methods.

2.4.1 The Expectation Maximization Algorithm

The EM Algorithm

The *expectation-maximization (EM)* algorithm [Dempster et al., 1977, McLachlan and Krishnan, 2007] is a powerful method for finding maximum likelihood solutions for models with latent or missing variables. In general, let $\mathbf{x}$ denote all observed variables, $\mathbf{z}$ denote all latent variables, and $\boldsymbol{\theta}$ denote *all* parameters of the model, and the likelihood function is given by

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) d\mathbf{z},$$

and the goal is to maximize the log of the likelihood $\log[p(\mathbf{x}|\boldsymbol{\theta})]$ (the discussion is similar to the case of discrete latent variables). The EM algorithm is of interest when direct maximization of the likelihood $p(\mathbf{x}|\boldsymbol{\theta})$ is difficult, but optimization of the complete-data likelihood $p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ is significantly easier (e.g., the case of the GMMs or pLSI model).

For any distribution $q(\mathbf{z})$ over latent variables, such that $q(\mathbf{z}) > 0$ if $p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) > 0$, it holds by the *Jensen's inequality* that

$$\mathcal{L}(\boldsymbol{\theta}) := \log \int q(\mathbf{z}) \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} dz \geq \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} := \mathcal{F}(q, \boldsymbol{\theta}),$$

where the functional $\mathcal{F}(q, \boldsymbol{\theta})$ on the RHS is a lower bound. The EM algorithm then maximizes this lower bound by the alternating maximization with respect to the distribution $q(\mathbf{z})$ over latent variables, keeping the parameters $\boldsymbol{\theta}^{old}$ fixed (the *E step*), and then by the maximization of the parameters $\boldsymbol{\theta}$, keeping the distribution $q^{old}(\mathbf{z})$ over latent variables fixed (the *M step*). One can show that such procedure increases (or does not change) the log-likelihood function $\mathcal{L}(\boldsymbol{\theta})$ at every iteration. However, even if the maximization problems at the E and M steps have closed form solutions, the overall procedure is not guaranteed to converge to a local optimum (although it usually does), but it is guaranteed to converge to a stationary point. In general, however, the EM algorithm does not have any global convergence guarantees.

Variational Inference

Variational inference can be used for the approximation of the E step of the EM algorithm for the models where the E step is intractable. In such case, the EM algorithm is referred to as the *variational EM algorithm*. The idea of variational inference [Jordan, 1999, Jaakkola, 2001] is to approximate the distribution of the latent variables by maximizing the lower bound $\mathcal{F}(q, \boldsymbol{\theta})$ over some class of functions (probability distributions) keeping the parameters $\boldsymbol{\theta}$ fixed. This has a straightforward probabilistic meaning since $\mathcal{L}(\boldsymbol{\theta}) - \mathcal{F}(q, \boldsymbol{\theta}) = KL(q||p)$, where $KL(q||p)$ is the KL divergence between the original distribution p of the latent sources and q is the approximate one. Therefore, maximizing the lower bound $\mathcal{L}(\boldsymbol{\theta})$ is equivalent to minimizing the KL-divergence $KL(q||p)$, which finds a closest to q distribution p from a given class of distributions. The class of distributions is chosen to be some *tractable* class, such as distributions that *factorize*, i.e. $q(\mathbf{z}) = \prod_k q_k(z_k)$, which gives the so-called *variational mean field inference* procedure. For more details, see, e.g., Bishop [2006], Murphy [2012].

Sampling Methods

Using sampling methods is another way to approximate the E step of the EM algorithm for models where the E step does not have a closed form solution. *Markov chain Monte Carlo (MCMC)* [Gilks et al., 1995] is a class of algorithms for sampling from complicated probability distributions based on constructing a Markov chain that converges to a desired distribution and then sampling from this Markov chain to approximate the objective of the M step. Each state of the chain is an assignment of values to the variables being sampled and, in practice, the state of the chain after a number of steps is used as a sample of the desired distribution. A popular sampling method is *Gibbs sampling*, which requires all the conditional distributions of the target distribution to be sampled exactly. For example, Griffiths [2002] describe such

Gibbs sampling procedure for latent Dirichlet allocation. Since the Markov chain is only known to converge to a desired distribution asymptotically, the practical implementation with a finite number of steps (which is often difficult to choose) does not have any guarantees on the obtained solution. For more details, see, e.g., Bishop [2006], Murphy [2012].

2.4.2 Moment Matching Techniques

In this section, we outline the main idea of the method of moments for the estimation of the linear transformation matrix in latent linear models on the example of the LDA model using the diagonal (symmetric CPD) form of the LDA cumulants (see Section 2.2.3). An advantage of this approach over the variational inference and sampling approaches described in Section 2.4.1 are theoretical guarantees on the quality of the recovered topic matrix [Anandkumar et al., 2012a, 2014]. Note that other algorithms for the estimation and inference with theoretical guarantees are available [Arora et al., 2012, 2013, 2015].

The algorithms described in this section are applicable only in the case when the columns of the topic matrix are linearly independent, which implies that $K \leq M$, i.e. the (under)complete case.

This class of algorithms is well known in the signal processing and machine learning literature (some of related references can be found in Section 3.2). In the ICA literature, these algorithms are known as cumulant-based algorithms with the prewhitening [see, e.g., Cardoso, 1989, 1990, Cardoso and Souloumiac, 1993, De Lathauwer, 2006, Comon and Jutten, 2010].

The common idea behind all these methods is to use the diagonal structure of the population statistics of a model for the estimation of the topic matrix. The problem is that the second-order information is often not sufficient for recovery (unless some additional assumptions, such as sparsity of the topic matrix, are made). This motivates to use the third-order [12] jointly with the second-order information, which can be seen as a problem of joint diagonalization of the matrix $\mathbf{S} := \mathbf{S}^{LDA}$ and tensor $\boldsymbol{\mathcal{T}} := \boldsymbol{\mathcal{T}}^{LDA}$ both in the symmetric CP form with the topic matrix in place of the factor matrix, i.e.

$$\mathbf{S} = \widetilde{\mathbf{D}}\widetilde{\mathbf{D}}^\top,$$
$$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{G}} \times_1 \widetilde{\mathbf{D}}^\top \times_2 \widetilde{\mathbf{D}}^\top \times_3 \widetilde{\mathbf{D}}^\top,$$

the core tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{K \times K \times K}$ is diagonal and this form of the matrix $\mathbf{S}$ is without loss of generality due to the permutation and scaling unidentifiability of the model (however, the columns of the "topic" matrix $\widetilde{\mathbf{D}}$ do not have to sum to one anymore). Let $\mathbf{A} \in \mathbb{R}^{M \times K}$ be a matrix such that $\mathbf{A}\mathbf{S}\mathbf{A}^\top$ and $\boldsymbol{\mathcal{T}} \times_1 \mathbf{A}^\top \times_2 \mathbf{A}^\top \times_3 \mathbf{A}^\top$ are diagonal.

12. Sometimes higher order information is used. For instance, when the sources in the ICA model are expected to have symmetric priors, the fourth-order information is used since the odd-order cumulants are zero for symmetric distributions.

Then, up to permutation and scaling, $\mathbf{A}\widetilde{\mathbf{D}} = \mathbf{I}$ and one can recover $\widetilde{\mathbf{D}}$ as $\widetilde{\mathbf{D}} = \mathbf{A}^{\dagger}$ (see more details in Section 3.4).

Prewhitening. The goal of the *prewhitening* procedure is to find a matrix $\mathbf{W} \in \mathbb{R}^{K \times M}$, which is called the *whitening* matrix, such that

$$\mathbf{W}\mathbf{S}\mathbf{W}^{\top} = \mathbf{I}. \tag{2.71}$$

This procedure takes its name from the fact that in the ICA context the matrix $\mathbf{S}$ is just a covariance matrix $\operatorname{cov}(\mathbf{x})$. Hence, the prewhitening transformation (2.71) corresponds to such linear transformation of the observation vector $\mathbf{x}$ into another vector $\mathbf{x} := \mathbf{W}\mathbf{x}$ with unit covariance $\operatorname{cov}(\mathbf{z}) = \mathbf{I}$.

A whitening matrix is not uniquely defined. Indeed, let $\mathbf{Q}$ be an arbitrary orthogonal matrix of the appropriate size and $\mathbf{W}$ be a whitening matrix. Then the matrix $\widetilde{\mathbf{W}}$ obtained by the right multiplication of the whitening matrix $\mathbf{W}$ with this orthogonal matrix $\mathbf{Q}$, i.e. $\widetilde{\mathbf{W}} := \mathbf{W}\mathbf{Q}$, is still a whitening matrix since it preserves the equality in (2.71) : $\widetilde{\mathbf{W}}\,\mathbf{S}\,\widetilde{\mathbf{W}}^{\top} = \mathbf{Q}\,\mathbf{W}\,\mathbf{S}\,\mathbf{W}^{\top}\mathbf{Q}^{\top} = \mathbf{Q}\mathbf{Q}^{\top} = \mathbf{I}$.

A whitening matrix can be computed via the eigendecomposition of $\mathbf{S}$. Let $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\top}$, then $\mathbf{W} = \left[\mathbf{\Lambda}_{1:K}^{1/2}\right]^{\dagger}\mathbf{U}_{1:K}^{\dagger}$, where $\mathbf{\Lambda}_{1:K}$ is the diagonal matrix formed by the first K largest eigenvalues and $\mathbf{U}_{1:K}$ is the matrix which contains the respective eigenvectors in columns.

The "Correct" Rotation. When $\mathbf{W}$ is found, the matrix $\mathbf{A}$ can be formed as $\mathbf{A} = \mathbf{Q}\mathbf{W}$, for some "correct" orthogonal matrix $\mathbf{Q}$. Therefore, one is left with the problem of finding such orthogonal matrix $\mathbf{Q}$ from the third-order information. For that, let us transform the tensor $\boldsymbol{\mathcal{T}}$ with the found whitening matrix along all modes, this gives the $K \times K \times K$ tensor $\overline{\boldsymbol{\mathcal{T}}} := \boldsymbol{\mathcal{T}} \times_1 \mathbf{W}^{\top} \times_2 \mathbf{W}^{\top} \times_3 \mathbf{W}^{\top}$. In the ideal population case, this tensor $\overline{\boldsymbol{\mathcal{T}}}$ is an orthogonally decomposable tensor (see Section 2.3.1 and the decomposition 2.50). In the practical finite sample case, such decomposition holds approximately. In any case, finding the factor matrix with one of the algorithms described in Section 2.3.1 boils down to finding a "correct" orthogonal matrix $\mathbf{Q}$. More details on such algorithms can be found in Section 3.4.

Guarantees. Hence, in the ideal population case, such two-step procedure with, e.g., the ED-based algorithm or the tensor power method used as the algorithm for the second step, guarantees the global solution the estimation problem in LDA. In the practical finite sample case, lower bounds on the quality of the recovery are available [Anandkumar et al., 2012a, 2014].

Non-Orthogonal Approaches. Note that the prewhitening step introduces some error and is not necessary the best way to solve the estimation problem [Cardoso, 1994b, De Lathauwer et al., 2005, Souloumiac, 2009b] given finite sample estimates of the matrix $\mathbf{S}$ and tensor $\boldsymbol{\mathcal{T}}$ in the symmetric CP form. Another class of algorithms is the so-called *non-orthogonal joint diagonalization algorithms by congruence*, which contract the tensor $\boldsymbol{\mathcal{T}}$ with random vectors without the preliminary prewhitening and then jointly diagonalize the obtained matrices but with a non-orthogonal matrix [see,

e.g., Afsari, 2006, 2008, Souloumiac, 2009a]. This problem is more difficult than the OJD problem described in Section 2.3.1.

Chapitre 3

Moment Matching-Based Estimation in Topic Models

Abstract

In this chapter, we draw explicit links between latent Dirichlet allocation (LDA ; see Section 1.2.4) and discrete versions of independent component analysis (ICA ; see Section 1.1.4). Using this strong connection between LDA and ICA, we introduce a novel *semiparametric* topic model, which we call *discrete independent component analysis (DICA)*. In the DICA model, no assumption on latent sources is made (although, non-Gaussian assumption is needed for identifiability ; see Chapter 4) and it is not necessary to know the distributions of latent sources in order to perform the estimation, which increases the expressive power of the model.

While early work has focused on graphical-model approximate inference techniques such as variational inference or Gibbs sampling (see Section 2.4), tensor-based moment matching techniques have recently emerged as strong competitors due to their computational speed and theoretical guarantees [Anandkumar et al., 2012a, 2013a, 2014, 2015a]. We show that similar to the higher-order statistics closely related to moments of the LDA model (see Section 2.2.3), the population higher-order statistics closely related to cumulants of the discrete ICA model can be represented as tensors in the form of symmetric non-negative canonical polyadic decomposition (see Section 2.1.2). This allows us to reuse numerous techniques from the ICA literature to develop novel tensor-based algorithms based on joint diagonalization for the estimation in topic models, showing improvement over its predecessors the spectral algorithm [Anandkumar et al., 2012a, 2015a] and the tensor power method [Anandkumar et al., 2014], which nevertheless has strong theoretical guarantees. We also prove that in some practical scenarios the new cumulant based DICA tensors have improved sample complexity over the LDA based moment tensors. The content of this chapter was previously published as Podosinnikova et al. [2015].

3.1 Contributions

Below we outline the contributions of this chapter.

- In Section 3.3, we introduce a novel *semiparametric* topic model—*discrete ICA*—which is closely related to both latent Dirichlet allocation and independent analysis, but has higher expressive power due to its semiparametric nature, where the distributions of the latent sources are not specified.
- In Section 3.3.2, we derive *novel cumulant-based tensors* for the gamma-Poisson and discrete ICA models and show that the population versions of these tensors takes the form of the symmetric non-negative canonical polyadic decomposition (a.k.a. *diagonal form*). We also present *sample complexity* results for natural sample estimators of these tensors.
- In Section 3.4, we propose a *novel algorithm for the estimation in topic models*, which is based on orthogonal joined diagonalization of contractions of DICA cumulant-based tensors after prewhitening. Other algorithms, such as the *tensor power method* and ED-based algorithm (a.k.a. *spectral method*) are applicable for the estimation as well. Since the gamma-Poisson topic model is a special case of the DICA model, the algorithms also apply to the gamma-Poisson model without any modification.
- In Section 3.5, we perform an *extensive experimental comparison* of the diagonalization algorithms (orthogonal joint diagonalization, the tensor power method, and the spectral method) for the estimation in the LDA and DICA models as well as, for the first time [1] to the best of our knowledge, compare these algorithms with the variational inference-based algorithms.

3.2 Related Work

The algorithms proposed in this chapter are closely related to both recent learning algorithms for latent Dirichlet allocation (LDA) [Anandkumar et al., 2012a, 2013a, 2015a, 2014, Arabshahi and Anandkumar, 2016] as well as orthogonal joint diagonalization type ICA algorithms [Bunse-Gerstner et al., 1993, Cardoso and Souloumiac, 1993, 1996, Cardoso, 1999, De Lathauwer, 2010]. Note also that a scalable implementation of the tensor power method for LDA [Anandkumar et al., 2014] is proposed by Wang et al. [2014]. Such algorithms often come with theoretical guarantees as opposed to the variational inference and sampling methods.

Another class of topic modeling algorithms with theoretical guarantees is based on the matrix factorization point of view [Arora et al., 2012, 2013, 2015]. In this case, the key assumption is the presence of *anchor words* in topics, i.e. words that appear mostly in this topic, which is related to sparsity. Similar to discrete ICA introduced in this chapter, they do not make any assumptions on the topic intensity distribution,

1. Some of these algorithms were previously compared with the Gibbs sampling-based methods by Wang et al. [2014], but our experiments are more detailed.

but the assumptions on the topic matrix are somewhat stronger than the ones in this chapter.

Another related work is by Arabshahi and Anandkumar [2016], where a new class of topic models, called *latent normalized infinitely divisible topic models*, is introduced. They prove that moment-based higher-order statistics of these topic models are tensors in the CP form and the tensor power method can be used for the estimation. This class of models is a direct extension of LDA where the topic intensities are modeled as a *normalized infinitely divisible (NID)* random vector [Favaro and Hadjicharalambous, 2011, Mangili and Benavoli, 2015], which is formed by normalizing independent positive infinitely divisible variables. The LDA model is a special case of such class of models. Moreover, some models from this class allow to model both positive and negative correlations among topics, thus, being more flexible than correlated topic models or pachinko allocation [Li and McCallum, 2006]. Note that diagonalization algorithms discussed in this chapter are also applicable for the estimation in latent NID topic models. As opposed to the DICA model, latent NID topic models require specification of the prior for topic intensities, while DICA is a semiparametric model, where the distribution is left unspecified.

3.3 Discrete ICA

In Section 1.2, we outlined latent linear models for count data, which include topic models and latent Dirichlet allocation in particular. We recalled the equivalence of the LDA model in the standard tokens formulation (1.16) to the counts formulation (1.17), which is also known as the multinomial PCA or Dirichlet-multinomial models. We also saw that these models are examples of the *admixture* model (1.12). In this section, we show that under mild assumptions the LDA model is equivalent to the gamma-Poisson model, which motivates us to introduce a new *semiparametric* model of *discrete ICA* in Section 3.3.1. This model is designed for working with non-negative discrete data and is able to adjust to different prior distributions on latent factors, hence having higher expressive power. We also show (in Section 3.3.2) that population higher-order cumulant-based statistics of this DICA model admit a representation in the form of the symmetric non-negative CP decomposition with the topic vectors as factors, which further allows to develop fast and efficient estimation algorithms (in Section 3.4). In Section 3.3.3, we present some sample complexity results for the DICA cumulant-based tensors.

3.3.1 Topic Models are PCA for Count Data

Latent Dirichlet Allocation [Blei et al., 2003] is a generative probabilistic model for discrete data such as text corpora. In accordance with this model, a document is modeled as an *admixture* over the vocabulary of M words with K latent topics. Specifically, the latent Dirichlet variable $\boldsymbol{\theta}$ (a.k.a. the vector of topic intensities) represents the topic mixture proportion over K topics for a document. The variable $\boldsymbol{\theta}$ takes values in the $(K-1)$-probability simplex $\boldsymbol{\Delta}_K$. Given $\boldsymbol{\theta}$, the topic state vector

$\mathbf{z}_\ell|\boldsymbol{\theta}$ of the ℓ-th token of this document is drawn from the discrete distribution with probability vector $\boldsymbol{\theta}$. The ℓ-th token $\mathbf{w}_\ell|\mathbf{z}_\ell, \boldsymbol{\theta}$ is then sampled from the discrete distribution with probability vector $\mathbf{d}_{\mathbf{z}_\ell}$, which stands for the k-th *topic* $\mathbf{d}_k$ where k is such that $[\mathbf{z}_\ell]_k = 1$. Each topic is a vector of probabilities over the words from the vocabulary subject to the probability simplex constraint, i.e., $\mathbf{d}_k \in \boldsymbol{\Delta}_M$ for all k. This generative process [2] of a document is summarized as

$$\begin{aligned} \big(L &\sim \text{Poisson}(\lambda)\big), \\ \boldsymbol{\theta} &\sim \text{Dirichlet}(\mathbf{c}), \\ \mathbf{z}_\ell|\boldsymbol{\theta} &\sim \text{Mult}(1, \boldsymbol{\theta}), \\ \mathbf{w}_\ell|\mathbf{z}_\ell &\sim \text{Mult}\left(1, \mathbf{d}_{\mathbf{z}_\ell}\right), \end{aligned} \qquad \text{LDA-tok model} \quad (3.1)$$

which is illustrated with the plate notation in Figure 3-1a on page 65. The Poisson assumption on the document length is discussed below. In Section 1.2.4, we rigorously showed the equivalence to the LDA-tok model to the LDA-counts model (1.17), which we refer to as the *LDA model* in this section :

$$\begin{aligned} \big(L &\sim \text{Poisson}(\lambda)\big), \\ \boldsymbol{\theta} &\sim \text{Dirichlet}(\mathbf{c}), \\ \mathbf{x}|\boldsymbol{\theta} &\sim \text{Mult}(L, \mathbf{D}\boldsymbol{\theta}), \end{aligned} \qquad \text{LDA model} \quad (3.2)$$

which is illustrated with a plate diagram in Figure 3-1b. This model is also known under the names of *multinomial PCA*, the *Dirichlet-multinomial* model, or *discrete PCA* [see also Buntine, 2002, Buntine and Jakulin, 2004, 2006] :

LDA as Discrete PCA. Principal component analysis (PCA) admits the following probabilistic interpretation [Roweis, 1998, Tipping and Bishop, 1999, see also Section 1.1.3] :

$$\begin{aligned} \boldsymbol{\alpha} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K), \\ \mathbf{x}\,|\,\boldsymbol{\alpha} &\sim \mathcal{N}(\mathbf{D}\boldsymbol{\alpha}, \sigma^2\mathbf{I}_M), \end{aligned} \qquad (3.3)$$

where $\mathbf{D} \in \mathbb{R}^{M\times k}$ is a linear transformation matrix called the factor loading matrix and $\sigma \in \mathbb{R}_{++}$ is a positive parameter. Since the maximum likelihood estimate of the matrix $\mathbf{D}$ in the model above is equivalent to the standard PCA solution, the model (3.3) is referred to as *probabilistic principal component analysis (PPCA)*.

The expectation of the observation vector in both models, (3.2) and (1.8), is equal to the linear transformation of the latent variables : $\mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})}(\mathbf{x}) = \mathbf{D}\boldsymbol{\theta}$ and $\mathbb{E}_{p(\mathbf{x}|\boldsymbol{\alpha})} = \mathbf{D}\boldsymbol{\alpha}$, respectively. Therefore, through the close connection between these two models, Buntine [2002] proposes to consider the LDA model (3.2) as a *discretization* of principal component analysis via replacing the normal likelihood in (3.3) with the multinomial one and appropriately adjusting the prior, which is usually chosen as the conjugate prior for the likelihood. This lets us see LDA as PCA for count data.

2. Recall the probability density function of the Dirichlet distribution in (A.2) and the probability mass function of the discrete distribution in (A.4).

Note that other extensions of PCA to count data were proposed in the literature. For example, the canonical PCA model [Murphy, 2012, Chapter 12] :

$$\begin{aligned} \boldsymbol{\alpha} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ \mathbf{x}|\boldsymbol{\alpha} &\sim \prod_{m=1}^{M} \text{Mult}(1, \mathcal{S}(\mathbf{D}_m \boldsymbol{\alpha} + \mathbf{w}_m)), \end{aligned}$$

where the parameters $\mathbf{D}_m \in \mathbb{R}^{M \times K}$ and $\mathbf{w}_m \in \mathbb{R}^M$, and the *softmax* function $\mathcal{S}(\mathbf{y})$ transforms a K-vector $\mathbf{y}$ of arbitrary real values to a K-dimensional vector $\mathbf{z} = \mathcal{S}(\mathbf{y})$ such that $\mathbf{z} \in \boldsymbol{\Delta}_K$: $z_j = [\mathcal{S}(\mathbf{y})]_j = e^{y_j} / \sum_k e^{y_k}$. Similar to probabilistic PCA, this model is *unidentifiable* due to the isotropic Gaussian prior for the latent variable. Moreover, the estimation and inference in this model are difficult tasks, especially, since the prior is not conjugate.

The LDA Document Length. Importantly, the LDA model does not model the document length. Indeed, although the original paper [Blei et al., 2003] proposes to model the document length as $L \sim \text{Poisson}(\lambda)$, this is never used in practice and, in particular, the parameter λ is not learned. Therefore, in the way that the LDA model is typically used, it does not provide a complete generative process of a document as there is no rule to sample L. In this section, by modeling the document length we make the link with the gamma-Poisson model and further motivate the discrete ICA model.

The Gamma-Poisson Model. The GP model[3] was introduced by Canny [2004]. It models the latent variables $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_K)$ as independent gamma variables and the counts $x_m|\boldsymbol{\alpha}$ (observations) are independent Poisson variables :

$$\begin{aligned} \alpha_k &\sim \text{Gamma}(c_k, b_k), \\ x_m|\boldsymbol{\alpha} &\sim \text{Poisson}([\mathbf{D}\boldsymbol{\alpha}]_m), \end{aligned} \qquad \text{GP model} \quad (3.4)$$

where, as before, the matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K] \in \mathbb{R}^{M \times K}$, the shape parameters $\mathbf{c} = (c_1, c_2, \ldots, c_K) \in \mathbb{R}^K_{++}$, and the rate parameters $\mathbf{b} = (b_1, b_2, \ldots, b_K) \in \mathbb{R}^K_{++}$. The GP model is illustrated with a plate diagram in Figure 3-1c. Since the expectation of the count vector $\mathbf{x} = (x_1, x_2, \ldots, x_M)$ is equal to $\mathbb{E}_{p(\mathbf{x}|\boldsymbol{\alpha})}(\mathbf{x}) = \mathbf{D}\boldsymbol{\alpha}$, the GP model is a particular case of the admixture model (1.12). In fact, the rate parameter controls the length of documents in the GP model and it is convenient to set $b_1 = b_2 = \cdots = b_K = b \in \mathbb{R}_{++}$ (see below). The gamma-Poisson model is known as a probabilistic model for non-negative matrix factorization : the original NMF updates [Lee and Seung, 1999, 2001] appear as the updates of the EM algorithm for maximum likelihood estimation in a particular gamma-Poisson model [Cemgil, 2009, Dikmen and Févotte, 2012, Paisley et al., 2014].

We will see shortly, that the LDA model (3.1) with a small extension is equivalent to

3. Recall the probability density function of a gamma random variable in (A.6) and the probability mass function of a Poisson random variable in (A.5).

the GP model (3.4) [Buntine and Jakulin, 2004, Canny, 2004]. The extension naturally arises when the document length for the LDA model is modeled as a random variable from the gamma-Poisson mixture (which is equivalent to a negative binomial random variable), i.e.,

$$L|\lambda \sim \text{Poisson}(\lambda), \qquad \lambda \sim \text{Gamma}(c_0, b), \tag{3.5}$$

where $c_0 := \sum_{k=1}^{K} c_k$.

The Connection Between the LDA and GP Models. We further show that the LDA model supplemented with the additional constraint (3.5) on the document length is equivalent to the GP model (3.4). Let $\mathbf{y} = \mathbf{D}\boldsymbol{\theta} \in \boldsymbol{\Delta}_M$. It is known [see, e.g., Ross, 2010], that if $L \sim \text{Poisson}(\lambda)$ and $\mathbf{x}|L \sim \text{Mult}(L, \mathbf{y})$ (which means that $L = \sum_m x_m$ with probability one), then $x_1, x_2, \ldots, x_M$ are mutually independent Poisson random variables with parameters $\lambda y_1, \lambda y_2, \ldots, \lambda y_M$. Hence, the LDA model with the document length assumption (3.5) is equivalent to the following model

$$\begin{aligned} \lambda &\sim \text{Gamma}(c_0, b), \\ \boldsymbol{\theta} &\sim \text{Dirichlet}(\mathbf{c}), \\ x_m \,|\, \lambda, \boldsymbol{\theta} &\sim \text{Poisson}([\mathbf{D}(\lambda\boldsymbol{\theta})]_m), \end{aligned} \tag{3.6}$$

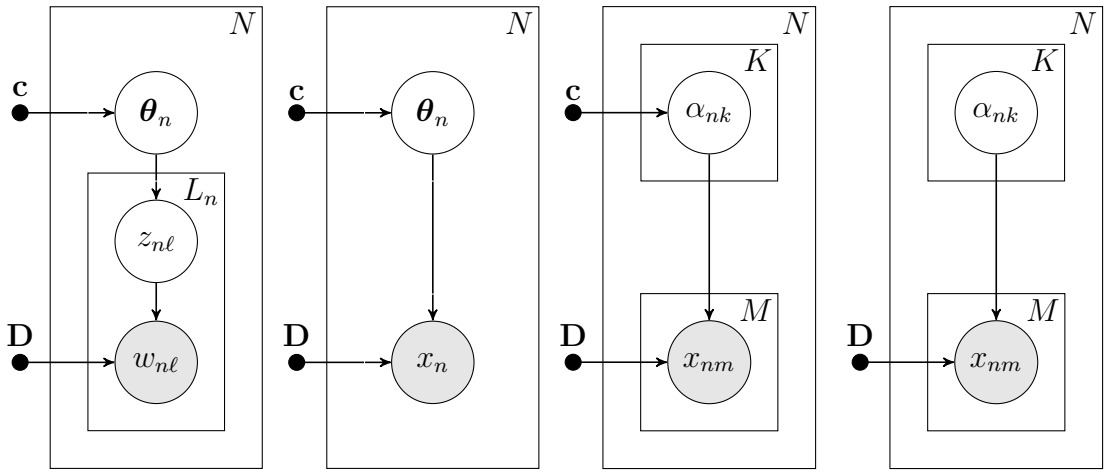
where $c_0 := \sum_k c_k$. To show the equivalence of the model (3.6) to the gamma-Poisson model (3.4), we use the fact that a Dirichlet random variable can be constructed from the normalization of independent gamma random variables [see, e.g., Frigyik et al., 2010]. More specifically, when $\alpha_1, \alpha_2, \ldots, \alpha_K$ are mutually independent gamma random variables, each $\alpha_k \sim \text{Gamma}(c_k, b)$, their sum is also a gamma random variable $\sum_k \alpha_k \sim \text{Gamma}(\sum_k c_k, b)$. The former is equivalent to λ. It is known [see, e.g., Frigyik et al., 2010], that a Dirichlet random variable can be sampled by first sampling independent gamma random variables (α_k) and then dividing each of them by their sum $(\lambda) : \theta_k = \alpha_k / \sum_{k'} \alpha_{k'}$, and, in other direction, the variables $\alpha_k = \lambda\theta_k$ are mutually independent, giving back the gamma-Poisson model (3.4).

The GP Document Length. The derivations above also imply that the document length of a document from the gamma-Poisson model (3.4) is the gamma-Poisson variable (3.5). Hence, by the law of total expectation and the law of total variance :

$$\begin{aligned} \mathbb{E}(L) &= \mathbb{E}\left[\mathbb{E}(L|\lambda)\right] = \mathbb{E}(\lambda) = c_0/b, \\ \text{var}(L) &= \text{var}\left[\mathbb{E}(L|\lambda)\right] + \mathbb{E}\left[\text{var}(L|\lambda)\right] = \text{var}(\lambda) + \mathbb{E}(\lambda) = c_0/b + c_0/b^2. \end{aligned}$$

This means that the rate parameter b can be seen as the scaling parameter for the document length, when c_0 is already prescribed : the smaller b, the larger $\mathbb{E}(L)$. On the other hand, if we allow c_0 to vary as well, only the ratio c_0/b is important for the document length. We can then interpret the role of c_0 as actually controlling the concentration of the distribution for the length L (through the variance). More specifically, we have that :

$$\frac{\text{var}(L)}{(\mathbb{E}(L))^2} = \frac{1}{\mathbb{E}(L)} + \frac{1}{c_0}. \tag{3.7}$$



(a) LDA (3.1) (b) LDA (3.2) (c) GP (3.4) (d) DICA (3.8)

FIGURE 3-1 – Plate diagrams for the models from Section 3.3.

For a fixed target document length $\mathbb{E}(L)$, we can increase the variance (and thus decrease the concentration) by using a smaller c_0.

Discrete ICA. Buntine and Jakulin [2004] propose to refer to the gamma-Poisson model (3.4) as a discrete ICA (by analogy with the discrete PCA for LDA). It is more natural, however, to name the following model

$$\begin{aligned} \alpha_1, \dots, \alpha_K &\sim \text{mutually independent}, \\ x_m|\boldsymbol{\alpha} &\sim \text{Poisson}([\mathbf{D}\boldsymbol{\alpha}]_m) \end{aligned} \qquad \text{DICA model} \quad (3.8)$$

as the *discrete*[4] *ICA (DICA)* model. The only difference between (3.8) and the standard ICA model without the additive noise (see, e.g., Section 1.1.4) is the presence of the Poisson noise which induces discrete, instead of continuous, values of x_m. Note also that (a) the discrete ICA model (3.8) is a *semiparametric* model [Bickel et al., 1998] that can adapt to any distribution on the topic intensities α_k (the distribution is unknown and we do not care to estimate this distribution) ; (b) the GP model (3.4) is a particular case of both the LDA model (3.2) and the DICA model (3.8) ; and (c) the DICA model is *identifiable* if the matrix $\mathbf{D}$ is full rank (note that the sources $\boldsymbol{\alpha}$ in the DICA model are assumed to be non-negative valued and, therefore, can not be Gaussian). The discrete ICA model is illustrated with a plate diagram in Figure 3-1d.

3.3.2 GP and Discrete ICA Cumulants

In this section, we derive and analyze the novel cumulant-based tensors of the DICA model (3.8). As the GP model (3.4) is a particular case of the DICA model, the

4. Note that the name discrete ICA was also used in the literature in the different context : for the separation of discrete sources, not the discreet observations as here [see, e.g., Senecal and Amblard, 2000, 2001].

results of this section also apply to the GP model. See Section 2.2 for the definition of cumulants and some of their properties which are necessary for the understanding of this section.

Cumulants. The first three *cumulants* of an $\mathbb{R}^M$-valued random variable $\mathbf{x}$ are defined as (see also Section 2.2) :

$$\operatorname{cum}(\mathbf{x}) := \mathbb{E}(\mathbf{x}), \tag{3.9}$$

$$\operatorname{cum}(\mathbf{x}, \mathbf{x}) := \operatorname{cov}(\mathbf{x}, \mathbf{x}) = \mathbb{E}\left[(\mathbf{x} - \mathbb{E}(\mathbf{x}))(\mathbf{x} - \mathbb{E}(\mathbf{x}))^\top\right], \tag{3.10}$$

$$\operatorname{cum}(\mathbf{x}, \mathbf{x}, \mathbf{x}) := \mathbb{E}\left[(\mathbf{x} - \mathbb{E}(\mathbf{x})) \otimes (\mathbf{x} - \mathbb{E}(\mathbf{x})) \otimes (\mathbf{x} - \mathbb{E}(\mathbf{x}))\right], \tag{3.11}$$

where we denoted $\operatorname{cum}(\mathbf{x}) = \boldsymbol{\kappa}_{\mathbf{x}}^{(1)}$, $\operatorname{cum}(\mathbf{x}, \mathbf{x}) = \boldsymbol{\kappa}_{\mathbf{x}}^{(2)}$, and $\operatorname{cum}(\mathbf{x}, \mathbf{x}, \mathbf{x}) = \boldsymbol{\kappa}_{\mathbf{x}}^{(3)}$ in the notation of Section 2.2. The essential property of cumulants, which does not hold for moments and that we use in this chapter, is that the cumulant tensor for a random vector with *independent* components is *diagonal*.

The $\mathbf{S}^{\mathbf{DICA}}$ Cumulant. Let $\mathbf{y} = \mathbf{D}\boldsymbol{\alpha}$; then for the Poisson random variable $x_m|y_m \sim \text{Poisson}(y_m)$, the expectation is $\mathbb{E}(x_m|y_m) = y_m$. Hence, by the law of total expectation and the linearity of expectation, the expectation in (3.9) has the following form

$$\mathbb{E}(\mathbf{x}) = \mathbb{E}(\mathbb{E}(\mathbf{x}|\mathbf{y})) = \mathbb{E}(\mathbf{y}) = \mathbf{D}\mathbb{E}(\boldsymbol{\alpha}). \tag{3.12}$$

Further, the variance of the Poisson random variable x_m is $\operatorname{var}(x_m|y_m) = y_m$ and, as $x_1, x_2, \ldots, x_M$ are conditionally independent given $\mathbf{y}$, then their covariance matrix is diagonal, i.e., $\operatorname{cov}(\mathbf{x}, \mathbf{x}|\mathbf{y}) = \operatorname{Diag}(\mathbf{y})$. Therefore, by the law of total covariance, the covariance in (3.10) has the form

$$\begin{aligned} \operatorname{cov}(\mathbf{x}, \mathbf{x}) &= \mathbb{E}\left[\operatorname{cov}(\mathbf{x}, \mathbf{x}|\mathbf{y})\right] + \operatorname{cov}\left[\mathbb{E}(\mathbf{x}|\mathbf{y}), \mathbb{E}(\mathbf{x}|\mathbf{y})\right] \\ &= \operatorname{Diag}\left[\mathbb{E}(\mathbf{y})\right] + \operatorname{cov}(\mathbf{y}, \mathbf{y}) \\ &= \operatorname{Diag}\left[\mathbb{E}(\mathbf{x})\right] + \mathbf{D}\operatorname{cov}(\boldsymbol{\alpha}, \boldsymbol{\alpha})\mathbf{D}^\top, \end{aligned} \tag{3.13}$$

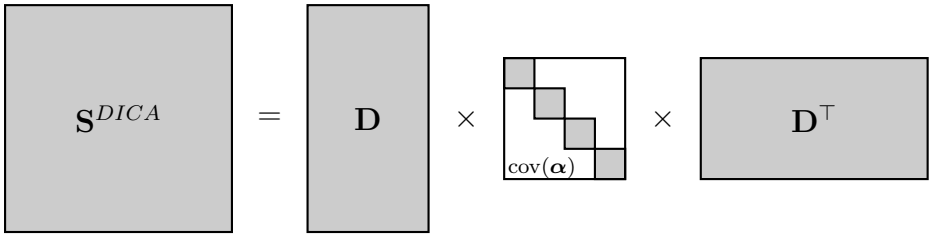
where the last equality follows by the multilinearity property of cumulants. Moving the first term from the RHS of (3.13) to the LHS, we define

$$\mathbf{S}^{DICA} := \operatorname{cov}(\mathbf{x}, \mathbf{x}) - \operatorname{Diag}\left[\mathbb{E}(\mathbf{x})\right]. \qquad \text{DICA S-cum.} \tag{3.14}$$

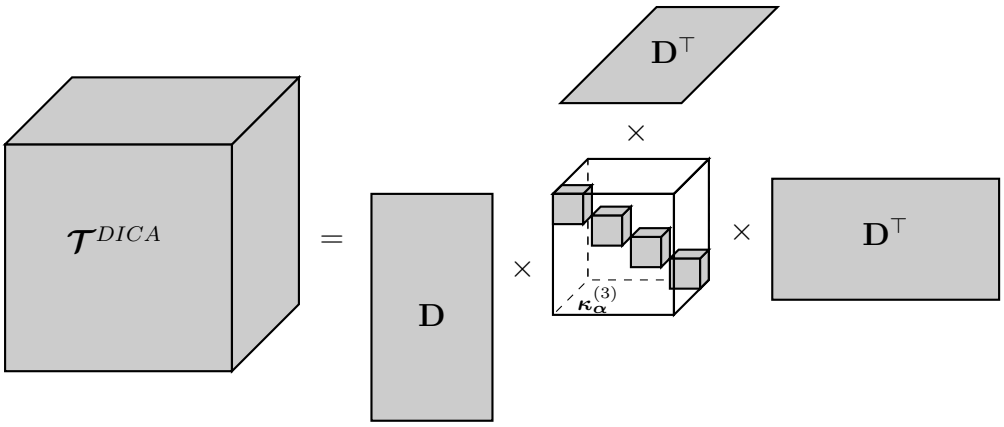
By the independence property of cumulants, the covariance of the latent variables $\boldsymbol{\alpha}$ is a diagonal matrix. Substituting this into the covariance matrix (3.13), and the covariance matrix (3.13) into the definition (3.14) of $\mathbf{S}^{DICA}$, we obtain the following diagonal structure of $\mathbf{S}^{DICA}$:

$$\mathbf{S}^{DICA} = \mathbf{D}\operatorname{cov}(\boldsymbol{\alpha}, \boldsymbol{\alpha})\mathbf{D}^\top = \sum\nolimits_k \operatorname{var}(\alpha_k)\,\mathbf{d}_k \otimes \mathbf{d}_k = \mathbf{D}\operatorname{Diag}\left[\operatorname{var}(\boldsymbol{\alpha})\right]\mathbf{D}^\top, \tag{3.15}$$

where $\mathbf{d}_k \otimes \mathbf{d}_k = \mathbf{d}_k\mathbf{d}_k^\top$ is the outer product. This diagonal structure is illustrated in Figure 3-2 and is different from the eigendecomposition : there is no orthogonality constraint on the matrix $\mathbf{D}$ in (3.15), but each column of $\mathbf{D}$ is constrained to the



(a) The DICA population $\mathbf{S}^{DICA}$ cumulant (3.15).



(b) The DICA population $\boldsymbol{\mathcal{T}}^{DICA}$ cumulant (3.18).

FIGURE 3-2 – The DICA population $\mathbf{S}^{DICA}$ and $\boldsymbol{\mathcal{T}}^{DICA}$ cumulants.

$(M-1)$-simplex.

The $\boldsymbol{\mathcal{T}}^{\mathbf{DICA}}$ Cumulant. By analogy with the second order case, using the law of total cumulance, the multilinearity property of cumulants, the independence of $\boldsymbol{\alpha}$, and the expression (3.13), we derive in Appendix B.1 the expression (B.2), similar to (3.13), for the order-3 cumulant (3.11) of the DICA model :

$$\begin{aligned}\mathrm{cum}(x_{m_1}, x_{m_2}, x_{m_3}) = &\left[\mathrm{cum}(\boldsymbol{\alpha}, \boldsymbol{\alpha}, \boldsymbol{\alpha}) \times_1 \mathbf{D}^\top \times_2 \mathbf{D}^\top \times_3 \mathbf{D}^\top)\right]_{m_1 m_2 m_3} \\ &- 2\delta(m_1, m_2, m_3)\, \mathbb{E}(x_{m_1}) + \delta(m_2, m_3)\, \mathrm{cov}(x_{m_1}, x_{m_2}) \\ &+ \delta(m_1, m_3)\, \mathrm{cov}(x_{m_1}, x_{m_2}) + \delta(m_1, m_2)\, \mathrm{cov}(x_{m_1}, x_{m_3}),\end{aligned} \tag{3.16}$$

where δ is the Kronecker delta. Moving all but the first terms in the RHS of this expression to the LHS, we define the tensor $\boldsymbol{\mathcal{T}}^{DICA}$ element-wise as follows

$$\begin{aligned}\mathcal{T}^{DICA}_{m_1 m_2 m_3} := &\ \mathrm{cum}(x_{m_1}, x_{m_2}, x_{m_3}) + 2\delta(m_1, m_2, m_3)\, \mathbb{E}(x_{m_1}) \\ &- \delta(m_2, m_3)\, \mathrm{cov}(x_{m_1}, x_{m_2}) \qquad \text{DICA T-cum.} \\ &- \delta(m_1, m_3)\, \mathrm{cov}(x_{m_1}, x_{m_2}) - \delta(m_1, m_2)\, \mathrm{cov}(x_{m_1}, x_{m_3}).\end{aligned} \tag{3.17}$$

Again, by the independence property of cumulants, the order-3 cumulant of the latent variable $\boldsymbol{\alpha}$ is a diagonal tensor. Substituting this into the order-3 cumulant (3.16) and then the order-3 cumulant in the definition (3.17) of the tensor $\boldsymbol{\mathcal{T}}^{DICA}$, we obtain the

following diagonal structure of the tensor $\boldsymbol{\mathcal{T}}^{DICA}$:

$$\boldsymbol{\mathcal{T}}^{DICA} = \sum_k \operatorname{cum}(\alpha_k, \alpha_k, \alpha_k)\, \mathbf{d}_k \otimes \mathbf{d}_k \otimes \mathbf{d}_k, \tag{3.18}$$

where $\otimes$ is the outer product. In fact, this is a non-negative symmetric CP decomposition (see Section 2.1.2) of the tensor $\boldsymbol{\mathcal{T}}^{DICA}$. See also the illustration in Figure 3-2. Sometimes, we will also refer to this form of the $\boldsymbol{\mathcal{T}}^{DICA}$ tensor and the form (3.15) of the $\mathbf{S}$ matrix as the *diagonal* form or structure. In Section 3.4, we demonstrate how to use these properties of the DICA cumulants for the estimation in the DICA model.

The LDA Moments. In Section 2.2.3, we reviewed the moment-based LDA matrix $\mathbf{S}^{LDA}$ (2.41) and tensor $\boldsymbol{\mathcal{T}}^{LDA}$ (2.42) [Anandkumar et al., 2012a, 2015a], which are analogues of the cumulant-based DICA matrix $\mathbf{S}^{DICA}$ (3.14) and tensor $\boldsymbol{\mathcal{T}}^{DICA}$ (3.17). Slightly abusing terminology, we refer to the matrix $\mathbf{S}^{LDA}$ (2.41) and the tensor $\boldsymbol{\mathcal{T}}^{LDA}$ (2.42) as the *LDA moments* and to the matrix $\mathbf{S}^{DICA}$ (3.14) and the tensor $\boldsymbol{\mathcal{T}}^{DICA}$ (3.17) as the *DICA cumulants*. The diagonal structure (2.43) and (2.44) of the LDA moments is similar to the diagonal structure (3.15) and (3.18) of the DICA cumulants, though arising through a slightly different argument. Indeed, the former is the result of properties of the Dirichlet distribution, while the latter is the result of the independence of $\boldsymbol{\alpha}$'s. However, one can think of the elements of a Dirichlet random vector as being almost independent (as, e.g., a Dirichlet random vector can be obtained from independent gamma variables through dividing each by their sum). Also, this closeness of the structures of the LDA moments and the DICA cumulants can be explained by the closeness of the respective models as discussed in Section 3.3. Importantly, due to this similarity, the algorithmic frameworks for both the DICA cumulants and the LDA moments coincide. However, LDA is parametric, while DICA is semiparametric.

The DICA cumulants have a somewhat more intuitive derivation than the LDA moments as they are expressed via the count vectors $\mathbf{x}$ (which are the sufficient statistics for the model) and not the tokens $\mathbf{w}_\ell$'s. Note also that the construction of the LDA moments depend on the unknown parameter c_0. Given that we are in an unsupervised setting and that moreover the evaluation of LDA is a difficult task [Wallach et al., 2009b], setting this parameter is non-trivial. In Section 3.5.4, we observe experimentally that the LDA moments are sensitive to the choice of c_0.

Note that another (slight) difference, which can be seen as an advantage, of the DICA cumulants from the LDA moments is that the former does not require a somewhat artificial condition of the $L \geq 3$ document length : they are well-defined for any document length.

3.3.3 Sample Complexity

Unbiased Finite Sample Estimators for the DICA Cumulants. In practice, population cumulants are never available and they have to be estimated using finite

sample estimators. Given a sample $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ of N observations or documents, we define a finite sample estimate $\widehat{\mathbf{S}}^{DICA}$ of $\mathbf{S}^{DICA}$ (3.14) and $\widehat{\boldsymbol{\mathcal{T}}}^{DICA}$ of $\boldsymbol{\mathcal{T}}^{DICA}$ (3.17) for the DICA cumulants as :

$$\widehat{\mathbf{S}}^{DICA} := \widehat{\mathrm{cov}}(\mathbf{x}, \mathbf{x}) - \mathrm{Diag}\left(\widehat{\mathbb{E}}(\mathbf{x})\right), \tag{3.19}$$

$$\begin{aligned}\widehat{\mathcal{T}}^{DICA}_{m_1 m_2 m_3} :=& \widehat{\mathrm{cum}}(x_{m_1}, x_{m_2}, x_{m_3}) + 2\delta(m_1, m_2, m_3)\widehat{\mathbb{E}}(x_{m_1}) \\ &- \delta(m_2, m_3)\widehat{\mathrm{cov}}(x_{m_1}, x_{m_2}) \\ &- \delta(m_1, m_3)\widehat{\mathrm{cov}}(x_{m_1}, x_{m_2}) \\ &- \delta(m_1, m_2)\widehat{\mathrm{cov}}(x_{m_1}, x_{m_3}),\end{aligned} \tag{3.20}$$

where unbiased estimators of the first three cumulants are

$$\begin{aligned}\widehat{\mathbb{E}}(x_{m_1}) =& \frac{1}{N}\sum_{n=1}^{N} x_{nm_1}, \\ \widehat{\mathrm{cov}}(x_{m_1}, x_{m_2}) =& \frac{1}{N-1}\sum_{n=1}^{N} z_{nm_1} z_{nm_2}, \\ \widehat{\mathrm{cum}}(x_{m_1}, x_{m_2}, x_{m_3}) =& \frac{N}{(N-1)(N-2)}\sum_{n=1}^{N} z_{nm_1} z_{nm_2} z_{nm_3},\end{aligned} \tag{3.21}$$

where the word vocabulary indices are $m_1, m_2, m_3 = 1, 2, \ldots, M$ and the centered documents $z_{nm} := x_{nm} - \widehat{\mathbb{E}}(x_m)$. (The latter is introduced only for compact representation of (3.21) and is different from the latent variable $\mathbf{z}$ in the LDA as well as other models.)

Expressions for fast implementation of these finite sample estimators are derived in Appendix C.1.2. These expressions are used for fast implementation of algorithms in a software package, which is a part of this thesis work (see Appendix C).

Similar expressions for finite sample estimators of the LDA moment-based tensors are presented in Section 2.2.3 and expressions for their fast implementation are derived in Appendix C.1.1. These expressions are also used for fast implementation of the respective algorithms in the mentioned software package.

Sample Complexity. The following sample complexity results apply to the sample estimates of the GP cumulants : [5]

Proposition 3.3.1. *Under the GP model, the expected error for the sample estimator*

5. Note that the expected squared error for the DICA cumulants is similar, but the expressions are less compact and, in general, depend on the prior on α_k.

$\widehat{S}$ (3.19) *for the GP cumulant* S (3.14) *is :*

$$\begin{aligned}\mathbb{E}\left[\|\widehat{\mathbf{S}}^{GP} - \mathbf{S}^{GP}\|_F\right] &\leq \sqrt{\mathbb{E}\left[\|\widehat{\mathbf{S}}^{GP} - \mathbf{S}^{GP}\|_F^2\right]} \\ &\leq O\left(\frac{1}{\sqrt{N}} \max\left[\Delta \bar{L}^2, \bar{c}_0 \bar{L}\right]\right),\end{aligned} \tag{3.22}$$

where $\Delta := \max_k \|\mathbf{d}_k\|_2^2$, $\bar{c}_0 := \min(1, c_0)$ *and* $\bar{L} := \mathbb{E}(L)$.

A high probability bound could be derived using concentration inequalities for Poisson random variables [Boucheron et al., 2013] ; but the expectation already gives the right order of magnitude for the error (for example via Markov's inequality). A sketch of a proof for Proposition 3.3.1 can be found in Appendix B.2. See also the discussion of this sample complexity results in the end of Section 3.4.1.

We do not present the exact expression for the expected squared error for the estimator of $\boldsymbol{\mathcal{T}}^{GP}$, but due to a similar structure in the derivation, we expect the analogous bound of $\mathbb{E}\big[\|\widehat{\boldsymbol{\mathcal{T}}}^{GP} - \boldsymbol{\mathcal{T}}^{GP}\|_F\big] \leq 1/\sqrt{N} \max\{\Delta^{3/2}\bar{L}^3, \bar{c}_0^{3/2}\bar{L}^{3/2}\}$.

Current sample complexity results of the LDA moments [Anandkumar et al., 2012a] can be summarized as $O(1/\sqrt{N})$. However, the proof (which can be found in the supplementary material [Anandkumar et al., 2013a]) analyzes only the case when finite sample estimates of the LDA moments are constructed from *one* triple per document, i.e., $\mathbf{w}_1 \otimes \mathbf{w}_2 \otimes \mathbf{w}_3$ only, and not from the U-statistics that average multiple (dependent) triples per document as in the practical expressions (2.45) and (2.46) (Section 2.2.3). Moreover, one has to be careful when comparing upper bounds. Nevertheless, comparing the bound (3.22) with the current theoretical results for the LDA moments, we see that the GP/DICA cumulants sample complexity contains the ℓ_2-norm of the columns of the topic matrix $\mathbf{D}$ in the numerator, as opposed to the $O(1)$ coefficient for the LDA moments. This norm can be significantly smaller than 1 for vectors in the simplex (e.g., $\Delta = O(1/\|\mathbf{d}_k\|_0)$ for sparse topics). This suggests that the GP/DICA cumulants may have better finite sample convergence properties than the LDA moments and our experimental results in Section 3.5 are indeed consistent with this statement. This difference, however, should decrease with the growth of the average document length in a corpus.

3.4 Estimation in the GP and DICA Models

In this section, we propose several algorithms for the estimation in the GP and DICA models. These algorithms are based on the symmetric CP structure of the DICA cumulant-based matrix $\mathbf{S}^{DICA}$ and third-order tensor $\boldsymbol{\mathcal{T}}^{DICA}$ and consist of a two step procedure similar to the one used for the estimation in the LDA (ICA) models as described in Section 2.4.2) : first, (a) the *prewhitening* of the matrix $\mathbf{S}^{DICA}$ and, second, (b) finding the "right" orthogonal transformation using one of the algorithms for the orthogonal symmetric CPD of the prewhitened version of the tensor $\boldsymbol{\mathcal{T}}^{DICA}$.

The second step can be performed in several different ways : through the eigendecomposition of a contraction of the prewhitened tensor $\boldsymbol{\mathcal{T}}^{DICA}$, through the tensor power method of the prewhitened tensor $\boldsymbol{\mathcal{T}}^{DICA}$, or through orthogonal joint diagonalization of several contractions of the prewhitened tensor $\boldsymbol{\mathcal{T}}^{DICA}$.

Prewhitening. By analogy with Section 2.4.2, we perform the *prewhitening* of the matrix $\mathbf{S}^{DICA}$, that is we find a matrix $\mathbf{W} \in \mathbb{R}^{K \times M}$ such that $\mathbf{W}\mathbf{S}^{DICA}\mathbf{W}^\top = \mathbf{I}_K$. Such matrix $\mathbf{W}$ is not uniquely defined, but can be easily found through the SVD of $\mathbf{S}^{DICA}$ and truncation of the $M - K$ smallest eigenpairs.

The diagonal form (3.15) of the matrix $\mathbf{S}^{DICA}$ can be rewritten, without loss of generality, in the form $\mathbf{S}^{DICA} = \widetilde{\mathbf{D}}\widetilde{\mathbf{D}}^\top$, where $\widetilde{\mathbf{D}} := \mathbf{D}\mathrm{cov}(\boldsymbol{\alpha})^{1/2}$, since $\mathrm{cov}(\boldsymbol{\alpha})$ is diagonal and due to the scaling unidentifiability. Since an orthogonally transformed whitening matrix is still a whitening matrix, there exists an orthogonal matrix $\mathbf{Q}$ such that $\mathbf{Q}\mathbf{W}\widetilde{\mathbf{D}} = \mathbf{I}$ (up to unidentifiable permutation and scaling which we ignore in this section). This means, finding the "right" orthogonal matrix $\mathbf{Q}$ would allow us to recover the matrix $\widetilde{\mathbf{D}}$ (up to permutation and scaling). To fix this degree of freedom, we use the third-order tensor $\boldsymbol{\mathcal{T}}^{DICA}$.

Let us transform the tensor $\boldsymbol{\mathcal{T}}^{DICA}$ with our whitening matrix $\mathbf{W}$ along all modes :

$$\overline{\boldsymbol{\mathcal{T}}}^{DICA} := \boldsymbol{\mathcal{T}}^{DICA} \times_1 \mathbf{W}^\top \times_2 \mathbf{W}^\top \times_3 \mathbf{W}^\top. \tag{3.23}$$

When the matrix $\mathbf{Q}$ from above is known, transforming this tensor along all modes with $\mathbf{Q}$ diagonalizes the tensor, i.e. $\overline{\boldsymbol{\mathcal{T}}}^{DICA} \times_1 \mathbf{Q}^\top \times_2 \mathbf{Q}^\top \times_3 \mathbf{Q}^\top$ is *diagonal*. This suggests a two-step routine for the estimation of $\mathbf{D}$, which consists of computing a whitening matrix $\mathbf{W}$ and an orthogonal matrix $\mathbf{Q}$. Below we briefly outline three algorithms for the approximation of such matrix $\mathbf{Q}$, which were described in Sections 2.1.2 and 2.4.2. These algorithms can be applied to any pair of $\mathbf{S}$ and $\boldsymbol{\mathcal{T}}$ in the diagonal (symmetric CP) form (4.21), including the ICA cumulants (see Section 2.2.2), the LDA moments (see Section 2.2.3), the DICA cumulants (see Section 3.3.2), and higher-order statistics of many other models.

The Eigendecomposition Based Algorithm. The easiest approach to the approximation of the matrix $\mathbf{Q}$ is through the eigendecomposition (ED) of a contracted with some vector $\mathbf{u} \in \mathbb{R}^K$ tensor $\overline{\boldsymbol{\mathcal{T}}}^{DICA}$ (see Section 2.4.2 for the definition of the contraction, a.k.a. projection, of a tensor with a vector), which is a matrix $\overline{\boldsymbol{\mathcal{T}}}^{DICA}(\mathbf{u})$. In the ideal case of the population cumulants, this method finds the global solution. However, in practice, when one deals with finite sample estimates of $\mathbf{S}^{DICA}$ and $\boldsymbol{\mathcal{T}}^{DICA}$, the diagonal form in (4.21) is only approximate. In this case, contracting a tensor with only one vector leads to an important loss of information (reducing K^3 to K^2 elements), which leads to poor approximations in practice, which we also observe in experiments (see Section 3.5).

The Tensor Power Method. Another approach for the approximation of $\mathbf{Q}$ is the tensor power method [see Anandkumar et al., 2014, and references therein], which is a direct extension of the matrix power method to tensors. The main idea of the

Algorithm 4 The OJD algorithm for GP/DICA cumulants

1: *Input :* $\mathbf{X} \in \mathbb{R}^{M \times N}$, K, $1 \leq P \leq K$ (number of random contractions)
2: Compute sample estimate $\widehat{\mathbf{S}}^{DICA} \in \mathbb{R}^{M \times M}$
3: Compute a whitening matrix $\widehat{\mathbf{W}} \in \mathbb{R}^{K \times M}$ of $\widehat{\mathbf{S}}$
4: Construct vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_P\}$ for random contractions as $\mathbf{v}_p = \widehat{\mathbf{W}}^\top \mathbf{u}_p$ and :
option (a) : Choose P vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_P\} \subseteq \mathbb{R}^K$ uniformly at random from the unit ℓ_2-sphere ($P = 1$ yields the ED-based algorithm)
option (b) : Set $P = K$ and choose vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_P\} \subseteq \mathbb{R}^K$ as the canonical basis $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_K$ of $\mathbb{R}^K$
5: Construct target matrices $\mathbf{B}_p = \widehat{\mathbf{W}} \widehat{\boldsymbol{\mathcal{T}}}^{DICA}(\widehat{\mathbf{W}}^\top \mathbf{u}_p)\widehat{\mathbf{W}}^\top \in \mathbb{R}^{K \times K}$ for each $p \in [P]$
6: Find an orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{K \times K}$ that jointly diagonalizes the matrices

$$\widehat{\mathbf{W}} \widehat{\mathbf{S}}^{DICA} \widehat{\mathbf{W}}^\top, \qquad \mathbf{B}_p, \quad p \in [P]$$

7: *Output :* the matrix $\mathbf{Q}\widehat{\mathbf{W}}$ and (nearly diagonal) matrices $\mathbf{Q}\mathbf{B}_p\mathbf{Q}^\top$

algorithm is to estimate the columns of $\mathbf{Q}$ one-by-one using the *deflation* approach. At each deflation step, a vector $\mathbf{q}_k$ is approximated through iterative tensor power updates as we outlined in Section 2.1.2. In the ideal case of the population cumulants, the algorithm finds the global solution at the quadratic convergence rate. In the practical case with the finite sample and possibly misspecification errors, tensor power method only finds an approximation of the matrix $\mathbf{Q}$. Given the additive noise to the orthogonal symmetric CPD structure of the sample estimate of the tensor $\overline{\boldsymbol{\mathcal{T}}}^{DICA}$ is not large, a TPM approximation of a vector $\mathbf{q}_k$ is close to this vector $\mathbf{q}_k$ in terms of the ℓ_2-error in accordance with the perturbation analysis of the tensor power method [Anandkumar et al., 2014].

Orthogonal Joint Matrix Diagonalization. Another algorithm for the approximation of the matrix $\mathbf{Q}$ can be seen as a stabilization of the eigendecomposition-based algorithm. The key idea is to take several contractions of the prewhitened tensor $\widehat{\boldsymbol{\mathcal{T}}}^{DICA} \times_1 \widehat{\mathbf{W}}^\top \times_2 \widehat{\mathbf{W}}^\top \times_3 \widehat{\mathbf{W}}^\top$ with different vectors $\mathbf{u}_p$, for $p \in [P]$, and *jointly* diagonalize the obtained matrices $\widehat{\mathbf{W}} \widehat{\boldsymbol{\mathcal{T}}}^{DICA}(\widehat{\mathbf{W}}^\top \mathbf{u}_p)\widehat{\mathbf{W}}^\top$. This is also known as *joint (symmetric) eigendecomposition* of several matrices and we described a Jacobi-like algorithm for this problem in Section 2.3.1. This method applied to the estimation problem in the DICA model is outlined in Algorithm 4.

The choice of the number P of the contraction (projection) vectors $\mathbf{v}_1$, $\mathbf{v}_2$, $\ldots$, $\mathbf{v}_P$ is not straightforward. The case $P = 1$ corresponds to the ED-based algorithm described above and leads to tremendous loss of information contained in the original tensor. The choice of the canonical basis in $\mathbb{R}^K$ and vectors $\mathbf{v}_p := \widehat{\mathbf{W}}^\top \mathbf{e}_p$ for $p \in [K]$ preserves all information contained in the original tensor, but can be computationally expensive if K is too large. Choosing P between 1 and K which leads to a good trade-off between the quality of the approximation and the computational complexity is thus of interest. In practice, choosing even small values of P, e.g, $P = 2$ or 3, lead to a significant

improvement over the ED-based algorithm. Ideally, it would be of interest to show rigorously that setting P to a value close to $\log(K)$ is sufficient for an accurate approximation (which is observed in practice).

Estimation of the Topic Matrix. An eventual goal of the procedure is the estimation of the topic matrix $\mathbf{D}$ given the outputs of the algorithms above : $\widehat{\mathbf{W}}$ and $\mathbf{Q}$. Since (up to permutation and scaling) $\mathbf{Q}\widehat{\mathbf{W}}\mathbf{D} = \mathbf{I}$, one can estimate $\widehat{\mathbf{D}}$ in a multistep procedure : (a) *compute* a matrix $\widehat{\mathbf{W}}^{\dagger}\mathbf{Q}^{\top}$, (b) for each column of this matrix, *set the sign* such that the vector (column) has less negative elements than positive, which is measured by the sum of squares of the elements of each sign, (c) *truncate* all negative values, and (d) *normalize* the columns of the resulting matrix to sum to one. The step in (b) is necessary due to the scaling unidentifiability which includes the scaling by -1. This heuristic procedure is necessary to preserve the non-negativity of the elements of the topic matrix. More justified procedure would be of interest, however, it is not readily available. One could consider using the non-negative CP decomposition instead of the two-step procedure outlined in this section, but this is not straightforward because of (a) the symmetric case and (b) the high dimension M of the original space before prewhitening. We tried to integrate a non-negativity enforcing regularization into the optimization problem of the OJD algorithm. However, the search space, which is the Stiefel manifold in this case, is too restrictive and the resulting algorithm does not have any significant increase of the approximation quality in practice.

3.4.1 Analysis of the Whitening and Recovery Error

In this section, we extend the topic recovery guarantees of Anandkumar et al. [2013a] to the ED-based algorithm for the GP and DICA models. Let $\mathbf{S} := \mathbf{S}^{DICA}$ and $\boldsymbol{\mathcal{T}} := \boldsymbol{\mathcal{T}}^{DICA}$.

We can follow a similar analysis as in Appendix C of Anandkumar et al. [2013a] to derive the topic recovery error given the sample estimate error. In particular, if we define the following sampling errors $E_{\mathbf{S}}$ and $E_{\boldsymbol{\mathcal{T}}}$:

$$\begin{aligned}\|\widehat{\mathbf{S}} - \mathbf{S}\| &\leq E_{\mathbf{S}},\\ \|\widehat{\boldsymbol{\mathcal{T}}}(\mathbf{u}) - \boldsymbol{\mathcal{T}}(\mathbf{u})\| &\leq \|\mathbf{u}\|_2 E_{\boldsymbol{\mathcal{T}}},\end{aligned}$$

then the following form of their Lemma C.2 holds for the DICA cumulants :

$$\|\widehat{\mathbf{W}}\widehat{\boldsymbol{\mathcal{T}}}(\widehat{\mathbf{W}}^{\top}\mathbf{u})\widehat{\mathbf{W}}^{\top} - \mathbf{W}\boldsymbol{\mathcal{T}}(\mathbf{W}^{\top}\mathbf{u})\mathbf{W}^{\top}\| \leq \nu\left[\frac{(\max_k \gamma_k)E_{\mathbf{S}}}{\sigma_K(\widetilde{\mathbf{D}})^2} + \frac{E_{\boldsymbol{\mathcal{T}}}}{\sigma_K(\widetilde{\mathbf{D}})^3}\right], \tag{3.24}$$

where $\sigma_k(\cdot)$ denotes the k-th singular value of a matrix, ν is a universal constant, and $\widetilde{\mathbf{D}}$ is such that $\mathbf{S} = \widetilde{\mathbf{D}}\widetilde{\mathbf{D}}^{\top}$. The values of $\gamma_k := \mathrm{cum}(\alpha_k)[\Delta(\alpha_k)]^{-1.5}$ for the DICA cumulants. Note that similar expression holds for the LDA moments with $\gamma_k = 2\sqrt{c_0(c_0+1))\left[c_k(c_0+2)^2\right]^{-1}}$. Moreover, the GP cumulants are a special case

of the DICA cumulants and substituting the gamma prior on $\boldsymbol{\alpha}$ one readily obtains $\gamma_k := 2c_k^{-0.5}$.

We note that the scaling for $\mathbf{S}$ is $O(L^2)$ for the GP/DICA cumulants, in contrast to $O(1)$ for the LDA moments. Thus, to compare the upper bound (3.24) for the two types of moments, we need to put it in quantities which are common. In the first section of the Appendix C of Anandkumar et al. [2013a], it was mentioned that $\sigma_K(\widetilde{\mathbf{D}}) \geq \sqrt{c_{\min}\left[c_0(c_0+1)\right]^{-1}}\sigma_K(\mathbf{D})$ for the LDA moments, where $c_{\min} := \min_k c_k$. We further switch to the GP model, where the dependence on L is transparent. However, similar results are readily available for the DICA model. So, in contrast, for the GP cumulants, we can show that $\sigma_K(\widetilde{\mathbf{D}}) \geq \bar{L}\sqrt{c_{\min}}c_0^{-1}\sigma_K(\mathbf{D})$, where $\bar{L} := c_0/b$ (i.e. $\bar{L} := \mathbb{E}(L)$) is the expected length of a document in the GP model. Using this lower bound for the singular vector, we thus get the following bound for the GP cumulant :

$$\begin{aligned}\|\widehat{\mathbf{W}}\widehat{\boldsymbol{\mathcal{T}}}^{GP}(\widehat{\mathbf{W}}^\top\mathbf{u})\widehat{\mathbf{W}}^\top - \mathbf{W}\boldsymbol{\mathcal{T}}^{GP}(\mathbf{W}^\top\mathbf{u})\mathbf{W}^\top\| \\ \leq \frac{\nu}{c_{\min}^{3/2}}\left[\frac{E_{\mathbf{S}}}{\bar{L}^2}\frac{2c_0^2}{\left[\sigma_K(\mathbf{D})\right]^2} + \frac{E_{\boldsymbol{\mathcal{T}}}}{\bar{L}^3}\frac{c_0^3}{\left[\sigma_K(\mathbf{D})\right]^3}\right].\end{aligned} \tag{3.25}$$

The $c_{\min}^{3/2}$ factor is common for both the LDA moment and GP cumulant, however, the sample error $E_{\mathbf{S}}$ term gets divided by $\bar{L}^2$ for the GP cumulant, as expected. Indeed, the prewhitening transformation for the GP $\mathbf{S}^{GP}$ matrix redivides the error $E_{\mathbf{S}}$ on $\mathbf{S}^{GP}$ (3.22) by $\bar{L}^2$, which is the scale of $\mathbf{S}^{GP}$. This means that the contribution from $\widehat{\mathbf{S}}^{GP}$ to the recovery error will scale as $O(1/\sqrt{N}\max\{\Delta, \bar{c}_0/\bar{L}\})$, where both Δ and $\bar{c}_0/\bar{L}$ are smaller than 1 and can be very small. This argument indicates that the (bound on the) sample complexity is better for GP cumulants vs. LDA moments.

The recovery error bound by Anandkumar et al. [2013a] is based on the bound (3.25), and thus by showing that the error $E_{\mathbf{S}}/\bar{L}^2$ for the GP cumulant is lower than the $E_{\mathbf{S}}$ term for the LDA moment, we expect to also gain a similar gain for the recovery error, as the rest of the argument is the same for both types of moments (see Appendix C.2, C.3 and C.4 by Anandkumar et al. [2013a]).

3.5 Experiments

In this section, (a) we compare experimentally the GP/DICA cumulants with the LDA moments and (b) the spectral algorithm [Anandkumar et al., 2012a], the tensor power method [Anandkumar et al., 2014] (TPM), the joint diagonalization (JD) algorithm from Algorithm 4, and variational inference for LDA [Blei et al., 2003].

3.5.1 Datasets

Real Data. Real data includes the associated press (AP) dataset, from D. Blei's web page,[6] with $N = 2,243$ documents and $M = 10,473$ vocabulary words and the average document length $\widehat{L} = 194$; the NIPS papers dataset[7] [Globerson et al., 2007] of $2,483$ NIPS papers and $14,036$ words, and $\widehat{L} = 1,321$; the KOS dataset,[8] from the UCI Repository, with $3,430$ documents and $6,906$ words, and $\widehat{L} = 136$.

Semi-Synthetic Data. Semi-synthetic data are constructed by analogy with Arora et al. [2013] : (1) the LDA parameters $\mathbf{D}$ and $\mathbf{c}$ are learned from the real datasets with variational inference and (2) synthetic data are sampled from a model of interest with the given parameters $\mathbf{D}$ and $\mathbf{c}$. This provides the ground truth parameters $\mathbf{D}$ and $\mathbf{c}$. For each setting, data are sampled 5 times and the results are averaged. We plot error bars that are the minimum and maximum values. For the AP data, $K \in \{10, 50\}$ topics are learned and, for the NIPS data, $K \in \{10, 90\}$ topics are learned. For larger K, the obtained topic matrix is ill-conditioned, which violates the identifiability condition for topic recovery using moment matching techniques [Anandkumar et al., 2012a]. All the documents with less than 3 tokens are resampled.

Sampling Techniques. All the sampling models have the parameter $\mathbf{c}$ which is set to $\mathbf{c} = c_0 \bar{\mathbf{c}} / \|\bar{\mathbf{c}}\|_1$, where $\bar{\mathbf{c}}$ is the learned $\mathbf{c}$ from the real dataset with variational LDA, and c_0 is a parameter that we can vary. The *GP* data are sampled from the gamma-Poisson model (3.4) with $b = c_0/\widehat{L}$ so that the expected document length is $\widehat{L}$. The *LDA-fix(L)* data are sampled from the LDA model (3.2) with the document length being fixed to a given L. The *LDA-fix2(γ,L_1,L_2)* data are sampled as follows : $(1-\gamma)$-portion of the documents are sampled from the *LDA-fix(L_1)* model with a given document length L_1 and γ-portion of the documents are sampled from the *LDA-fix(L_2)* model with a given document length L_2.

3.5.2 Code and Complexity

Our (mostly Matlab) implementations of the diagonalization algorithms (JD, Spec, and TPM) for both the GP/DICA cumulants and LDA moments are available online.[9] Moreover, all datasets and the code for reproducing our experiments are available.[10] Each experiment was run in a single thread.

The bottleneck for the spectral (i.e. ED-based), JD, and TPM algorithms is the computation of the cumulants/moments. However, the expressions (C.5) and (C.4) provide efficient formulas for fast computation of the GP/DICA cumulants and LDA moments ($O(RNK + NK^2)$, where R is the largest number of non-zeros in the count vector x over all documents, see Appendix C.1), which makes even the Matlab implementation fast for large datasets (see, e.g., Table 3.1). Since all diagonalization

6. http://www.cs.columbia.edu/~blei/lda-c
7. http://ai.stanford.edu/~gal/data
8. https://archive.ics.uci.edu/ml/datasets/Bag+of+Words
9. https://github.com/anastasia-podosinnikova/dica-light
10. https://github.com/anastasia-podosinnikova/dica

algorithms (spectral, JD, TPM) perform the whitening step once, it is sufficient to compare their complexities by the number of times the cumulants/moments are computed.

Spectral. The spectral algorithm estimates the cumulants/moments only once leading to $O(NK(R+K))$ complexity and, therefore, is the fastest.

OJD. For JD, rather than estimating P cumulants/moments separately, one can jointly estimate these values by precomputing and reusing some terms (e.g., WX). However, the complexity is still $O(PNK(R+K))$, although in practice it is sufficient to have $P = K$ or even smaller.

TPM. For TPM some parts of the cumulants/moments can also be precomputed, but as TPM normally does many more iterations than P, it can be significantly slower. In general, the complexity of TPM can be significantly influenced by the initialization of the parameters of the algorithm. There are two main parameters : L_{TPM} is the number of random restarts within one deflation step and N_{TPM} is the maximum number of iterations for each of L_{TPM} random restarts (note that these are different from the number of documents N and document L). Some restarts converge very fast (in much less than N_{TPM} iterations), while others are slow. Moreover, as follows from theoretical results [Anandkumar et al., 2014] and, as we observed in practice, the restarts which converge to a good solution converge fast, while slow restarts, normally, converge to a worse solution. Nevertheless, in the worst case, the complexity is $O(N_{TPM}L_{TPM}NK(R+K))$.

Note that for the experiment in Figure 3-3, $L_{TPM} = 10$ and $N_{TPM} = 100$ and the run with the best objective is chosen. We believe that these values are reasonable in a sense that they provide a good accuracy solution ($\varepsilon = 10^{-5}$ for the norm of the difference of the vectors from the previous and the current iteration) in a little number of iterations, however, they may not be the best ones.

JD Implementation. For the orthogonal joint diagonalization algorithm, we implemented a faster C++ version of the previous Matlab implementation [11] by J.-F. Cardoso. Moreover, the orthogonal joint diagonalization routine can be initialized in different ways : (a) with the $K \times K$ identity matrix or (b) with a random orthogonal $K \times K$ matrix. We tried different options and in nearly all cases the algorithm converged to the same solution, implying that initialization with the identity matrix is sufficient.

Whitening Matrix. For the large vocabulary size M, computation of a whitening matrix can be expensive (in terms of both memory and time). One possible solution would be to reduce the vocabulary size with, e.g., by selecting according to the tf-idf score, which is a standard practice in the topic modeling context. Another option is using a stochastic eigendecomposition [see, e.g., Halko et al., 2011] to approximate the whitening matrix.

Variational Inference. For variational inference, we used the code of D. Blei and

11. http://perso.telecom-paristech.fr/~cardoso/Algo/Joint_Diag/joint_diag_r.m

modified it for the estimation of a non-symmetric Dirichlet prior $\mathbf{c}$, which is known to be important [Wallach et al., 2009a]. The default values of the tolerance/maximum number of iterations parameters are used for variational inference. The computational complexity of one iteration for one document of the variational inference algorithm is $O(RK)$, where R is the number of non-zeros in the count vector for this document, which is then performed a significant number of times for each document.

Evaluation. The evaluation of topic recovery for semi-synthetic data is performed with the ℓ_1-error between the recovered $\widehat{\mathbf{D}}$ and true $\mathbf{D}$ topic matrices with the best permutation of columns : $\mathrm{err}_{\ell_1}(\widehat{\mathbf{D}}, \mathbf{D}) := \min_{\boldsymbol{\pi} \in \mathrm{PERM}} \frac{1}{2K} \sum_k \|\widehat{\mathbf{d}}_{\pi_k} - \mathbf{d}_k\|_1 \quad \in [0, 1]$. The minimization is over the possible permutations $\boldsymbol{\pi} \in \mathrm{PERM}$ of the columns of $\widehat{\mathbf{D}}$ and can be efficiently obtained with the Hungarian algorithm for bipartite matching [Kuhn, 1955].

Evaluation of the Real Data Experiments. For the evaluation of topic recovery in the real data case, we use an approximation of the log-likelihood for held out documents as the metric. The approximation is computed using a Chib-style method as described by Wallach et al. [2009b] using the implementation by the authors. [12] Importantly, this evaluation method is applicable for both the LDA model as well as the GP model. Indeed, as it follows from Section 3.3, the GP model is equivalent to the LDA model when conditioning on the length of a document L (with the same c_k hyper parameters), while the LDA model does not make any assumption on the document length. For the test log-likelihood comparison, we thus treat the GP model as a LDA model (we do not include the likelihood of the document length).

We use our Matlab implementation of the GP/DICA cumulants, the LDA moments, and the diagonalization algorithms. The datasets and the code for reproducing our experiments are available online. [13]

Initialization of the Parameter c_0 for the LDA Moments. The construction of the LDA moments requires the parameter c_0, which is not trivial to set in the unsupervised setting of topic modeling, especially taking into account the complexity of the evaluation for topic models [Wallach et al., 2009b]. For the semi-synthetic experiments, the true value of c_0 is provided to the algorithms. It means that the LDA moments, in this case, have access to some oracle information, which in practice is never available. For real data experiments, c_0 is set to the value obtained with variational inference. Experiments show that this choice was somewhat important (see, e.g., Figure 3-5). However, this requires more thorough investigation.

3.5.3 Comparison of the Diagonalization Algorithms

In Figure 3-3, we compare the diagonalization algorithms on the semi-synthetic AP dataset for $K = 50$ using the GP sampling. We compare the tensor power method [TPM ; Anandkumar et al., 2014], the spectral algorithm (Spec), the orthogonal joint

12. http://homepages.inf.ed.ac.uk/imurray2/pub/09etm
13. https://github.com/anastasia-podosinnikova/dica

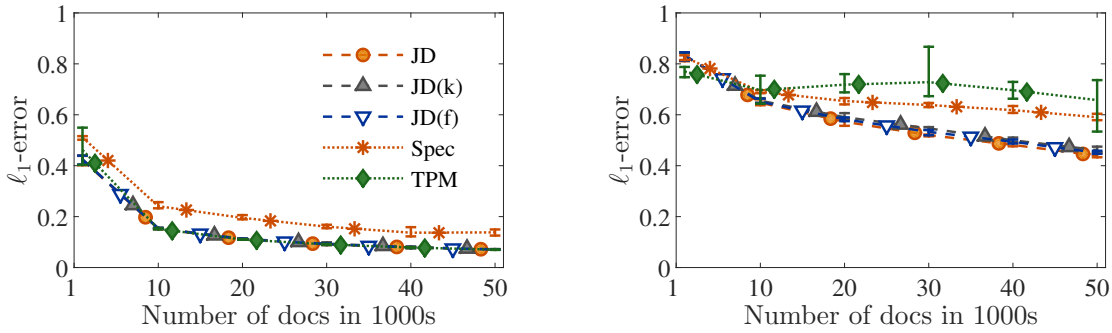


FIGURE 3-3 – Comparison of the diagonalization algorithms. The topic matrix $\mathbf{D}$ and Dirichlet parameter $\mathbf{c}$ are learned for $K = 50$ from AP ; $\mathbf{c}$ is scaled to sum up to 0.5 and b is set to fit the expected document length $\widehat{L} = 200$. The semi-synthetic dataset is sampled from *GP* ; number of documents N varies from $1,000$ to $50,000$. *Left :* GP/DICA moments. *Right :* LDA moments. *Note* : a smaller value of the ℓ_1-error is better.

diagonalization algorithm (JD) described in Algorithm 4 with different options to choose the random projections : JD(k) takes $P = K$ vectors $\mathbf{u}_p$ sampled uniformly from the unit ℓ_2-sphere in $\mathbb{R}^K$ and selects $\mathbf{v}_p = \mathbf{W}^\top \mathbf{u}_p$ (option (a) in Algorithm 4) ; JD selects the full basis $\mathbf{e}_1, \ldots, \mathbf{e}_K$ in $\mathbb{R}^K$ and sets $\mathbf{v}_p = \mathbf{W}^\top \mathbf{e}_p$ (as JADE [Cardoso and Souloumiac, 1993]) (option (b) in Algorithm 4) ; $JD(f)$ chooses the full canonical basis of $\mathbb{R}^M$ as the projection vectors (computationally expensive).

Both the GP/DICA cumulants and LDA moments are well-specified in this setup. However, the LDA moments have a slower finite sample convergence and, hence, a larger estimation error for the same value N. As expected, the spectral algorithm is always slightly inferior to the joint diagonalization algorithms. With the GP/DICA cumulants, where the estimation error is low, all algorithms demonstrate good performance, which also fulfills our expectations. However, although TPM shows almost perfect performance in the case of the GP/DICA cumulants (left), it significantly deteriorates for the LDA moments (right), which can be explained by the larger estimation error of the LDA moments and lack of robustness of TPM. Overall, the orthogonal joint diagonalization algorithm with initialization of random projections as $\mathbf{W}^\top$ multiplied with the canonical basis in $\mathbb{R}^K$ (JD) is both computationally efficient and fast.

Runtimes of the Diagonalization Algorithms

In Table 3.1, we present the running times of the algorithms from Section 3.5.3. JD and JD(k) are significantly faster than JD(f) as expected, although the performance in terms of the ℓ_1-error is nearly the same for all of them. This indicates that preference should be given to the JD or JD(k) algorithms.

The running time of all LDA-algorithms is higher than the one of the GP/DICA-algorithms. This indicates that the computational complexity of the LDA-moments

	min	mean	max
JD-GP	148	192	247
JD-LDA	252	284	366
JD(k)-GP	157	190	247
JD(k)-LDA	264	290	318
JD(f)-GP	1628	1846	2058
JD(f)-LDA	2545	2649	2806
Spec-GP	101	107	111
Spec-LDA	107	140	193
TPM-GP	1734	2393	2726
TPM-LDA	12723	16460	19356

TABLE 3.1 – The running times in seconds of the algorithms from Figure 3-3, corresponds to the case when $N = 50,000$. Each algorithm was run 5 times, so the times in the table display the minimum (min), mean, and maximum (max) time.

is slightly higher than the one of the GP/DICA-cumulants (compare, e.g., the times for the spectral algorithm which almost completely consist of the computation of the moments/cumulants). Moreover, the runtime of TPM-LDA is significantly higher (half an hour vs. several hours) than the one of TPM-GP/DICA. This can be explained by the fact that the LDA-moments have more noise than the GP/DICA-cumulants and, hence, the convergence is slower. Interestingly, all versions of JD algorithm are not that sensitive to noise.

Computation of a whitening matrix is roughly 30 sec (this time is the same for all algorithms and is included in the numbers above).

3.5.4 The GP/DICA Cumulants vs. the LDA Moments

In Figure 3-4, when sampling from the *GP* model (top, left), both the GP/DICA cumulants and LDA moments are well specified, which implies that the approximation error (i.e., the error w.r.t. the model (mis)fit) is low for both. The GP/DICA cumulants achieve low values of the estimation error already for $N = 10,000$ documents independently of the number of topics, while the convergence is slower for the LDA moments. When sampling from the *LDA-fix(200)* model (top, right), the GP/DICA cumulants are then mis-specified and their approximation error is high, although the estimation error is low due to the faster finite sample convergence. One reason of poor performance of the GP/DICA cumulants, in this case, is the absence of variance in the document length. Indeed, if documents with two different lengths are mixed by sampling from the *LDA-fix2(0.5,20,200)* model (bottom, left), the GP/DICA cumulants performance improves. Moreover, the experiment with a changing fraction γ of documents (bottom, right) shows that a non-zero variance on the length improves the performance of the GP/DICA cumulants. As in practice real corpora usually have a non-zero variance for the document length, this bad scenario for the GP/DICA cumulants is not likely to happen.

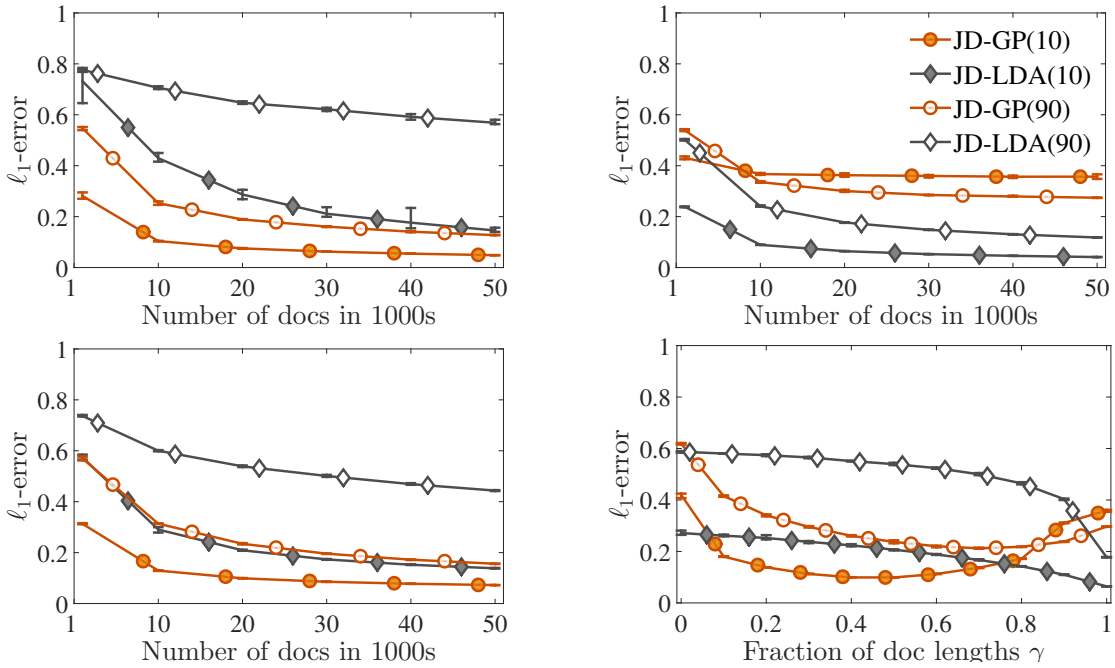


FIGURE 3-4 – Comparison of the GP/DICA cumulants and LDA moments. Two topic matrices and parameters $\mathbf{c}_1$ and $\mathbf{c}_2$ are learned from the NIPS dataset for $K = 10$ and 90; $\mathbf{c}_1$ and $\mathbf{c}_2$ are scaled to sum up to $c_0 = 1$. Four corpora of different sizes N from $1,000$ to $50,000$: *top, left :* b is set to fit the expected document length $\widehat{L} = 1300$; sampling from the *GP* model; *top, right :* sampling from the *LDA-fix(200)* model; *bottom, left :* sampling from the *LDA-fix2(0.5,20,200)* model. *Bottom, right :* the number of documents here is fixed to $N = 20,000$; sampling from the *LDA-fix2(γ,20,200)* model varying the values of the fraction γ from 0 to 1 with the step 0.1. *Note* : a smaller value of the ℓ_1-error is better.

The LDA Moments vs. Parameter $\mathbf{c_0}$

In this section, we experimentally investigate the dependence of the LDA moments on the parameter c_0. In Figure 3-5, the joint diagonalization algorithm with the LDA moment is compared for different values of c_0 provided to the algorithm. The data is generated similarly to Figure 3-4. The experiment indicates that the LDA moments are somewhat sensitive to the choice of c_0. For example, the recovery ℓ_1-error doubles when moving from the correct choice $c_0 = 1$ to an alternative $c_0 = 0.1$ for $K = 10$ on the *LDAfix(*200*)* dataset (JD-LDA(10) line on the right of Figure 3-5).

Comparison of the ℓ_1- and ℓ_2-Errors

The sample complexity results [Anandkumar et al., 2012a] for the spectral algorithm for the LDA moments allow straightforward extensions to the GP/DICA cumulants, if the results from Proposition 3.3.1 are taken into account. The analysis is, however, in terms of the ℓ_2-norm. Therefore, in Figure 3-6, we provide experimental comparison of the ℓ_1- and ℓ_2-errors to verify that they are indeed behaving similarly.

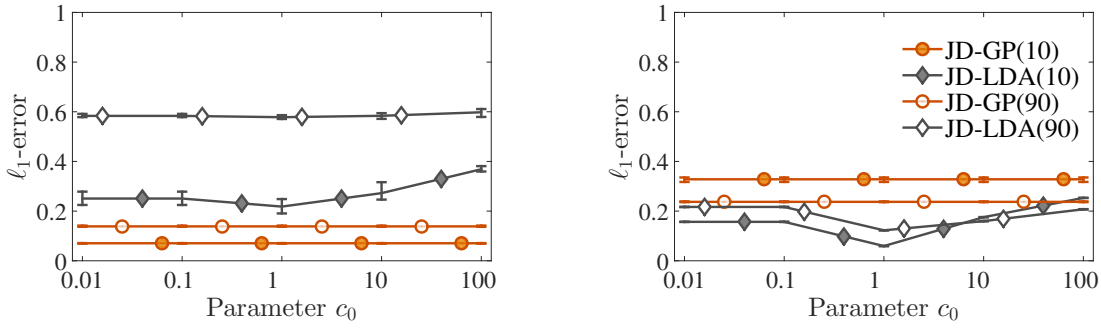


FIGURE 3-5 – Performance of the LDA moments depending on the parameter c_0. D and c are learned from the AP dataset for $K = 10$ and $K = 50$ and true $c_0 = 1$. JD-GP(10) for $K = 10$ and JD-GP(50) for $K = 50$. Number of sampled documents $N = 20,000$. For the error bars, each dataset is resampled 5 times. Data **(left)** : *GP* sampling ; **(right)** : *LDAfix(200)* sampling. *Note* : a smaller value of the ℓ_1-error is better.

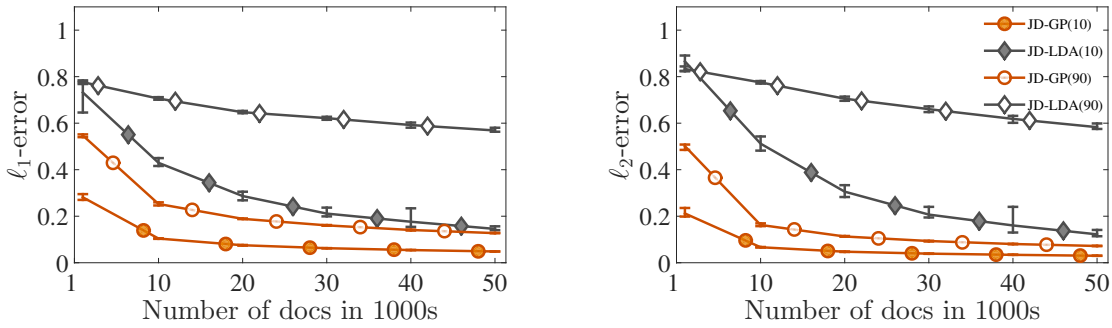


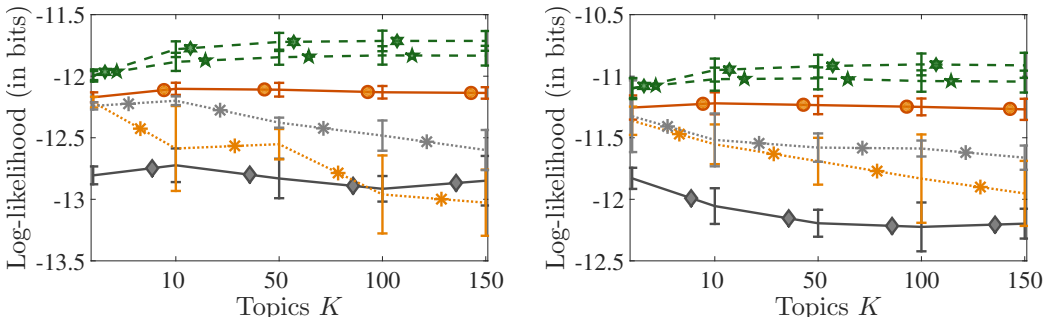
FIGURE 3-6 – Comparison of the ℓ_1- and ℓ_2-errors on the NIPS semi-synthetic dataset as in Figure 3-4 (top, left). The ℓ_2-norms of the topics were normalized to [0,1] for the computation of the ℓ_2 error.

3.5.5 Real Data Experiments

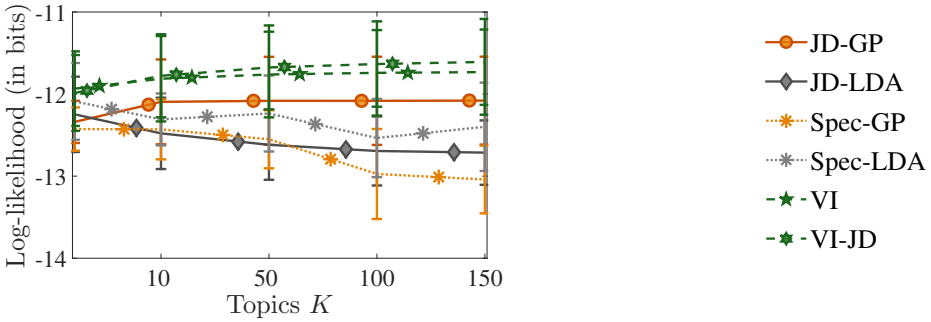
In this section, we compare the algorithms on the AP, KOS, and NIPS datasets.

Datasets. The detailed experimental setup is as follows. Each dataset is separated into 5 training/evaluation pairs, where the documents for evaluation are chosen randomly and non-repetitively among the folds (600 documents are held out for KOS ; 400 documents are held out for AP ; 450 documents are held out for NIPS). Then, the model parameters are learned for a different number of topics. The evaluation of the held-out documents is performed with averaging over 5 folds. In Figure 3-7a and Figure 3-7b, on the y-axis, the predictive log-likelihood in bits averaged per token is presented.

Note that, as the LDA moments require at least 3 tokens in each document, 1 document from the NIPS dataset and 3 documents from the AP dataset, which did not fulfill this requirement, were removed.



(a) The AP (left) and KOS (right) Datasets.



(b) The NIPS Dataset.

FIGURE 3-7 – Experiments with real data. *Note* : a higher value of the log-likelihood is better.

Algorithms. In Figure 3-7a, JD-GP, Spec-GP, JD-LDA, and Spec-LDA are compared with variational inference (VI) and with variational inference initialized with the output of JD-GP (VI-JD). We measure the held out log-likelihood per token. The orthogonal joint diagonalization algorithm with the GP/DICA cumulants (JD-GP) demonstrates promising performance. In particular, the GP/DICA cumulants significantly outperform the LDA moments. Moreover, although variational inference performs better than the JD-GP algorithm, restarting variational inference with the output of the JD-GP algorithm systematically leads to better results. Similar behavior has already been observed [see, e.g., Cohen and Collins, 2014].

Importantly, we observed that VI when initialized with the output of the JD-GP is consistently better in terms of the predictive log-likelihood. Therefore, the new algorithm can be used for more clever initialization of other LDA/GP inference methods.

We also observe that the joint diagonalization algorithm for the LDA moments is worse than the spectral algorithm. This indicates that the diagonal structure (2.43) and (2.44) might not be present in the sample estimates (2.45) and (2.46) due to either model misspecification or to finite sample complexity issues.

3.6 Conclusion

In this chapter, we have proposed a new set of tensors for a discrete ICA model related to LDA, where word counts are directly modeled. These moments make fewer assumptions regarding distributions, and are theoretically and empirically more robust than previously proposed tensors for LDA, both on synthetic and real data. Following the ICA literature, we showed that our joint diagonalization procedure is also more robust. Once the topic matrix has been estimated in a semiparametric way where topic intensities are left unspecified, it would be interesting to learn the unknown distributions of the independent topic intensities.

Chapitre 4

Moment Matching-Based Estimation in Multi-View Models

Abstract

Canonical correlation analysis (CCA), originally introduced by Hotelling [1936], and its probabilistic extension [e.g., Bach and Jordan, 2005], which we call *Gaussian CCA*, are widely used tools in applications with multi-view data, e.g., when working with text in several languages [e.g., Vinokourov et al., 2002] or in scene text recognition [e.g., Gordo, 2015]. Some extensions of Gaussian CCA are also applied, e.g., for mapping visual and textual features to the same latent space [e.g., Socher and Fei-Fei, 2010] or for machine translation [e.g., Haghighi et al., 2008].

In this chapter, we introduce a novel *semiparametric* extension of Gaussian CCA for multi-view models, which is able to model discrete data, count data, or a combination of discrete and count data that appear, e.g., in the applications mentioned above. We prove the essential identifiability of the new model, which also imply the identifiability of the discrete ICA model from Chapter 3. We first show that the higher-order cumulants of this model are in the form of the non-symmetric non-negative CPD and non-orthogonal joint diagonalization (NOJD) algorithms by congruence can be applied for the estimation. We further introduce *generalized covariance matrices*, which reduce the estimation problem in the model to the problem of (approximate) non-symmetric simultaneous eigendecomposition, which can be solved with the NOJD by congruence algorithms. The algorithms based on both higher-order cumulants and generalized covariance matrices demonstrate equivalent performance in terms of the solution quality, while dealing with the generalized covariance matrices is much easier than dealing with tensors, which significantly simplifies implementations. The content of this chapter was previously published as [Podosinnikova et al., 2016].

4.1 Contributions

Below we outline the contributions of this chapter.

- In Section 4.3, we introduce the novel *non-Gaussian CCA* model by relaxing the Gaussianity assumption on the sources in the Gaussian CCA model. We distinguish several special cases : (a) the discrete non-Gaussian CCA model, which is a direct extension of the discrete ICA model, to model multi-view count data and (b) the mixed CCA model, where one view models continuous data while the other models count data.
- In Section 4.4.1, we introduce cumulant-based higher order statistics of the discrete CCA model and show that their population variant has the form of the non-negative non-symmetric CPD. Moreover, in Section 4.4.2, we introduce the so called *generalized covariance matrices* which are Hessians of the cumulant generating function evaluated at some vector and are somewhat related to the contractions of third-order cumulants. In the population case, these matrices can jointly be reduced to the simultaneous non-symmetric eigendecomposition form. Importantly, the form of the generalized covariance matrices and higher-order cumulants is not dependent of the view specific noise and only requires the independence of the noise between the views.
- In Section 4.5, we show that the estimation in the non-Gaussian CCA models can be reduced to an (approximate) non-symmetric eigendecomposition problem which can be solved with the non-negative joint diagonalization algorithms by congruence. Note that this problem is difference from the (approximate) symmetric eigendecomposition problem and, in particular, such algorithms as non-orthogonal joint diagonalization by congruence or the prewhitening-based methods as in Section 3.4, can not be applied.
- In Section 4.6, we experimentally compare the new models and algorithms on synthetic and real datasets.

4.2 Related Work

The models introduced in this chapter are closely related to many other models. In particular, we have already mentioned (see Section 1.2.3 and 2.1.2) that probabilistic latent semantic indexing can actually be seen as a model for multi-view data [Hofmann, 1999a,b, Hofmann et al., 1999, Chew et al., 2007]. This is also similar to topic models for annotated data [Blei and Jordan, 2003]. The key difference of the models introduced in this chapter from such topic models for annotated data is that the non-Gaussian CCA models are semiparametric and do not make any assumptions on the distributions of the latent sources.

Some other extensions of Gaussian CCA were proposed in the literature : exponential family CCA [Virtanen, 2010, Klami et al., 2010] and Bayesian CCA [see, e.g., Klami et al., 2013, and references therein]. Although exponential family CCA can also be discretized, it assumes in practice that the prior of the sources is a special combination

of Gaussians. Bayesian CCA models the factor loading matrices and the covariance matrix of Gaussian CCA. Sampling or approximate variational inference are used for estimation and inference. Both models, however, lack our identifiability guarantees and are quite different from the non-Gaussian CCA models.

Note that non-Gaussian CCA is different from the independent subspace model. Another similar model — *multiset canonical correlation analysis* [Li et al., 2009] and similar models—where each data view is modeled as a separate ICA problem—is different from non-Gaussian CCA as well.

In the context of the estimation through the method of moments, Song et al. [2014] consider a multi-view framework to deal with non-parametric mixture components, while our approach is semi-parametric with an explicit linear structure (our loading matrices) and makes the explicit link with CCA. Moreover, the non-Gaussian CCA model is closely related to the model of Anandkumar et al. [2013b]. However, they consider the DAG point of view and, like Arora et al. [2012], they assume that the topic matrix follows the graph expansion property which is related to the anchor words assumption.

4.3 Non-Gaussian CCA

In Section 1.3.1, we review *classical CCA* for the so called multi-view or aligned data as well as its probabilistic interpretation in the form of *probabilistic CCA* graphical model [Browne, 1979, Bach and Jordan, 2005, Klami et al., 2013]. Similar to factor analysis and probabilistic PCA, this probabilistic CCA model has isotropic Gaussian latent variables and, therefore, is unidentifiable. In this section, we extend probabilistic CCA to a more general case of non-Gaussian latent variables and prove the *identifiability* of this model.

Likewise ICA (see Section 1.1.4) and discrete ICA, introduced in Section 3.3, this non-Gaussian CCA model is *semiparametric* and no assumptions on the latent sources are needed for the estimation. We also consider two special cases of discrete and mixed CCA, where the observations are in the form of count data or a combination of count and continuous data respectively. First, let us recall the main concepts from Section 1.3.1.

4.3.1 Non-Gaussian, Discrete, and Mixed CCA

Gaussian Canonical Correlation Analysis. We saw in Section 1.3.1, that the following model, which we will refer to as the *Gaussian CCA (GCCA)* model,

$$\begin{aligned} \mathbf{x}^{(1)}|\boldsymbol{\alpha} &\sim \mathcal{N}(\boldsymbol{\mu}_1 + \mathbf{D}_1\,\boldsymbol{\alpha}, \boldsymbol{\Psi}_1), \\ \mathbf{x}^{(2)}|\boldsymbol{\alpha} &\sim \mathcal{N}(\boldsymbol{\mu}_2 + \mathbf{D}_2\,\boldsymbol{\alpha}, \boldsymbol{\Psi}_2), \end{aligned} \tag{4.1}$$

where the independent sources $\boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$ are Gaussian and the matrices $\boldsymbol{\Psi}_1 \in \mathbb{R}^{M_1 \times M_1}$ and $\boldsymbol{\Psi}_2 \in \mathbb{R}^{M_2 \times M_2}$ are positive definite, can be seen as a probabilistic inter-

pretation of CCA. Indeed, the maximum likelihood estimators of the parameters $\mathbf{D}_1$ and $\mathbf{D}_2$ coincide with canonical correlation directions, up to permutation, scaling, and left-multiplication by *any invertible matrix*. Likewise factor analysis and probabilistic PCA, Gaussian CCA is *unidentifiable* due to the Gaussian prior on the sources.

By analogy with factor analysis, the GCCA model (4.1) can be equivalently represented with the following generative process

$$\begin{aligned} \mathbf{x}^{(1)} &= \boldsymbol{\mu}_1 + \mathbf{D}_1 \boldsymbol{\alpha} + \boldsymbol{\varepsilon}^{(1)}, \\ \mathbf{x}^{(2)} &= \boldsymbol{\mu}_2 + \mathbf{D}_2 \boldsymbol{\alpha} + \boldsymbol{\varepsilon}^{(2)}, \end{aligned} \tag{4.2}$$

where the additive noise vectors are normal random variables, $\boldsymbol{\varepsilon}^{(1)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}_1)$ and $\boldsymbol{\varepsilon}^{(2)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}_2)$, and are *independent* :

$$\begin{aligned} &\alpha_1, \ldots, \alpha_K \text{ are mutually independent,} \\ &\boldsymbol{\alpha} \perp\!\!\!\perp \boldsymbol{\varepsilon}^{(1)}, \boldsymbol{\varepsilon}^{(2)} \quad \text{and} \quad \boldsymbol{\varepsilon}^{(1)} \perp\!\!\!\perp \boldsymbol{\varepsilon}^{(2)}. \end{aligned} \tag{4.3}$$

The formulation in (4.2) indicates that GCCA is an extension of factor analysis to two views. The difference between the two models is that the CCA covariance matrices of the noise, $\boldsymbol{\Psi}_1$ and $\boldsymbol{\Psi}_2$, are not restricted to be diagonal as in factor analysis and hence the view-specific CCA noise may be *arbitrary correlated*. The only requirement is that there are *no correlations across the views*.

The Stacking Trick. More specifically, to see how GCCA is related to factor analysis, one can stack the view specific vectors (matrices) into a single vector (matrix) :

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \boldsymbol{\varepsilon}^{(1)} \\ \boldsymbol{\varepsilon}^{(2)} \end{pmatrix}, \tag{4.4}$$

which leads exactly to the generative model of factor analysis (1.7), i.e. $\mathbf{x} = \boldsymbol{\mu} + \mathbf{D}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}$, with the only difference of the assumptions on the additive noise. Indeed, in factor analysis the additive noise is zero-mean Gaussian variable with the diagonal covariance matrix $\boldsymbol{\Psi}$, while in GCCA the covariance of the zero-mean Gaussian additive noise is :

$$\boldsymbol{\Psi} = \begin{pmatrix} \boldsymbol{\Psi}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_2 \end{pmatrix}, \tag{4.5}$$

i.e. it has a block diagonal structure. Using such *stacking trick*, one can see that the marginal distribution of the observations under the GCCA model is the same as the one for factor analysis (1.6) with the covariance matrix of the noise in the form (4.5).

Non-Gaussian Canonical Correlation Analysis. ICA addresses the unidentifiability of factor analysis by relaxing the Gaussianity assumption. We use the same strategy of relaxing the Gaussianity assumption of the sources in the GCCA model

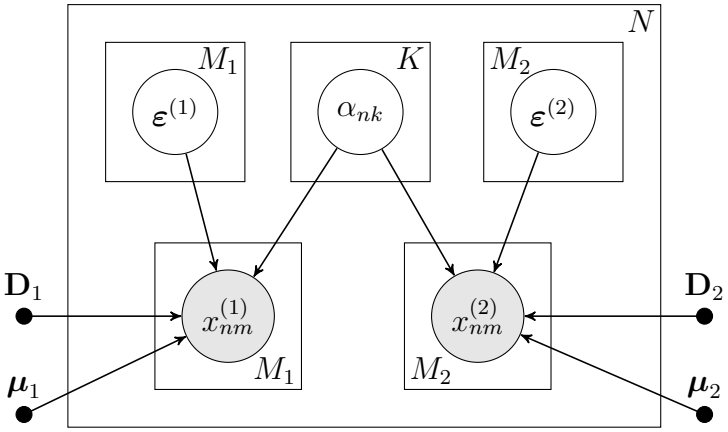


FIGURE 4-1 – The non-Gaussian CCA model.

to introduce the new model

$$\begin{aligned} \mathbf{x}^{(1)} &= \mathbf{D}_1 \boldsymbol{\alpha} + \boldsymbol{\varepsilon}^{(1)}, \\ \mathbf{x}^{(2)} &= \mathbf{D}_2 \boldsymbol{\alpha} + \boldsymbol{\varepsilon}^{(2)}, \end{aligned} \qquad \text{Non-Gaussian CCA} \quad (4.6)$$

where we only keep the independence assumption (4.3). We refer to this model as *non-Gaussian CCA (NCCA)* and illustrate it in Figure 4-1. Note that we put the constant shift parameters to zero, $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \mathbf{0}$, which can be done without loss of generality since in practice data can be centered to zero-mean.

Discrete Canonical Correlation Analysis. Similarly to discrete ICA from Section 3.3 we can further "discretize" non-Gaussian CCA (4.6) by applying the Poisson distribution to each view (independently on each variable) :

$$\begin{aligned} \mathbf{x}^{(1)} | \boldsymbol{\alpha}, \boldsymbol{\varepsilon}^{(1)} &\sim \text{Poisson}\big(\mathbf{D}_1 \boldsymbol{\alpha} + \boldsymbol{\varepsilon}^{(1)}\big), \\ \mathbf{x}^{(2)} | \boldsymbol{\alpha}, \boldsymbol{\varepsilon}^{(2)} &\sim \text{Poisson}\big(\mathbf{D}_2 \boldsymbol{\alpha} + \boldsymbol{\varepsilon}^{(2)}\big), \end{aligned} \qquad \text{Discrete CCA} \quad (4.7)$$

where again the independence (4.3) assumption on the latent sources is made. This gives us the (non-Gaussian) *discrete CCA (DCCA)* model, which is adapted to *count data* (e.g., such as word counts in the bag-of-words model of text). In this case, the sources $\boldsymbol{\alpha}$, the noise $\boldsymbol{\varepsilon}^{(1)}$ and $\boldsymbol{\varepsilon}^{(2)}$, and the matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ have non-negative components. Note that this model can be seen as a semiparametric extension of the multi-view topic models, i.e. topic models for annotated data [see, e.g, Blei and Jordan, 2003].

Noisy Discrete ICA. Using the stacking trick (4.4), it is straightforward to show that the discrete CCA model (4.7) is a special case of the following *noisy discrete ICA* model :

$$\mathbf{x} | \boldsymbol{\alpha} \sim \text{Poisson}(\mathbf{D}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}), \qquad \text{Noisy Discrete ICA} \quad (4.8)$$

where the sources $\boldsymbol{\alpha}$ are independent and the noise and sources are independent

$\boldsymbol{\alpha} \perp\!\!\!\perp \boldsymbol{\varepsilon}$. This model reduces to discrete ICA (3.8) when the noise $\boldsymbol{\varepsilon}$ tends to zero. When the noise $\boldsymbol{\varepsilon}$ takes the form

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \boldsymbol{\varepsilon}^{(1)} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varepsilon}^{(2)} \end{pmatrix}, \tag{4.9}$$

we obtain the discrete CCA model (4.7). This connection allows us to adapt the discrete ICA cumulant-based tensors from Section 3.3.2 to construct similar higher-order statistics for the discrete CCA model in Section 4.4.1.

Note that there are significant differences between discrete CCA (or non-Gaussian CCA) and discrete ICA (or respectively classical independent component analysis). One difference, as we will see later, is that the estimation in discrete (non-Gaussian) CCA can be performed not only in the presence of Gaussian noise,[1] but for noise with a more general structure as long as the view-specific noises, $\boldsymbol{\varepsilon}^{(1)}$ and $\boldsymbol{\varepsilon}^{(2)}$, are independent. More importantly, the estimation methods for discrete (non-Gaussian) CCA proposed in this chapter can also be applied in the *overcomplete* case. For example, let us consider the following linear discrete ICA model :

$$\begin{aligned} \mathbf{x}^{(1)}|\boldsymbol{\alpha}, \boldsymbol{\varepsilon}^{(1)} &\sim \text{Poisson}(\mathbf{D}_1\boldsymbol{\alpha} + \mathbf{F}_1\boldsymbol{\beta}^{(1)}), \\ \mathbf{x}^{(2)}|\boldsymbol{\alpha}, \boldsymbol{\varepsilon}^{(2)} &\sim \text{Poisson}(\mathbf{D}_2\boldsymbol{\alpha} + \mathbf{F}_2\boldsymbol{\beta}^{(2)}), \end{aligned} \tag{4.10}$$

where $\boldsymbol{\varepsilon}^{(1)} = \mathbf{F}_1\boldsymbol{\beta}^{(1)}$ and $\boldsymbol{\varepsilon}^{(2)} = \mathbf{F}_2\boldsymbol{\beta}^{(2)}$ and the independence assumption (4.3) takes place. By stacking variables we obtain the discrete ICA model $\mathbf{x}|\boldsymbol{\gamma} \sim \text{Poisson}(\mathbf{D}\boldsymbol{\gamma})$, where

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \end{pmatrix}, \quad \boldsymbol{\gamma} = \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\varepsilon}^{(1)} \\ \boldsymbol{\varepsilon}^{(2)} \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} \mathbf{D}_1 & \mathbf{F}_1 & \mathbf{0} \\ \mathbf{D}_2 & \mathbf{0} & \mathbf{F}_2 \end{pmatrix}.$$

The matrix $\mathbf{D} \in \mathbb{R}^{M \times K_0}$, where $M = M_1 + M_2$, $K_0 = K + K_1 + K_2$, and K_1 and K_2 are the numbers of columns in the matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ respectively. Discrete CCA can perform the estimation in such model even if $K_0 > M$ as long as $\mathbf{D}_1$ and $\mathbf{D}_2$ have full column rank. On the contrary, the cumulant-based estimation method for discrete ICA (see Section 3.4) can not work if $K_0 > M$ (see also the experiments in Section 4.6.1).

Mixed Canonical Correlation Analysis. Finally, by combining non-Gaussian and discrete CCA, we also introduce the *mixed CCA (MCCA)* model :

$$\begin{aligned} \mathbf{x}^{(1)} &= \mathbf{D}_1\boldsymbol{\alpha} + \boldsymbol{\varepsilon}^{(1)}, \\ \mathbf{x}^{(2)}|\boldsymbol{\alpha}, \boldsymbol{\varepsilon}^{(2)} &\sim \text{Poisson}\big(\mathbf{D}_2\boldsymbol{\alpha} + \boldsymbol{\varepsilon}^{(2)}\big), \end{aligned} \qquad \text{Mixed CCA} \tag{4.11}$$

which is adapted to a *combination of non-negative discrete and continuous* data (e.g., such as images represented as continuous vectors aligned with text represented as

1. Note that the cumulant-based estimation for ICA, as well as for the new models, can handle Gaussian noise with known covariance due to the properties of cumulants (see Section 2.2.1 or Comon and Jutten [2010]).

counts). Note that no assumptions are made on distributions of the sources $\boldsymbol{\alpha}$ except for independence (4.3), which makes the MCCA model, as well as the NCCA and GCCA models, *semiparametric*.

Both discrete and mixed CCA models can be seen as special cases of the non-Gaussian CCA model with a complicated noise structure. However, it is useful to distinguish these two cases from the practical point of view plus we can benefit from the special structure when constructing the algorithms. All the three models can be used with different applications, e.g., in machine translation of text data in different languages [Vinokourov et al., 2002, Haghighi et al., 2008] or in computer vision for annotated images or videos [Socher and Fei-Fei, 2010, Gordo, 2015].

4.3.2 Identifiability of Non-Gaussian CCA

In this section, the identifiability of the factor loading matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ is discussed. As before, the identifiability that we consider is the *essential identifiability*, i.e. the identifiability up to permutation and scaling.

ICA can be seen as an identifiable analog of factor analysis (see Section 1.1.4). Indeed, it is well known that ICA is identifiable if at most one source is Gaussian [Comon, 1994], while factor analysis is unidentifiable (see Section 1.1.2). Note that this identifiability results of ICA assume the linear independence of the columns of the mixing matrix which also implies that we are in the (over-)determined, a.k.a. (under-)complete, case ($K \leq M$). The other overcomplete case is more complicated and is outside of the scope of this chapter.

The factor loading matrices, $\mathbf{D}_1$ and $\mathbf{D}_2$, and the latent sources, $\boldsymbol{\alpha}$, of the Gaussian CCA model (4.1), which can be seen as a multi-view extension of PPCA, are identifiable only up to multiplication by any invertible matrix [Bach and Jordan, 2005]. We show the identifiability results for the new models (4.6), (4.7), and (4.11) : the factor loading matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ (hence the latent sources $\boldsymbol{\alpha}$) of these models are identifiable if at most one source is Gaussian (see the following Section 4.3.3 for a proof). These results also apply to the discrete ICA model (3.8) from Section 3.3

Theorem 4.3.1. *Assume that matrices* $\mathbf{D}_1 \in \mathbb{R}^{M_1 \times K}$ *and* $\mathbf{D}_2 \in \mathbb{R}^{M_2 \times K}$*, where* $K \leq \min(M_1, M_2)$*, have full rank. If the covariance matrices* $\operatorname{cov}\left(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}\right)$ *and* $\operatorname{cov}\left(\mathbf{x}^{(2)}, \mathbf{x}^{(2)}\right)$ *exist and if at most one source* α_k*, for* $k = 1, \ldots, K$*, is Gaussian and none of the sources are deterministic, then the models* (4.6)*,* (4.7)*, and* (4.11) *are identifiable (up to scaling and joint permutation).*

Importantly, the permutation unidentifiability does not destroy the alignment in the factor loading matrices, that is, for some permutation matrix $\boldsymbol{\Pi}$, if $\mathbf{D}_1\boldsymbol{\Pi}$ is the factor loading matrix of the first view, than $\mathbf{D}_2\boldsymbol{\Pi}$ must be the factor loading matrix of the second view. This property is important for the interpretability of the factor loading matrices and, in particular, is used in our experiments in Section 4.6.

4.3.3 The Proof of Theorem 4.3.1

In this section, we prove that the factor loading matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ of the non-Gaussian CCA (4.6), discrete CCA (4.7), and mixed CCA (4.11) models are identifiable up to permutation and scaling if at most one source α_k is Gaussian. We provide a complete proof for the non-Gaussian CCA case and show that the other two cases can be proved by analogy.

Identifiability of Non-Gaussian CCA (4.6)

The proof uses the notion of the *second characteristic function (SCF)* of an $\mathbb{R}^M$-valued random variable $\mathbf{x}$ (see Section 2.2) :

$$\boldsymbol{\phi}_{\mathbf{x}}(\mathbf{t}) = \log \mathbb{E}(e^{i\mathbf{t}^\top \mathbf{x}}),$$

for all $\mathbf{t} \in \mathbb{R}^M$. The SCF completely defines the probability distribution of $\mathbf{x}$. Important difference between the SCF and the cumulant generating function (2.22) is that the former always exists.

The following property of the SCF is of central importance for the proof : if two random variables, $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$, are independent, then $\boldsymbol{\phi}_{\mathbf{A}_1\mathbf{z}^{(1)}+\mathbf{A}_2\mathbf{z}^{(2)}}(\mathbf{t}) = \boldsymbol{\phi}_{\mathbf{z}^{(1)}}(\mathbf{A}_1^\top \mathbf{t}) + \boldsymbol{\phi}_{\mathbf{z}^{(2)}}(\mathbf{A}_2^\top \mathbf{t})$, where $\mathbf{A}_1$ and $\mathbf{A}_2$ are any matrices of compatible sizes.

We can now use our CCA model to derive an expression of $\boldsymbol{\phi}_{\mathbf{x}}(\mathbf{t})$, where $\mathbf{x}$ is the vector formed by stacking two vectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$. Indeed, defining a vector $\mathbf{x}$ by stacking the vectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, the SCF of $\mathbf{x}$ for any $\mathbf{t} = [\mathbf{t}_1;\ \mathbf{t}_2]$, takes the form

$$\begin{aligned}
\boldsymbol{\phi}_{\mathbf{x}}(\mathbf{t}) &= \log \mathbb{E}(e^{i\mathbf{t}_1^\top \mathbf{x}^{(1)} + i\mathbf{t}_2^\top \mathbf{x}^{(2)}}) \\
&\overset{(a)}{=} \log \mathbb{E}(e^{i\boldsymbol{\alpha}^\top(\mathbf{D}_1^\top \mathbf{t}_1 + \mathbf{D}_2^\top \mathbf{t}_2) + i\mathbf{t}_1^\top \boldsymbol{\varepsilon}^{(1)} + i\mathbf{t}_2^\top \boldsymbol{\varepsilon}^{(2)}}) \\
&\overset{(b)}{=} \log \mathbb{E}(e^{i\boldsymbol{\alpha}^\top(\mathbf{D}_1^\top \mathbf{t}_1 + \mathbf{D}_2^\top \mathbf{t}_2)}) \\
&\qquad + \log \mathbb{E}(e^{i\mathbf{t}_1^\top \boldsymbol{\varepsilon}^{(1)}}) + \log \mathbb{E}(e^{i\mathbf{t}_2^\top \boldsymbol{\varepsilon}^{(2)}}) \\
&= \boldsymbol{\phi}_{\boldsymbol{\alpha}}(\mathbf{D}_1^\top \mathbf{t}_1 + \mathbf{D}_2^\top \mathbf{t}_2) + \boldsymbol{\phi}_{\boldsymbol{\varepsilon}^{(1)}}(\mathbf{t}_1) + \boldsymbol{\phi}_{\boldsymbol{\varepsilon}^{(2)}}(\mathbf{t}_2),
\end{aligned}$$

where in (a) we substituted the definition (4.6) of $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ and in (b) we used the independence $\boldsymbol{\alpha} \perp\!\!\!\perp \boldsymbol{\varepsilon}^{(1)} \perp\!\!\!\perp \boldsymbol{\varepsilon}^{(2)}$. Therefore, the blockwise mixed derivatives of $\boldsymbol{\phi}_{\mathbf{x}}$ are equal to

$$\partial_1 \partial_2 \boldsymbol{\phi}_{\mathbf{x}}(\mathbf{t}) = \mathbf{D}_1 \boldsymbol{\phi}''_{\boldsymbol{\alpha}}(\mathbf{D}_1^\top \mathbf{t}_1 + \mathbf{D}_2^\top \mathbf{t}_2)\mathbf{D}_2^\top, \tag{4.12}$$

where $\partial_1 \partial_2 \boldsymbol{\phi}_{\mathbf{x}}(\mathbf{t}) := \nabla_{\mathbf{t}_1} \nabla_{\mathbf{t}_2} \boldsymbol{\phi}_{\mathbf{x}}(\mathbf{h}(\mathbf{t}_1, \mathbf{t}_2)) \in \mathbb{R}^{M_1 \times M_2}$ and $\boldsymbol{\phi}''_{\boldsymbol{\alpha}}(\mathbf{u}) := \nabla^2_{\mathbf{u}} \boldsymbol{\phi}_{\boldsymbol{\alpha}}(\mathbf{u})$, does not depend on the noise vectors $\boldsymbol{\varepsilon}^{(1)}$ and $\boldsymbol{\varepsilon}^{(2)}$. Note that we denoted $\mathbf{h}(\mathbf{t}_1, \mathbf{t}_2) = \mathbf{D}_1^\top \mathbf{t}_1 + \mathbf{D}_2^\top \mathbf{t}_2$.

For simplicity, we first prove the identifiability result when all components of the common sources are non-Gaussian. The high level idea of the proof is as follows. We assume two different representations of $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ and using (4.12) and the independence of the components of $\boldsymbol{\alpha}$ and the noises (assumption (1.22)), we first

show that the two potential dictionaries are related by an orthogonal matrix (and not any invertible matrix), and then show that this implies that the two potential sets of independent components are (orthogonal) linear combinations of each other, which, for non-Gaussian components which are not reduced to point masses, imposes that this orthogonal transformation is the combination of a permutation matrix and marginal scaling—a standard result from the ICA literature [Comon, 1994, Theorem 11].

Let us then assume that two equivalent representations of non-Gaussian CCA exist :

$$\begin{aligned} \mathbf{x}^{(1)} &= \mathbf{D}_1\boldsymbol{\alpha} + \boldsymbol{\varepsilon}^{(1)} = \mathbf{E}_1\boldsymbol{\beta} + \boldsymbol{\eta}^{(1)}, \\ \mathbf{x}^{(2)} &= \mathbf{D}_2\boldsymbol{\alpha} + \boldsymbol{\varepsilon}^{(2)} = \mathbf{E}_2\boldsymbol{\beta} + \boldsymbol{\eta}^{(2)}, \end{aligned} \tag{4.13}$$

where the other sources $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)^\top$ are also assumed mutually independent and non-degenerate. As a standard practice in the ICA literature and without loss of generality as the sources have non-degenerate components, one can assume that the sources have unit variances, i.e. $\mathrm{cov}(\boldsymbol{\alpha}, \boldsymbol{\alpha}) = I$ and $\mathrm{cov}(\boldsymbol{\beta}, \boldsymbol{\beta}) = I$, by respectively rescaling the columns of the factor loading matrices. Under this assumption, the two expressions of the cross-covariance matrix are

$$\mathrm{cov}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \mathbf{D}_1\mathbf{D}_2^\top = \mathbf{E}_1\mathbf{E}_2^\top, \tag{4.14}$$

which, given that $\mathbf{D}_1$, $\mathbf{D}_2$ have full rank, implies that [2]

$$\mathbf{E}_1 = \mathbf{D}_1\mathbf{M}, \quad \mathbf{E}_2 = \mathbf{D}_2\mathbf{M}^{-\top}, \tag{4.15}$$

where $\mathbf{M} \in \mathbb{R}^{K\times K}$ is some invertible matrix. Substituting the representations (4.13) into the blockwise mixed derivatives of the SCF (4.12) and using the expressions (4.15) give

$$\begin{aligned} &\mathbf{D}_1\boldsymbol{\phi}''_{\boldsymbol{\alpha}}(\mathbf{D}_1^\top\mathbf{t}_1 + \mathbf{D}_2^\top\mathbf{t}_2)\mathbf{D}_2^\top \\ &\quad = \mathbf{D}_1\mathbf{M}\boldsymbol{\phi}''_{\boldsymbol{\beta}}(\mathbf{M}^\top\mathbf{D}_1^\top\mathbf{t}_1 + \mathbf{M}^{-1}\mathbf{D}_2^\top\mathbf{t}_2)\mathbf{M}^{-1}\mathbf{D}_2^\top, \end{aligned}$$

for all $\mathbf{t}_1 \in \mathbb{R}^{M_1}$ and $\mathbf{t}_2 \in \mathbb{R}^{M_2}$. Since the matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ have full rank, this can be rewritten as

$$\begin{aligned} &\boldsymbol{\phi}''_{\boldsymbol{\alpha}}(\mathbf{D}_1^\top\mathbf{t}_1 + \mathbf{D}_2^\top\mathbf{t}_2) \\ &\quad = \mathbf{M}\boldsymbol{\phi}''_{\boldsymbol{\beta}}(\mathbf{M}^\top\mathbf{D}_1^\top\mathbf{t}_1 + \mathbf{M}^{-1}\mathbf{D}_2^\top\mathbf{t}_2)\mathbf{M}^{-1}, \end{aligned}$$

which holds for all $\mathbf{t}_1 \in \mathbb{R}^{M_1}$ and $\mathbf{t}_2 \in \mathbb{R}^{M_2}$. Moreover, still since $\mathbf{D}_1$ and $\mathbf{D}_2$ have full rank, we have, for any $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^K$ the existence of $\mathbf{t}_1 \in \mathbb{R}^{M_1}$ and $\mathbf{t}_2 \in \mathbb{R}^{M_2}$, such that $\mathbf{u}_1 = \mathbf{D}_1^\top\mathbf{t}_1$ and $\mathbf{u}_2 = \mathbf{D}_2^\top\mathbf{t}_2$, that is,

$$\boldsymbol{\phi}''_{\boldsymbol{\alpha}}(\mathbf{u}_1 + \mathbf{u}_2) = \mathbf{M}\boldsymbol{\phi}''_{\boldsymbol{\beta}}(\mathbf{M}^\top\mathbf{u}_1 + \mathbf{M}^{-1}\mathbf{u}_2)\mathbf{M}^{-1}, \tag{4.16}$$

for all $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^K$.

2. The fact that $\mathbf{D}_1$, $\mathbf{D}_2$ have full rank and that $\mathbf{E}_1$, $\mathbf{E}_2$ have K columns, combined with (4.14), implies that $\mathbf{E}_1$, $\mathbf{E}_2$ have also full rank.

We will now prove two facts :

(F1) For any vector $\mathbf{v} \in \mathbb{R}^K$, then $\boldsymbol{\phi}''_{\boldsymbol{\beta}}((\mathbf{M}^\top\mathbf{M} - \mathbf{I})\mathbf{v}) = -\mathbf{I}$, which will imply that $\mathbf{M}\mathbf{M}^\top = \mathbf{I}$ because of the non-Gaussian assumptions.

(F2) If $\mathbf{M}\mathbf{M}^\top = \mathbf{I}$, then $\boldsymbol{\phi}''_{\boldsymbol{\alpha}}(\mathbf{u}) = \boldsymbol{\phi}''_{\mathbf{M}\boldsymbol{\beta}}(\mathbf{u})$ for any $\mathbf{u} \in \mathbb{R}^K$, which will imply that $\mathbf{M}$ is the composition of a permutation and a scaling. This will end the proof.

Proof of fact (F1). By letting $\mathbf{u}_1 = \mathbf{M}\mathbf{v}$ and $\mathbf{u}_2 = -\mathbf{M}\mathbf{v}$, we get :

$$\boldsymbol{\phi}''_{\boldsymbol{\alpha}}(\mathbf{0}) = \mathbf{M}\boldsymbol{\phi}''_{\boldsymbol{\beta}}((\mathbf{M}^\top\mathbf{M} - \mathbf{I})\mathbf{v})\mathbf{M}^{-1}, \tag{4.17}$$

Since [3] $\boldsymbol{\phi}''_{\boldsymbol{\alpha}}(\mathbf{0}) = -\mathrm{cov}(\boldsymbol{\alpha}) = -\mathbf{I}$, one gets

$$\boldsymbol{\phi}''_{\boldsymbol{\beta}}((\mathbf{M}^\top\mathbf{M} - \mathbf{I})\mathbf{v}) = -\mathbf{I},$$

for any $\mathbf{v} \in \mathbb{R}^K$.

Using the property that $\boldsymbol{\phi}''_{\mathbf{A}^\top\boldsymbol{\beta}}(\mathbf{v}) = \mathbf{A}^\top\boldsymbol{\phi}''_{\boldsymbol{\beta}}(\mathbf{A}\mathbf{v})\mathbf{A}$ for any matrix $\mathbf{A}$, and in particular with $\mathbf{A} = \mathbf{M}^\top\mathbf{M} - \mathbf{I}$, we have that $\boldsymbol{\phi}''_{\mathbf{A}^\top\boldsymbol{\beta}}(\mathbf{v}) = -\mathbf{A}^\top\mathbf{A}$, i.e. is constant.

If the second derivative of a function is constant, the function is quadratic. Therefore, $\boldsymbol{\phi}_{\mathbf{A}^\top\boldsymbol{\beta}}(\cdot)$ is a quadratic function. Since the SCF completely defines the distribution of its variable (see, e.g., Jacod and Protter [2004]), $\mathbf{A}^\top\boldsymbol{\beta}$ must be Gaussian (the SCF of a Gaussian random variable is a quadratic function). Given Lemma 9 from Comon [1994] (i.e., Cramer's lemma : a linear combination of non-Gaussian random variables cannot be Gaussian unless the coefficients are all zero), this implies that $\mathbf{A} = \mathbf{0}$, and hence $\mathbf{M}^\top\mathbf{M} = \mathbf{I}$, i.e., $\mathbf{M}$ is an orthogonal matrix.

Proof of fact (F2). Plugging $\mathbf{M}^\top = \mathbf{M}^{-1}$ into (4.16), with $\mathbf{u}_1 = 0$ and $\mathbf{u}_2 = u$, gives

$$\boldsymbol{\phi}''_{\boldsymbol{\alpha}}(\mathbf{u}) = \mathbf{M}\boldsymbol{\phi}''_{\boldsymbol{\beta}}(\mathbf{M}^\top\mathbf{u})\mathbf{M}^\top = \boldsymbol{\phi}''_{\mathbf{M}\beta}(\mathbf{u}), \tag{4.18}$$

for any $\mathbf{u} \in \mathbb{R}^K$. By integrating both sides of (4.18) and using $\boldsymbol{\phi}_{\boldsymbol{\alpha}}(\mathbf{0}) = \boldsymbol{\phi}_{\mathbf{M}\boldsymbol{\beta}}(\mathbf{0}) = 0$, we get that $\boldsymbol{\phi}_{\boldsymbol{\alpha}}(\mathbf{u}) = \boldsymbol{\phi}_{\mathbf{M}\boldsymbol{\beta}}(\mathbf{u}) + i\boldsymbol{\gamma}^\top\mathbf{u}$ for all $\mathbf{u} \in \mathbb{R}^K$ for some constant vector $\boldsymbol{\gamma}$. Using again that the SCF completely defines the distribution, it follows that $\boldsymbol{\alpha} - \boldsymbol{\gamma}$ and $\mathbf{M}\boldsymbol{\beta}$ have the same distribution. Since both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ have independent components, this is only possible when $\mathbf{M} = \boldsymbol{\Lambda}\mathbf{P}$, where $\mathbf{P}$ is a permutation matrix and $\boldsymbol{\Lambda}$ is some diagonal matrix [Comon, 1994, Theorem 11].

Case of a Single Gaussian Source

Without loss of generality, we assume that the potential Gaussian source is the first one for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The first change is in the proof of fact (F1). We use the same argument up to the point where we conclude that $\mathbf{A}^\top\boldsymbol{\beta}$ is a Gaussian vector. As only $\boldsymbol{\beta}_1$ can be Gaussian, Cramer's lemma implies that only the first row of $\mathbf{A}$ can have non-zero components, that is $\mathbf{A} = \mathbf{M}^\top\mathbf{M} - \mathbf{I} = \mathbf{e}_1\mathbf{f}^\top$, where $\mathbf{e}_1$ is the first basis

3. Note that $\nabla^2_{\mathbf{u}}\boldsymbol{\phi}_{\boldsymbol{\alpha}}(\mathbf{u}) = -\frac{\mathbb{E}(\boldsymbol{\alpha}\boldsymbol{\alpha}^\top e^{i\mathbf{u}^\top\boldsymbol{\alpha}})}{\mathbb{E}(e^{i\mathbf{u}^\top\boldsymbol{\alpha}})} + \boldsymbol{\mathcal{E}}_{\boldsymbol{\alpha}}(\mathbf{u})\boldsymbol{\mathcal{E}}_{\boldsymbol{\alpha}}(\mathbf{u})^\top$, where $\boldsymbol{\mathcal{E}}_{\boldsymbol{\alpha}}(\mathbf{u}) = \frac{\mathbb{E}(\boldsymbol{\alpha}e^{i\mathbf{u}^\top\boldsymbol{\alpha}})}{\mathbb{E}(e^{i\mathbf{u}^\top\boldsymbol{\alpha}})}$.

vector and $\mathbf{f}$ any vector. Since $\mathbf{M}^\top\mathbf{M}$ is symmetric, we must have

$$\mathbf{M}^\top\mathbf{M} = \mathbf{I} + a\mathbf{e}_1\mathbf{e}_1^\top,$$

where a is a constant scalar different than -1 as $\mathbf{M}^\top\mathbf{M}$ is invertible. This implies that $\mathbf{M}^\top\mathbf{M}$ is an invertible diagonal matrix $\boldsymbol{\Lambda}$, and hence $\mathbf{M}\boldsymbol{\Lambda}^{-1/2}$ is an orthogonal matrix, which in turn implies that $\mathbf{M}^{-1} = \boldsymbol{\Lambda}^{-1}\mathbf{M}^\top$.

Plugging this into (4.16) gives, for any $\mathbf{u}_1$ and $\mathbf{u}_2$:

$$\boldsymbol{\phi}''_{\boldsymbol{\alpha}}(\mathbf{u}_1 + \mathbf{u}_2) = \mathbf{M}\boldsymbol{\phi}''_{\boldsymbol{\beta}}(\mathbf{M}^\top\mathbf{u}_1 + \boldsymbol{\Lambda}^{-1}\mathbf{M}^\top\mathbf{u}_2)\boldsymbol{\Lambda}^{-1}\mathbf{M}^\top.$$

Given that diagonal matrices commute and that $\boldsymbol{\phi}''_{\boldsymbol{\beta}}$ is diagonal for independent sources, this leads to

$$\boldsymbol{\phi}''_{\boldsymbol{\alpha}}(\mathbf{u}_1 + \mathbf{u}_2) = \mathbf{M}\boldsymbol{\Lambda}^{-1/2}\boldsymbol{\phi}''_{\boldsymbol{\beta}}(\mathbf{M}^\top\mathbf{u}_1 + \boldsymbol{\Lambda}^{-1}\mathbf{M}^\top\mathbf{u}_2)\boldsymbol{\Lambda}^{-1/2}\mathbf{M}^\top.$$

For any given $\mathbf{v} \in \mathbb{R}^K$, we are looking for $\mathbf{u}_1$ and $\mathbf{u}_2$ such that $\mathbf{M}^\top\mathbf{u}_1 + \boldsymbol{\Lambda}^{-1}\mathbf{M}^\top\mathbf{u}_2 = \boldsymbol{\Lambda}^{-1/2}\mathbf{M}^\top\mathbf{v}$ and $\mathbf{u}_1 + \mathbf{u}_2 = \mathbf{v}$, which is always possible by setting $\mathbf{M}^\top\mathbf{u}_2 = (\boldsymbol{\Lambda}^{-1/2} + \mathbf{I})^{-1}\mathbf{M}^\top\mathbf{v}$ and $\mathbf{M}^\top\mathbf{u}_1 = \mathbf{M}^\top\mathbf{v} - \mathbf{M}^\top\mathbf{u}_2$ by using the special structure of $\boldsymbol{\Lambda}$. Thus, for any $\mathbf{v}$,

$$\boldsymbol{\phi}''_{\boldsymbol{\alpha}}(\mathbf{v}) = \mathbf{M}\boldsymbol{\Lambda}^{-1/2}\boldsymbol{\phi}''_{\boldsymbol{\beta}}(\boldsymbol{\Lambda}^{-1/2}\mathbf{M}^\top\mathbf{v})\boldsymbol{\Lambda}^{-1/2}\mathbf{M}^\top = \boldsymbol{\phi}''_{\mathbf{M}\boldsymbol{\Lambda}^{-1/2}\boldsymbol{\beta}}(\mathbf{v}).$$

Integrating as previously, this implies that the characteristic function of $\boldsymbol{\alpha}$ and $\mathbf{M}\boldsymbol{\Lambda}^{-1/2}\boldsymbol{\beta}$ differ only by a linear function $i\boldsymbol{\gamma}^\top\mathbf{v}$, and thus, that $\boldsymbol{\alpha} - \boldsymbol{\gamma}$ and $\mathbf{M}\boldsymbol{\Lambda}^{-1/2}\boldsymbol{\beta}$ have the same distribution. This in turn, from Comon [1994, Theorem 11], implies that $\mathbf{M}\boldsymbol{\Lambda}^{-1/2}$ is a product of a scaling and a permutation, which ends the proof.

Identifiability of Discrete CCA (4.7) and Mixed CCA (4.11)

Given the discrete CCA model, the SCF $\boldsymbol{\phi}_{\mathbf{x}}(\mathbf{t})$ takes the form

$$\begin{aligned}\boldsymbol{\phi}_{\mathbf{x}}(\mathbf{t}) = \boldsymbol{\phi}_{\boldsymbol{\alpha}}(\mathbf{D}_1^\top(e^{i\mathbf{t}_1} - 1) + \mathbf{D}_2^\top(e^{i\mathbf{t}_2} - 1)) \\ + \boldsymbol{\phi}_{\boldsymbol{\varepsilon}^{(1)}}(e^{i\mathbf{t}_1} - 1) + \boldsymbol{\phi}_{\boldsymbol{\varepsilon}^{(2)}}(e^{i\mathbf{t}_2} - 1),\end{aligned}$$

where $e^{i\mathbf{t}_j}$, for $j = 1, 2$, denotes a vector with the m-th element equal to $e^{i[\mathbf{t}_j]_m}$, and we used the arguments analogous with the non-Gaussian case. The rest of the proof extends with a correction that sometimes one has to replace $\mathbf{D}_j$ with $\mathrm{Diag}[e^{i\mathbf{t}_j}]\mathbf{D}_j$ and that $\mathbf{u}_j = \mathbf{D}_j^\top(e^{i\mathbf{t}_j} - 1)$ for $j = 1, 2$. For the mixed CCA case, only the part related to $\mathbf{x}^{(2)}$ and $\mathbf{D}_2$ changes in the same way as for the discrete CCA case.

4.4 Cumulants and Generalized Covariance Matrices

In this section, we first derive the cumulant-based tensors for the discrete CCA model (Section 4.4.1). We show that the population versions of these higher-order statistics take the form of a *non-symmetric canonical polyadic decomposition* (see Section 2.1.2).

This is different from Chapter 3, where the decomposition of the cumulant-based tensor of discrete ICA was instead *symmetric*. Nevertheless, this allows us to develop fast algorithms for the estimation in DCCA model (see Section 4.5).

We further introduce *generalized covariance matrices* (Section 4.4.2) for all the new models (4.6), (4.7), and (4.11). We show that these generalized covariance matrices have a special diagonal form, which is equivalent to non-symmetric CP decomposition in the matrix case. This allows us to develop fast algorithms for the estimation in the new models (Section 4.5). The use of generalized covariance matrices allows us to achieve the results equivalent to the ones based on tensors while working with matrices, which simplifies the derivations tremendously. This makes the generalized covariance matrices an attractive tool in the moment matching framework.

4.4.1 Discrete CCA Cumulants

In this section, we derive the DCCA cumulant-based higher-order statistics as an extension of the discrete ICA cumulant-based statistics derived in Section 3.3.2.

Discrete ICA Cumulants. In Section 3.3, we introduced the discrete ICA model (3.8), where the observation vector $\mathbf{x} \in \mathbb{R}^M$ has conditionally independent Poisson components with mean $\mathbf{D}\boldsymbol{\alpha}$ and the latent sources vector $\boldsymbol{\alpha} \in \mathbb{R}^K$ has independent non-negative components :

$$\mathbf{x}|\boldsymbol{\alpha} \sim \text{Poisson}(\mathbf{D}\boldsymbol{\alpha}), \tag{4.19}$$

where for the estimation of the topic matrix $\mathbf{D}$, we proposed an algorithm based on the moment matching method with the DICA cumulants. In Section 3.3.2, we defined these DICA cumulants : the $\mathbf{S}^{DICA}$-covariance matrix and $\boldsymbol{\mathcal{T}}^{DICA}$-cumulant tensor, as

$$\begin{aligned} \mathbf{S}^{DICA} &:= \text{cov}(\mathbf{x}, \mathbf{x}) - \text{Diag}\left[\mathbb{E}(\mathbf{x})\right], \\ \mathcal{T}^{DICA}_{m_1 m_2 m_3} &:= \text{cum}(x_{m_1}, x_{m_2}, x_{m_3}) + \tau_{m_1 m_2 m_3}, \end{aligned} \tag{4.20}$$

where the indices m_1, m_2, and m_3 take the values in $1, \ldots, M$, and

$$\begin{aligned} \tau_{m_1 m_2 m_3} = &\, 2\delta\left(m_1, m_2, m_3\right) \mathbb{E}(x_{m_1}) - \delta\left(m_2 m_3\right) \text{cov}(x_{m_1}, x_{m_2}) \\ &- \delta\left(m_1, m_3\right) \text{cov}(x_{m_1}, x_{m_2}) - \delta\left(m_1, m_2\right) \text{cov}(x_{m_1}, x_{m_3}), \end{aligned}$$

where δ stands for the Kronecker delta. For completeness, we outline the derivation from Section 3.3.2 below. Let $\mathbf{y} := \mathbf{D}\boldsymbol{\alpha}$. By the law of total expectation $\mathbb{E}(\mathbf{x}) = \mathbb{E}(\mathbf{x}|\mathbf{y}) = \mathbb{E}(\mathbf{y})$ and by the law of total covariance

$$\begin{aligned} \text{cov}(\mathbf{x}, \mathbf{x}) &= \mathbb{E}[\text{cov}(\mathbf{x}, \mathbf{x}|\mathbf{y})] + \text{cov}[\mathbb{E}(\mathbf{x}|\mathbf{y}),\ \mathbb{E}(\mathbf{x}|\mathbf{y})] \\ &= \text{Diag}[\mathbb{E}(\mathbf{y})] + \text{cov}(\mathbf{y}, \mathbf{y}), \end{aligned}$$

since all the cumulants of a Poisson random variable with parameter $\mathbf{y}$ are equal to $\mathbf{y}$. Therefore, $\mathbf{S}^{DICA} = \text{cov}(\mathbf{y}, \mathbf{y})$. Similarly, by the law of total cumulance $\boldsymbol{\mathcal{T}}^{DICA} =$

$\text{cum}(\mathbf{y}, \mathbf{y}, \mathbf{y})$. Then, by the multilinearity property for cumulants, one obtains

$$\begin{aligned} \mathbf{S}^{DICA} &= \mathbf{D}\,\text{cov}(\boldsymbol{\alpha}, \boldsymbol{\alpha})\mathbf{D}^\top, \\ \boldsymbol{\mathcal{T}}^{DICA} &= \text{cum}(\boldsymbol{\alpha}, \boldsymbol{\alpha}, \boldsymbol{\alpha}) \times_1 \mathbf{D}^\top \times_2 \mathbf{D}^\top \times_3 \mathbf{D}^\top, \end{aligned} \tag{4.21}$$

where $\times_i$ denotes the i-mode tensor-matrix product (see Section 2.1.1). Since the covariance $\text{cov}(\boldsymbol{\alpha}, \boldsymbol{\alpha})$ and cumulant $\text{cum}(\boldsymbol{\alpha}, \boldsymbol{\alpha}, \boldsymbol{\alpha})$ of the independent sources are diagonal, (4.21) is called the *diagonal form*. The expressions (4.21) are also known as the symmetric non-negative CP decomposition of tensors (see Section 2.1.2) ; it is non-negative since the columns of the topic matrix are non-negative.

Noisy Discrete ICA Cumulants. We showed that the noisy DICA model (4.8) contains both discrete CCA and discrete ICA models as special cases. It is straightforward to obtain the cumulants of the noisy DICA model. Let $\mathbf{y} := \mathbf{D}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}$ and $\mathbf{S}^{NDICA} := \mathbf{S}^{DICA}$ and $\boldsymbol{\mathcal{T}}^{NDICA} := \boldsymbol{\mathcal{T}}^{DICA}$ are defined as in (4.20). Then a simple extension of the derivations from above gives $\mathbf{S}^{NDICA} = \text{cov}(\mathbf{y}, \mathbf{y})$ and $\boldsymbol{\mathcal{T}}^{NDICA} = \text{cum}(\mathbf{y}, \mathbf{y}, \mathbf{y})$. Since the covariance matrix (cumulant tensor) of the sum of two independent multivariate random variables, $\mathbf{D}\boldsymbol{\alpha}$ and $\boldsymbol{\varepsilon}$, is equal to the sum of the covariance matrices (cumulant tensors) of these variables, the "perturbed" version of the decomposition in (4.21) follows

$$\begin{aligned} \mathbf{S}^{NDICA} &= \mathbf{D}\,\text{cov}(\boldsymbol{\alpha}, \boldsymbol{\alpha})\mathbf{D}^\top + \text{cov}(\boldsymbol{\varepsilon}), \\ \boldsymbol{\mathcal{T}}^{NDICA} &= \text{cum}(\boldsymbol{\alpha}, \boldsymbol{\alpha}, \boldsymbol{\alpha}) \times_1 \mathbf{D}^\top \times_2 \mathbf{D}^\top \times_3 \mathbf{D}^\top + \text{cum}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}). \end{aligned} \tag{4.22}$$

Note that this expressions are essentially equivalent to the ones for the noisy cumulants of noisy ICA model from Section 2.2.2.

DCCA Cumulants. By the stacking trick (4.4), discrete CCA (4.7) gives a noisy version of discrete ICA with a special form of the covariance matrix of the noise :

$$\text{cov}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}) = \begin{pmatrix} \text{cov}\big(\boldsymbol{\varepsilon}^{(1)}, \boldsymbol{\varepsilon}^{(1)}\big) & \mathbf{0} \\ \mathbf{0} & \text{cov}\big(\boldsymbol{\varepsilon}^{(2)}, \boldsymbol{\varepsilon}^{(2)}\big) \end{pmatrix}, \tag{4.23}$$

which is due to the independence $\boldsymbol{\varepsilon}^{(1)} \perp\!\!\!\perp \boldsymbol{\varepsilon}^{(2)}$. Similarly, the cumulant $\text{cum}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon})$ of the noise has only two diagonal blocks which are non-zero. Therefore, considering only those parts of the $\mathbf{S}^{NDICA}$-covariance matrix and $\boldsymbol{\mathcal{T}}^{NCCA}$-cumulant tensor of noisy DICA that correspond to zero blocks of the covariance $\text{cov}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon})$ and cumulant $\text{cum}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon})$, gives immediately a matrix and tensor with a diagonal structure similar to the one in (4.21). Those blocks are the cross-covariance and cross-cumulants of $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$.

We define the $\mathbf{S}^{DCCA}$-*covariance matrix of discrete CCA* [4] as the cross-covariance matrix of $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$:

$$\mathbf{S}_{12}^{DCCA} := \text{cov}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}). \tag{4.24}$$

4. Note that $\mathbf{S}_{21}^{DCCA} := \text{cov}(\mathbf{x}^{(2)}, \mathbf{x}^{(1)})$ is just the transpose of $\mathbf{S}_{12}^{DCCA}$.

From (4.22) and (4.23), the matrix $\mathbf{S}_{12}^{DCCA}$ has the following diagonal form

$$\mathbf{S}_{12}^{DCCA} = \mathbf{D}_1 \operatorname{cov}(\boldsymbol{\alpha}, \boldsymbol{\alpha}) \mathbf{D}_2^\top. \tag{4.25}$$

Similarly, we define the $\boldsymbol{\mathcal{T}}^{DCCA}$*-cumulant tensors of discrete CCA*, $M_1 \times M_2 \times M_1$ tensor $\boldsymbol{\mathcal{T}}_{121}^{DCCA}$ and $M_1 \times M_2 \times M_2$ tensor $\boldsymbol{\mathcal{T}}_{122}^{DCCA}$, through the cross-cumulants of $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, for $j = 1, 2$:

$$\left[\boldsymbol{\mathcal{T}}_{12j}^{DCCA}\right]_{m_1 m_2 \tilde{m}_j} := \operatorname{cum}\left(x_{m_1}^{(1)}, x_{m_2}^{(2)}, x_{m_j}^{(j)}\right) - \delta\left(m_j, \tilde{m}_j\right) \operatorname{cov}\left(x_{m_1}^{(1)}, x_{m_2}^{(2)}\right), \tag{4.26}$$

where the indices m_1, m_2, and $\tilde{m}_j$ take the values $m_1 \in 1, \ldots, M_1$, $m_2 \in 1, \ldots, M_2$, and $\tilde{m}_j \in 1, \ldots, M_j$. From (4.21) and the mentioned block structure (4.23) of $\operatorname{cov}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon})$, the DCCA $\boldsymbol{\mathcal{T}}^{DCCA}$-cumulants have the diagonal form :

$$\begin{aligned} \boldsymbol{\mathcal{T}}_{121}^{DCCA} &= \operatorname{cum}(\boldsymbol{\alpha}, \boldsymbol{\alpha}, \boldsymbol{\alpha}) \times_1 \mathbf{D}_1^\top \times_2 \mathbf{D}_2^\top \times_3 \mathbf{D}_1^\top, \\ \boldsymbol{\mathcal{T}}_{122}^{DCCA} &= \operatorname{cum}(\boldsymbol{\alpha}, \boldsymbol{\alpha}, \boldsymbol{\alpha}) \times_1 \mathbf{D}_1^\top \times_2 \mathbf{D}_2^\top \times_3 \mathbf{D}_2^\top. \end{aligned} \tag{4.27}$$

In Section 4.5, we show how to estimate the factor loading matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ using the diagonal form (4.25) and (4.27). Before that, in Section 4.4.2, we first derive the generalized covariance matrices of discrete ICA and the CCA models (4.6), (4.7), and (4.11) as an extension of ideas by Yeredor [2000], Todros and Hero [2013].

4.4.2 Generalized Covariance Matrices

In this section, we introduce the generalization of the **S**-covariance matrix for both DICA and the CCA models (4.6), (4.7), and (4.11), which are obtained through the Hessian of the cumulant generating function. We show that (a) the generalized covariance matrices can be used for approximation of the $\boldsymbol{\mathcal{T}}$-cumulant tensors using directional derivatives and (b) in the DICA case, these generalized covariance matrices have the diagonal form analogous to (4.21), and, in the CCA case, they have the diagonal form analogous to (4.25). Therefore, generalized covariance matrices can be seen as a substitute for the $\boldsymbol{\mathcal{T}}$-cumulant tensors in the moment matching framework. This (a) significantly simplifies derivations and the final expressions used for implementation of resulting algorithms and (b) potentially improves the sample complexity, since only the second-order information is used.

Generalized Covariance Matrices

The idea of generalized covariance matrices is inspired by the similar extension of the ICA cumulants by Yeredor [2000].

Cumulants and the Cumulant-Generating Function. We review some of the notions introduced in Section 2.2. The cumulant generating function (CGF) of a multivariate random variable $x \in \mathbb{R}^M$ is defined as :

$$\mathbf{K}_{\mathbf{x}}(\mathbf{t}) = \log \mathbb{E}(e^{\mathbf{t}^\top \mathbf{x}}), \tag{4.28}$$

for $\mathbf{t} \in \mathbb{R}^M$. The cumulants $\boldsymbol{\kappa}_{\mathbf{x}}^{(s)}$, for $s = 1, 2, 3, \ldots$, are the coefficients of the Taylor series expansion of the CGF evaluated at zero. Therefore, the cumulants are the derivatives of the CGF evaluated at zero : $\boldsymbol{\kappa}_{\mathbf{x}}^{(s)} = \nabla^s \mathbf{K}_{\mathbf{x}}(\mathbf{0})$, $s = 1, 2, 3, \ldots$, where $\nabla^s \mathbf{K}_{\mathbf{x}}(\mathbf{t})$ is the s-th order derivative of $\mathbf{K}_{\mathbf{x}}(\mathbf{t})$ with respect to $\mathbf{t}$. Thus, the expectation of $\mathbf{x}$ is the gradient $\mathbb{E}(\mathbf{x}) = \nabla \mathbf{K}_{\mathbf{x}}(\mathbf{0})$ and the covariance of $\mathbf{x}$ is the Hessian $\text{cov}(\mathbf{x}, \mathbf{x}) = \nabla^2 \mathbf{K}_{\mathbf{x}}(\mathbf{0})$ of the CGF evaluated at zero.

Generalized Cumulants. The extension of cumulants then follows immediately : for $\mathbf{t} \in \mathbb{R}^M$, we refer to the derivatives $\nabla^s \mathbf{K}_{\mathbf{x}}(\mathbf{t})$ of the CGF as the *generalized cumulants*. The respective parameter $\mathbf{t}$ is called a *processing point*. In particular, the gradient, $\nabla \mathbf{K}_{\mathbf{x}}(\mathbf{t})$, and Hessian, $\nabla^2 \mathbf{K}_{\mathbf{x}}(\mathbf{t})$, of the CGF are referred to as the *generalized expectation* and *generalized covariance matrix*, respectively :

$$\boldsymbol{\mathcal{E}}_x(t) := \nabla \mathbf{K}_{\mathbf{x}}(\mathbf{t}) = \frac{\mathbb{E}\big(\mathbf{x} e^{\mathbf{t}^\top \mathbf{x}}\big)}{\mathbb{E}\big(e^{\mathbf{t}^\top \mathbf{x}}\big)}, \tag{4.29}$$

$$\boldsymbol{\mathcal{C}}_x(t) := \nabla^2 \mathbf{K}_{\mathbf{x}}(\mathbf{t}) = \frac{\mathbb{E}\big(\mathbf{x}\mathbf{x}^\top e^{\mathbf{t}^\top \mathbf{x}}\big)}{\mathbb{E}\big(e^{\mathbf{t}^\top \mathbf{x}}\big)} - \boldsymbol{\mathcal{E}}_{\mathbf{x}}(\mathbf{t}) \boldsymbol{\mathcal{E}}_{\mathbf{x}}(\mathbf{t})^\top. \tag{4.30}$$

Finite Sample Estimators of Generalized Cumulants. Following Yeredor [2000], Slapak and Yeredor [2012b], we use the most direct way of defining finite sample estimators of the generalized expectation (4.29) and covariance matrix (4.30). Given a finite sample $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, an estimator of the generalized expectation is

$$\widehat{\boldsymbol{\mathcal{E}}}_{\mathbf{x}}(\mathbf{t}) = \frac{\sum_{n=1}^N \mathbf{x}_n w_n}{\sum_{n=1}^N w_n} \tag{4.31}$$

where weights $w_n = e^{\mathbf{t}^\top \mathbf{x}_n}$ and an estimator of the generalized covariance is

$$\widehat{\boldsymbol{\mathcal{C}}}_{\mathbf{x}}(\mathbf{t}) = \frac{\sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^\top w_n}{\sum_{n=1}^N w_n} - \widehat{\boldsymbol{\mathcal{E}}}_{\mathbf{x}}(\mathbf{t}) \widehat{\boldsymbol{\mathcal{E}}}_{\mathbf{x}}(\mathbf{t})^\top. \tag{4.32}$$

Similarly, an estimator of the generalized **S**-covariance matrix is then

$$\widehat{\boldsymbol{\mathcal{C}}}_{\mathbf{x}^{(1)},\mathbf{x}^{(2)}}(\mathbf{t}) = \frac{\sum_{n=1}^N \mathbf{x}_n^{(1)} \mathbf{x}_n^{(2)\top} w_n}{\sum_{n=1}^N w_n} - \frac{\sum_{n=1}^N \mathbf{x}_n^{(1)} w_n}{\sum_{n=1}^N w_n} \frac{\sum_{n=1}^N \mathbf{x}_n^{(2)\top} w_n}{\sum_{n=1}^N w_n},$$

where $\mathbf{x} = \big[\mathbf{x}^{(1)}; \mathbf{x}^{(2)}\big]$ and $\mathbf{t} = \big[\mathbf{t}_1; \mathbf{t}_2\big]$ for some $\mathbf{t}_1 \in \mathbb{R}^{M_1}$ and $\mathbf{t}_2 \in \mathbb{R}^{M_2}$. Note that some properties of the generalized expectation (4.29) and covariance matrix [5] (4.30) and their finite sample estimators (4.31) and (4.32) are analyzed by Slapak and Yeredor [2012b].

Generalized Covariance Matrix is Diagonal for an Independent Vector. Likewise cumulants, the generalized cumulants are diagonal if a vector is independent.

5. Note that we find the name "generalized covariance matrix" to be more meaningful than "charrelation" matrix as was proposed by previous authors [see, e.g. Slapak and Yeredor, 2012a,b].

We show this in particular for the generalized expectation and covariance. Indeed, the sources $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ are mutually independent. Therefore, for some $\mathbf{h} \in \mathbb{R}^K$, their CGF (4.28) $\mathbf{K}_{\boldsymbol{\alpha}}(\mathbf{h}) = \log \mathbb{E}(e^{\boldsymbol{\alpha}^\top \mathbf{h}})$ takes the form

$$\mathbf{K}_{\boldsymbol{\alpha}}(\mathbf{h}) = \sum_{k=1}^{K} \log \left[\mathbb{E}(e^{\alpha_k h_k})\right].$$

Therefore, the k-th element of the generalized expectation (4.29) of $\boldsymbol{\alpha}$ is (separable in α_k) :

$$[\boldsymbol{\mathcal{E}}_{\boldsymbol{\alpha}}(\mathbf{h})]_k = \frac{\mathbb{E}(\alpha_k e^{\alpha_k h_k})}{\mathbb{E}(e^{\alpha_k h_k})}, \tag{4.33}$$

and the generalized covariance (4.30) of $\boldsymbol{\alpha}$ is diagonal due to the separability and its k-th diagonal element is :

$$[\boldsymbol{\mathcal{C}}_{\boldsymbol{\alpha}}(\mathbf{h})]_{kk} = \frac{\mathbb{E}(\alpha_k^2 e^{\alpha_k h_k})}{\mathbb{E}(e^{\alpha_k h_k})} - [\boldsymbol{\mathcal{E}}_{\boldsymbol{\alpha}}(\mathbf{h})]_k^2 . \tag{4.34}$$

Likewise covariance matrices, these Hessians (a.k.a. generalized covariance matrices) are subject to the multilinearity property for a linear transformations of a vector, hence the resulting diagonal structure of the form (4.21). This is essentially the previous ICA work [Yeredor, 2000, Todros and Hero, 2013]. Below we generalize these ideas first to the discrete ICA case and then to the CCA models (4.6), (4.7), and (4.11).

Discrete ICA Generalized Covariance Matrices

Likewise covariance matrices, generalized covariance matrices of a vector with independent components are diagonal : they satisfy the multilinearity property $\boldsymbol{\mathcal{C}}_{\mathbf{D}\boldsymbol{\alpha}}(\mathbf{h}) = \mathbf{D}\,\boldsymbol{\mathcal{C}}_{\boldsymbol{\alpha}}(\mathbf{h})\mathbf{D}^\top$, and are equal to covariance matrices when $\mathbf{h} = \mathbf{0}$. Therefore, we can expect that the derivations of the diagonal form (4.21) of the $\mathbf{S}$-covariance matrices extends to the generalized covariance matrices case. By analogy with (4.20), we define the *generalized* $\mathbf{S}$*-covariance matrix* of DICA :

$$\mathbf{S}^{DICA}(\mathbf{t}) := \boldsymbol{\mathcal{C}}_{\mathbf{x}}(\mathbf{t}) - \mathrm{Diag}[\boldsymbol{\mathcal{E}}_{\mathbf{x}}(\mathbf{t})]. \tag{4.35}$$

To derive the analog of the diagonal form (4.21) for $\mathbf{S}^{DICA}(\mathbf{t})$, we have to compute all the expectations in (4.29) and (4.30) for a Poisson random variable $\mathbf{x}$ with the parameter $\mathbf{y} = \mathbf{D}\boldsymbol{\alpha}$. In the derivations below we use the fact that the moment generating function of a Poisson random variable x with the parameter y has a very special form $\mathbf{M}_x(t) = y(e^t - 1)$ and therefore all the cumulants of x are equal to y.

Given the discrete ICA model (3.8), the generalized expectation (4.29) of $\mathbf{x} \in \mathbb{R}^M$

takes the form

$$\begin{aligned}\boldsymbol{\mathcal{E}}_{\mathbf{x}}(\mathbf{t}) &= \frac{\mathbb{E}(\mathbf{x}e^{\mathbf{t}^\top \mathbf{x}})}{\mathbb{E}(e^{\mathbf{t}^\top \mathbf{x}})} = \frac{\mathbb{E}\left[\mathbb{E}(\mathbf{x}e^{\mathbf{t}^\top \mathbf{x}}|\boldsymbol{\alpha})\right]}{\mathbb{E}\left[\mathbb{E}(e^{\mathbf{t}^\top \mathbf{x}}|\boldsymbol{\alpha})\right]} \\ &= \mathrm{Diag}[e^{\mathbf{t}}]\mathbf{D}\frac{\mathbb{E}(\boldsymbol{\alpha}e^{\boldsymbol{\alpha}^\top \mathbf{h}(\mathbf{t})})}{\mathbb{E}(e^{\boldsymbol{\alpha}^\top \mathbf{h}(\mathbf{t})})} \\ &= \mathrm{Diag}[e^{\mathbf{t}}]\mathbf{D}\boldsymbol{\mathcal{E}}_{\boldsymbol{\alpha}}(\mathbf{h}(\mathbf{t})),\end{aligned}$$

where $\mathbf{t} \in \mathbb{R}^M$ is a parameter, $\mathbf{h}(\mathbf{t}) = \mathbf{D}^\top(e^{\mathbf{t}} - \mathbf{1})$, and $e^{\mathbf{t}}$ denotes an M-vector with the m-th element equal to e^{t_m}. Note that in the last equation we used the definition (4.29) of the generalized expectation $\boldsymbol{\mathcal{E}}_{\boldsymbol{\alpha}}(\cdot)$.

Further, the generalized covariance (4.30) of $\mathbf{x}$ takes the form

$$\begin{aligned}\boldsymbol{\mathcal{C}}_{\mathbf{x}}(\mathbf{t}) &= \frac{\mathbb{E}(\mathbf{x}\mathbf{x}^\top e^{\mathbf{t}^\top \mathbf{x}})}{\mathbb{E}(e^{\mathbf{t}^\top \mathbf{x}})} - \boldsymbol{\mathcal{E}}_{\mathbf{x}}(\mathbf{t})\boldsymbol{\mathcal{E}}_{\mathbf{x}}(\mathbf{t})^\top \\ &= \frac{\mathbb{E}\left[\mathbb{E}(\mathbf{x}\mathbf{x}^\top e^{\mathbf{t}^\top \mathbf{x}}|\boldsymbol{\alpha})\right]}{\mathbb{E}\left[\mathbb{E}(e^{\mathbf{t}^\top \mathbf{x}}|\boldsymbol{\alpha})\right]} - \boldsymbol{\mathcal{E}}_{\mathbf{x}}(\mathbf{t})\boldsymbol{\mathcal{E}}_{\mathbf{x}}(\mathbf{t})^\top.\end{aligned}$$

Plugging into this expression the expression for $\boldsymbol{\mathcal{E}}_{\mathbf{x}}(\mathbf{t})$ and

$$\begin{aligned}\mathbb{E}(\mathbf{x}\mathbf{x}^\top e^{\mathbf{t}^\top \mathbf{x}}|\boldsymbol{\alpha}) = \mathrm{Diag}[e^{\mathbf{t}}]\mathbf{D}\mathbb{E}(\boldsymbol{\alpha}\boldsymbol{\alpha}^\top e^{\boldsymbol{\alpha}^\top \mathbf{h}(\mathbf{t})})\mathbf{D}^\top \mathrm{Diag}[e^{\mathbf{t}}] \\ + \mathrm{Diag}[e^{\mathbf{t}}]\mathrm{Diag}\left[\mathbf{D}\mathbb{E}(\boldsymbol{\alpha}e^{\boldsymbol{\alpha}^\top \mathbf{h}(\mathbf{t})})\right]\end{aligned}$$

we get

$$\boldsymbol{\mathcal{C}}_{\mathbf{x}}(\mathbf{t}) = \mathrm{Diag}[\boldsymbol{\mathcal{E}}_{\mathbf{x}}(\mathbf{t})] + \mathrm{Diag}[e^{\mathbf{t}}]\mathbf{D}\boldsymbol{\mathcal{C}}_{\boldsymbol{\alpha}}(\mathbf{h}(\mathbf{t}))\mathbf{D}^\top \mathrm{Diag}[e^{\mathbf{t}}],$$

where we used the definition (4.30) of the generalized covariance of $\boldsymbol{\alpha}$. This gives

$$\mathbf{S}^{DICA}(\mathbf{t}) = \left(\mathrm{Diag}[e^{\mathbf{t}}]\mathbf{D}\right)\, \mathcal{C}_{\boldsymbol{\alpha}}\left(\mathbf{h}(\mathbf{t})\right)\, \left(\mathrm{Diag}[e^{\mathbf{t}}]\mathbf{D}\right)^\top, \tag{4.36}$$

which is a diagonal form similar (and equivalent for $\mathbf{t} = \mathbf{0}$) to (4.21) since the generalized covariance matrix $\boldsymbol{\mathcal{C}}_{\boldsymbol{\alpha}}(\mathbf{h})$ of independent sources is diagonal (see Equation (4.34)). Therefore, the generalized DICA $\mathbf{S}$-covariance matrices, estimated at different processing points $\mathbf{t}$, can be used as a substitute of the contractions of the DICA $\boldsymbol{\mathcal{T}}$-cumulant tensors in the moment matching framework. Interestingly enough, the latter can be approximated by the former via the directional derivatives, as we now show.

Approximating DICA $\mathcal{T}$-Cumulants with a Generalized Covariance Matrix. Let $f_{mm'}(\mathbf{t}) = [\boldsymbol{\mathcal{C}}_{\mathbf{x}}(\mathbf{t})]_{mm'}$ be a function $\mathbb{R} \to \mathbb{R}^M$ corresponding to the (m, m')-th element of the generalized covariance matrix. Then the following holds for its directional derivative at $\mathbf{t}_0$ along the direction $\mathbf{t}$:

$$\langle \nabla f_{mm'}(\mathbf{t}_0), \mathbf{t}\rangle = \lim_{\delta \to 0} \frac{f_{mm'}(\mathbf{t}_0 + \delta\mathbf{t}) - f_{mm'}(\mathbf{t}_0)}{\delta},$$

where $\langle \cdot, \cdot \rangle$ stands for the inner product. Therefore, when using the fact that $\nabla f(\mathbf{t}_0) = \nabla \boldsymbol{\mathcal{C}}_{\mathbf{x}}(\mathbf{t}_0)$ is the generalized order-3 cumulant of $\mathbf{x}$ at $\mathbf{t}_0$ and the definition of the contraction of a tensor $\boldsymbol{\mathcal{T}}$ with a vector $\mathbf{t}$, defined element-wise as $[\boldsymbol{\mathcal{T}}(\mathbf{t})]_{m_1 m_2} = \sum_{m_3=1}^{M} \mathcal{T}_{m_1 m_2 m_3} t_{m_3}$, one obtains for $\mathbf{t}_0 = \mathbf{0}$ the approximation of the cumulant $\operatorname{cum}(\mathbf{x}, \mathbf{x}, \mathbf{x})$ with the generalized covariance matrix $\boldsymbol{\mathcal{C}}_{\mathbf{x}}(\mathbf{t})$. Indeed, in such case $\langle \nabla f_{mm'}(\mathbf{t}_0), \mathbf{t} \rangle$ becomes the (m, m')-th element of the cumulant $\operatorname{cum}(\mathbf{x}, \mathbf{x}, \mathbf{x})$ contracted with $\mathbf{t}$, while the terms on the RHS are $f_{mm'}(\delta \mathbf{t}) = \boldsymbol{\mathcal{C}}_{\mathbf{x}}(\delta \mathbf{t})$ and $f_{mm'}(\mathbf{0}) = \boldsymbol{\mathcal{C}}_{\mathbf{x}}(\mathbf{0}) = \operatorname{cov}(\mathbf{x}, \mathbf{x})$.

CCA Generalized Covariance Matrices

For the CCA models (4.6), (4.7), and (4.11), straightforward generalizations of the ideas from Section 2.2.1 leads to the following definition of the *generalized CCA* **S**-*covariance matrix* :

$$\mathbf{S}_{12}(\mathbf{t}) := \frac{\mathbb{E}\big(\mathbf{x}^{(1)} \mathbf{x}^{(2)\top} e^{\mathbf{t}^\top \mathbf{x}}\big)}{\mathbb{E}\big(e^{\mathbf{t}^\top \mathbf{x}}\big)} - \frac{\mathbb{E}\big(\mathbf{x}^{(1)} e^{\mathbf{t}^\top \mathbf{x}}\big)}{\mathbb{E}\big(e^{\mathbf{t}^\top \mathbf{x}}\big)} \frac{\mathbb{E}\big(\mathbf{x}^{(2)\top} e^{\mathbf{t}^\top \mathbf{x}}\big)}{\mathbb{E}\big(e^{\mathbf{t}^\top \mathbf{x}}\big)}, \tag{4.37}$$

where the vectors $\mathbf{x}$ and $\mathbf{t}$ are obtained by vertically stacking $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ and $\mathbf{t}_1$ and $\mathbf{t}_2$ as in the stacking trick (4.4). We first briefly show that the generalized CCA **S**-cumulants contain equivalent information to the one from contracted CCA $\boldsymbol{\mathcal{T}}$-cumulants, which is similar to the discrete ICA case above. We then prove the diagonal form of the generalized CCA **S**-cumulants for the mixed, non-Gaussian, and discrete CCA models.

Approximating CCA $\boldsymbol{\mathcal{T}}$-Cumulants with a Generalized Covariance Matrix. Let us define $\mathbf{v}_1 = \mathbf{W}_1^\top \mathbf{u}_1$ and $\mathbf{v}_2 = \mathbf{W}_2^\top \mathbf{u}_2$ for some $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^K$. Then, approximations for the $\boldsymbol{\mathcal{T}}$-cumulants (4.26) of discrete CCA take the following form : $\mathbf{W}_1 \boldsymbol{\mathcal{T}}_{121}(\mathbf{v}_1) \mathbf{W}_2$ is approximated by the generalized **S**-covariances (4.37) $\mathbf{S}_{12}(\mathbf{t})$ via the following expression

$$\begin{aligned} \mathbf{W}_1 \boldsymbol{\mathcal{T}}_{121}(\mathbf{v}_1) \mathbf{W}_2 \approx & \frac{\mathbf{W}_1 \mathbf{S}_{12}(\delta \mathbf{t}_1) \mathbf{W}_2^\top - \mathbf{W}_1 \mathbf{S}_{12}(\mathbf{0}) \mathbf{W}_2^\top}{\delta} \\ & - \mathbf{W}_1 \operatorname{Diag}(\mathbf{v}_1) \mathbf{S}_{12} \mathbf{W}_2^\top, \end{aligned}$$

where $\mathbf{t}_1 = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{0} \end{pmatrix}$ and $\mathbf{W}_1 \boldsymbol{\mathcal{T}}_{122}(\mathbf{v}_2) \mathbf{W}_2$ is approximated by the generalized **S**-covariances $\mathbf{S}_{12}(\mathbf{t})$ via

$$\begin{aligned} \mathbf{W}_1 \boldsymbol{\mathcal{T}}_{122}(\mathbf{v}_2) \mathbf{W}_2 \approx & \frac{\mathbf{W}_1 \mathbf{S}_{12}(\delta \mathbf{t}_2) \mathbf{W}_2^\top - \mathbf{W}_1 \mathbf{W}_{12}(\mathbf{0}) \mathbf{W}_2^\top}{\delta} \\ & - \mathbf{W}_1 \mathbf{S}_{12} \operatorname{Diag}(\mathbf{v}_2) \mathbf{W}_2^\top, \end{aligned}$$

where $\mathbf{t}_2 = \begin{pmatrix} \mathbf{0} \\ \mathbf{v}_2 \end{pmatrix}$ and δ are chosen to be small.

Mixed CCA. To show the diagonal form, similar to the one in (4.36), of the generalized CCA **S**-cumulants, we present the detailed proof for the mixed CCA case ; the other two cases are contained in this proof and therefore are omitted. The CGF (4.28)

of mixed CCA (4.11) can be written as

$$\begin{aligned}
\mathbf{K}_{\mathbf{x}}(\mathbf{t}) &= \log \mathbb{E}\big(e^{\mathbf{t}^{(1)\top}\mathbf{x}^{(1)}+\mathbf{t}^{(2)\top}\mathbf{x}^{(2)}}\big) \\
&= \log \mathbb{E}\left[\mathbb{E}\big(e^{\mathbf{t}^{(1)\top}\mathbf{x}^{(1)}+\mathbf{t}^{(2)\top}\mathbf{x}^{(2)}}\big|\boldsymbol{\alpha}, \boldsymbol{\varepsilon}^{(1)}, \boldsymbol{\varepsilon}^{(2)}\big)\right] \\
&\overset{(a)}{=} \log \mathbb{E}\left[\mathbb{E}\big(e^{\mathbf{t}^{(1)\top}\mathbf{x}^{(1)}}\big|\boldsymbol{\alpha}, \boldsymbol{\varepsilon}^{(1)}\big)\mathbb{E}\big(e^{\mathbf{t}^{(2)\top}\mathbf{x}^{(2)}}|\boldsymbol{\alpha}, \boldsymbol{\varepsilon}^{(2)}\big)\right] \\
&\overset{(b)}{=} \log \mathbb{E}\left(e^{\mathbf{t}^{(1)\top}\left(\mathbf{D}_1\boldsymbol{\alpha}+\boldsymbol{\varepsilon}^{(1)}\right)}e^{\left[\mathbf{D}_2\boldsymbol{\alpha}+\boldsymbol{\varepsilon}^{(2)\top}(e^{\mathbf{t}_2}-1)\right]}\right) \\
&\overset{(c)}{=} \log \mathbb{E}\left(e^{\boldsymbol{\alpha}^\top\mathbf{h}(\mathbf{t})}\right) + \log \mathbb{E}\left(e^{\boldsymbol{\varepsilon}^{(2)\top}(e^{\mathbf{t}_2}-1)}\right) + \log \mathbb{E}(e^{\mathbf{t}^{(1)\top}\boldsymbol{\varepsilon}^{(1)}}),
\end{aligned}$$

where $\mathbf{h}(\mathbf{t}) = \mathbf{D}_1^\top\mathbf{t}_1 + \mathbf{D}_2^\top(e^{\mathbf{t}_2}-1)$, in (a) we used the conditional independence of $\mathbf{x}^{(2)}$ and $\mathbf{x}^{(2)}$, in (b) we used that $\mathbb{E}(e^{\mathbf{t}^\top\mathbf{x}}) = e^{\mathbf{y}^\top(e^{\mathbf{t}}-\mathbf{1})}$, and in (c) we used the independence assumption (4.3).

The generalized CCA $\mathbf{S}$-covariance matrix is defined (equivalent to (4.37)) as

$$\mathbf{S}_{12}(\mathbf{t}) := \nabla_{\mathbf{t}_2}\nabla_{\mathbf{t}_1}\mathbf{K}_{\mathbf{x}}(\mathbf{t}).$$

Its gradient with respect to $\mathbf{t}_1$ is

$$\nabla_{\mathbf{t}_1}\mathbf{K}_{\mathbf{x}}(\mathbf{t}) = \frac{\mathbf{D}_1\mathbb{E}\big(\boldsymbol{\alpha}e^{\boldsymbol{\alpha}^\top\mathbf{h}(\mathbf{t})}\big)}{\mathbb{E}\big(e^{\boldsymbol{\alpha}^\top\mathbf{h}(\mathbf{t})}\big)} + \frac{\mathbb{E}\big(\boldsymbol{\varepsilon}^{(1)}e^{\mathbf{t}^{(1)\top}\boldsymbol{\varepsilon}^{(1)}}\big)}{\mathbb{E}\big(e^{\mathbf{t}^{(1)\top}\boldsymbol{\varepsilon}^{(1)}}\big)},$$

where the last term does not depend on $\mathbf{t}_2$. Computing the gradient of this expression with respect to $\mathbf{t}_2$ gives

$$\mathbf{S}_{12}^{MCCA}(\mathbf{t}) = \mathbf{D}_1\mathcal{C}_{\boldsymbol{\alpha}}(\mathbf{h}(\mathbf{t}))\left(\mathrm{Diag}\big[e^{\mathbf{t}_2}\big]\mathbf{D}_2\right)^\top,$$

where we substituted expression (4.34) for the generalized covariance of the independent sources. Straightforward extension of this argument leads to similar expressions for discrete and non-Gaussian CCA. We summarize below these expressions.

Discrete, Non-Gaussian, and Mixed Generalized Covariance. In the discrete CCA case, $\mathbf{S}_{12}^{DCCA}(\mathbf{t})$ is essentially the upper-right block of the generalized $\mathbf{S}$-covariance matrix $\mathbf{S}^{DICA}(\mathbf{t})$ of DICA and has the form

$$\mathbf{S}_{12}^{DCCA}(\mathbf{t}) = \left(\mathrm{Diag}[e^{\mathbf{t}_1}]\mathbf{D}_1\right)\boldsymbol{\mathcal{C}}_{\boldsymbol{\alpha}}(\mathbf{h}(\mathbf{t}))\left(\mathrm{Diag}[e^{\mathbf{t}_2}]\mathbf{D}_2\right)^\top, \tag{4.38}$$

where $\mathbf{h}(\mathbf{t}) = \mathbf{D}^\top(e^{\mathbf{t}}-\mathbf{1})$ and the matrix $\mathbf{D}$ is obtained by vertically stacking $\mathbf{D}_1$ and $\mathbf{D}_2$ by analogy with (4.4). For non-Gaussian CCA, the diagonal form is

$$\mathbf{S}_{12}^{NCCA}(\mathbf{t}) = \mathbf{D}_1\,\boldsymbol{\mathcal{C}}_{\boldsymbol{\alpha}}\left(\mathbf{h}(\mathbf{t})\right)\,\mathbf{D}_2^\top, \tag{4.39}$$

where $\mathbf{h}(\mathbf{t}) = \mathbf{D}_1^\top \mathbf{t}_1 + \mathbf{D}_2^\top \mathbf{t}_2$. Finally, for mixed CCA,

$$\mathbf{S}_{12}^{MCCA}(\mathbf{t}) = \mathbf{D}_1 \, \boldsymbol{\mathcal{C}}_{\boldsymbol{\alpha}} \left(\mathbf{h}(\mathbf{t})\right) \left(\mathrm{Diag}[e^{\mathbf{t}_2}]\mathbf{D}_2\right)^\top, \tag{4.40}$$

where $\mathbf{h}(\mathbf{t}) = \mathbf{D}_1^\top \mathbf{t}_1 + \mathbf{D}_2^\top (e^{\mathbf{t}_2} - \mathbf{1})$. Since the generalized covariance matrix of the sources $\boldsymbol{\mathcal{C}}_{\boldsymbol{\alpha}}(\cdot)$ is diagonal, expressions (4.38)–(4.40) have the desired diagonal form.

4.5 Estimation in Non-Gaussian, Discrete, and Mixed CCA

The standard algorithms such as TPM or orthogonal joint diagonalization cannot be used for the estimation of $\mathbf{D}_1$ and $\mathbf{D}_2$. Indeed, even after whitening, the matrices appearing in the diagonal form (4.25) and (4.27) or (4.38)–(4.40) are *not* orthogonal. This can be explained by the fact that in the CCA case the population cumulant-tensors take the form of non-symmetric rather than symmetric, as in the ICA or discrete ICA case, CP decomposition. As an alternative, we use Jacobi-like non-orthogonal diagonalization (by similarity) algorithms [Fu and Gao, 2006, Iferroudjene et al., 2009, Luciani and Albera, 2010]. These algorithms are discussed in Section 2.3.2 and we briefly outline the main ideas here.

The estimation of the factor loading matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ of the CCA models (4.6), (4.7), and (4.11) via non-orthogonal joint diagonalization algorithms consists of the following steps : (a) construction of a set of matrices, called *target matrices*, to be jointly diagonalized (using finite sample estimators), (b) the prewhitening step, (c) a non-orthogonal joint diagonalization step, and (d) the final estimation of the factor loading matrices.

Target Matrices. There are two ways to construct target matrices : either with the CCA $\mathbf{S}$-matrices (4.24) and $\boldsymbol{\mathcal{T}}$-cumulants (4.26) or the generalized covariance matrices (4.37) (D/N/MCCA). These matrices are estimated with finite sample estimators (see Section 4.4).

When dealing with the $\mathbf{S}$- and $\boldsymbol{\mathcal{T}}$-cumulants, the target matrices are obtained via tensor projections. We define a projection $\boldsymbol{\mathcal{T}}(\mathbf{v}) \in \mathbb{R}^{M_1 \times M_2}$ of a third-order tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{M_1 \times M_2 \times M_3}$ onto a vector $\mathbf{v} \in \mathbb{R}^{M_3}$ as

$$[\boldsymbol{\mathcal{T}}(\mathbf{v})]_{m_1 m_2} := \sum_{m_3=1}^{M_3} [\mathcal{T}]_{m_1 m_2 m_3} v_{m_3}. \tag{4.41}$$

Note that the projection $\boldsymbol{\mathcal{T}}(\mathbf{v})$ is a matrix. Therefore, given $2P$ vectors

$$\left\{\mathbf{v}_{11}, \mathbf{v}_{21}, \mathbf{v}_{12}, \mathbf{v}_{22}, \ldots, \mathbf{v}_{1P}, \mathbf{v}_{2P}\right\},$$

one can construct $2P + 1$ matrices

$$\{\mathbf{S}_{12}, \quad \boldsymbol{\mathcal{T}}_{121}(\mathbf{v}_{1p}), \quad \boldsymbol{\mathcal{T}}_{122}(\mathbf{v}_{2p}), \quad \text{for } p = 1, \ldots, P\}, \tag{4.42}$$

which have the diagonal form (4.25) and (4.27). Importantly, the tensors are never constructed (see Anandkumar et al. [2012a, 2014], Podosinnikova et al. [2015] and Appendix C.2.1). The (computationally efficient) construction of target matrices from $\mathbf{S}$- and $\boldsymbol{\mathcal{T}}$-cumulants is discussed by in Appendix C.2.1.

Alternatively, the target matrices can be constructed by estimating the generalized $\mathbf{S}$-covariance matrices at $P+1$ processing points $\mathbf{0}, \mathbf{t}_1, \ldots, \mathbf{t}_P \in \mathbb{R}^{M_1+M_2}$:

$$\{\mathbf{S}_{12} = \mathbf{S}_{12}(\mathbf{0}), \;\; \mathbf{S}_{12}(\mathbf{t}_1), \;\; \ldots, \;\; \mathbf{S}_{12}(\mathbf{t}_P)\}, \tag{4.43}$$

which also have the diagonal form (4.38)–(4.40). It is interesting to mention the connection between the $\boldsymbol{\mathcal{T}}$-cumulants and the generalized $\mathbf{S}$-covariance matrices. The $\boldsymbol{\mathcal{T}}$-cumulant can be approximated via the directional derivative of the generalized covariance matrix. However, in general, e.g., $\mathbf{S}_{12}(\mathbf{t})$ with $\mathbf{t} = [\mathbf{t}_1; \mathbf{0}]$ is not exactly the same as $\boldsymbol{\mathcal{T}}_{121}(\mathbf{t}_1)$ and the former can be non-zero even when the latter is zero. This is important since order-4 and higher statistics are used with the method of moments when there is a risk that an order-3 statistic is zero like for symmetric sources. In general, the use of higher-order statistics increases the sample complexity and makes the resulting expressions quite complicated. Therefore, replacing the $\boldsymbol{\mathcal{T}}$-cumulants with the generalized $\mathbf{S}$-covariance matrices is potentially beneficial.

Prewhitening. The matrices $\mathbf{W}_1 \in \mathbb{R}^{K \times M_1}$ and $\mathbf{W}_2 \in \mathbb{R}^{K \times M_2}$ are called *whitening matrices* of $\mathbf{S}_{12}$ if

$$\mathbf{W}_1 \mathbf{S}_{12} \mathbf{W}_2^\top = \mathbf{I}_K, \tag{4.44}$$

where $\mathbf{I}_K$ is the K-dimensional identity matrix. $\mathbf{W}_1$ and $\mathbf{W}_2$ are only defined up to multiplication by any invertible matrix $\mathbf{W} \in \mathbb{R}^{K \times K}$, since any pair of matrices $\widehat{\mathbf{W}}_1 = \mathbf{Q}\mathbf{W}_1$ and $\widehat{\mathbf{W}}_2 = \mathbf{Q}^{-\top}\mathbf{W}_2$ also satisfy (4.44). In fact, using higher-order information (i.e. the $\boldsymbol{\mathcal{T}}$-cumulants or the generalized covariances for $\mathbf{t} \neq \mathbf{0}$) allows to solve this ambiguity.

The whitening matrices can be computed via SVD of $\mathbf{S}_{12}$ as in the DICA case (see Section 3.4). When M_1 and M_2 are too large, one can use a randomized SVD algorithm [see, e.g., Halko et al., 2011] to avoid the construction of the large matrix $\mathbf{S}_{12}$ and to decrease the computational time.

Applying Whitening Transform to DCCA $\boldsymbol{\mathcal{T}}$-Cumulants. Transformation of the $\boldsymbol{\mathcal{T}}$-cumulants (4.42) with whitening matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ gives new tensors $\widehat{\boldsymbol{\mathcal{T}}}_{12j} \in \mathbb{R}^{K \times K \times K}$:

$$\widehat{\boldsymbol{\mathcal{T}}}_{12j} := \boldsymbol{\mathcal{T}}_{12j} \times_1 \mathbf{W}_1^\top \times_2 \mathbf{W}_2^\top \times_3 \mathbf{W}_j^\top, \tag{4.45}$$

where $j = 1, 2$. Combining this transformation with the projection (4.41), one obtains $2P+1$ matrices

$$\mathbf{W}_1 \mathbf{S}_{12} \mathbf{W}_2^\top, \quad \mathbf{W}_1 \boldsymbol{\mathcal{T}}_{12j}(\mathbf{W}_j^\top \mathbf{u}_{jp}) \mathbf{W}_2^\top, \tag{4.46}$$

where $p = 1, \ldots, P$ and $j = 1, 2$ and we used $\mathbf{v}_{jp} = \mathbf{W}_j^\top \mathbf{u}_{jp}$ to take into account whitening along the third direction. By choosing $\mathbf{u}_{jp} \in \mathbb{R}^K$ to be the canonical vectors of the R^K, the number of tensor projections is reduced from $M = M_1 + M_2$ to $2K$.

The Choice of Projection Vectors or Processing Points. For the DCCA $\mathbf{S}$ and $\boldsymbol{\mathcal{T}}$-cumulants (4.42), we choose the K projection vectors as $\mathbf{v}_{1p} = \mathbf{W}_1^\top \mathbf{e}_p$ and $\mathbf{v}_{2p} = \mathbf{W}_2^\top \mathbf{e}_p$, where $\mathbf{e}_p$ is one of the columns of the K-identity matrix (i.e., a canonical vector). For the generalized $\mathbf{S}$-covariances (4.43), we choose the processing points as $\mathbf{t}_{1p} = \delta_1 \mathbf{v}_{1p}$ and $\mathbf{t}_{2p} = \delta_2 \mathbf{v}_{2p}$, where δ_j, for $j = 1, 2$ are set to a small value such as 0.1 divided by $\sum_m \mathbb{E}(|x_{jm}|)/M_j$, for $j = 1, 2$.

When projecting a tensor $\boldsymbol{\mathcal{T}}_{12j}$ onto a vector, part of the information contained in this tensor gets lost. To preserve all information, one could project a tensor $\boldsymbol{\mathcal{T}}_{12j}$ onto the canonical basis of $\mathbb{R}^{M_j}$ to obtain M_j matrices. However, this would be an expensive operation in terms of both memory and computational time. In practice, we use the fact, that the tensor $\boldsymbol{\mathcal{T}}_{12j}$, for $J = 1, 2$, is transformed with whitening matrices (4.45). Hence, the projection vector has to include multiplication by the whitening matrices. Since they reduce the dimension to K, choosing the canonical basis in $\mathbb{R}^K$ becomes sufficient. Hence, the choice $\mathbf{v}_{1p} = \mathbf{W}_1^\top \mathbf{e}_p$ and $\mathbf{v}_{2p} = \mathbf{W}_2^\top \mathbf{e}_p$, where $\mathbf{e}_p$ is one of the columns of the K-identity matrix. Importantly, in practice, the tensors are never constructed (see Appendix C.2.1).

The choice of the processing points of the generalized covariance matrices has to be done carefully. Indeed, if the values of $\mathbf{t}_1$ or $\mathbf{t}_2$ are too large, the exponentials blow up. Hence, it is reasonable to maintain the values of the processing points very small. Therefore, for $j = 1, 2$, we set $\mathbf{t}_{jp} = \delta_j \mathbf{v}_{jp}$ where δ_j is proportional to a parameter δ which is set to a small value ($\delta = 0.1$ by default), and the scale is determined by the inverse of the empirical average of the component of $\mathbf{x}^{(j)}$, i.e. :

$$\delta_j := \delta \frac{N M_j}{\sum_{n=1}^{N} \sum_{m=1}^{M_j} |X_{mn}^{(j)}|}, \tag{4.47}$$

for $j = 1, 2$. See Section 4.6 for an experimental comparison of different values of δ (the default value used in other experiments is $\delta = 0.1$).

Non-Orthogonal Joint Diagonalization (NOJD). Let us consider joint diagonalization of the generalized covariance matrices (4.43) (the same procedure holds for the $\mathbf{S}$- and $\boldsymbol{\mathcal{T}}$-cumulants (4.42)). Given the whitening matrices $\mathbf{W}_1$ and $\mathbf{W}_2$, the transformation of the generalized covariance matrices (4.43) gives $P+1$ matrices

$$\{\mathbf{W}_1 \mathbf{S}_{12} \mathbf{W}_2^\top, \quad \mathbf{W}_1 \mathbf{S}_{12}(\mathbf{t}_p) \mathbf{W}_2^\top, \quad p \in [P]\}, \tag{4.48}$$

where each matrix is in $\mathbb{R}^{K \times K}$ and has reduced dimension since $K < M_1, M_2$. In practice, finite sample estimators are used to construct (4.43).

Due to the diagonal form (4.25) and (4.38)–(4.40), each matrix in (4.43) has the form [6] $(\mathbf{W}_1 \mathbf{D}_1) \operatorname{Diag}(\cdot) (\mathbf{W}_2 \mathbf{D}_2)^\top$. Both $\mathbf{D}_1$ and $\mathbf{D}_2$ are (full) K-rank matrices and $\mathbf{W}_1$ and $\mathbf{W}_2$ are K-rank by construction. Therefore, the square matrices $\mathbf{V}_1 = \mathbf{W}_1 \mathbf{D}_1$ and $\mathbf{V}_2 = \mathbf{W}_2 \mathbf{D}_2$ are invertible. From (4.25) and prewhitening, we get $\mathbf{V}_1 \operatorname{cov}(\boldsymbol{\alpha}, \boldsymbol{\alpha}) \mathbf{V}_2^\top = \mathbf{I}$

6. Note that when the diagonal form has terms $\operatorname{Diag}[e^{\mathbf{t}}]$, we simply multiply the expression by $\operatorname{Diag}[e^{-\mathbf{t}}]$.

and hence $\mathbf{V}_2 = \text{Diag}[\text{var}(\boldsymbol{\alpha})^{-1}]\mathbf{V}_1^{-1}$ (the covariance matrix of the sources is diagonal and we assume they are non-deterministic, i.e. $\text{var}(\boldsymbol{\alpha}) \neq \mathbf{0}$). Substituting this into $\mathbf{W}_1\mathbf{S}_{12}(\mathbf{t})\mathbf{W}_2^\top$ and using the diagonal form (4.38)–(4.40), we obtain that the matrices in (4.43) have the form $\mathbf{V}_1\text{Diag}(\cdot)\mathbf{V}_1^{-1}$. Hence, we deal with the problem of the following type : Given P non-defective (a.k.a. diagonalizable) matrices $\mathcal{B} = \{\mathbf{B}_1, \ldots, \mathbf{B}_P\}$, where each matrix $\mathbf{B}_p \in \mathbb{R}^{K\times K}$, find an invertible matrix $\mathbf{Q} \in \mathbb{R}^{K\times K}$ such that

$$\mathbf{Q}\mathcal{B}\mathbf{Q}^{-1} = \{\mathbf{Q}\mathbf{B}_1\mathbf{Q}^{-1}, \quad \ldots, \quad \mathbf{Q}\mathbf{B}_P\mathbf{Q}^{-1}\} \tag{4.49}$$

are (jointly) as diagonal as possible. This can be seen as a joint non-symmetric eigenvalue problem. This problem should not be confused with the classical joint diagonalization problem by congruence (JDC), where $\mathbf{Q}^{-1}$ is replaced by $\mathbf{Q}^\top$, except when $\mathbf{Q}$ is an orthogonal matrix [Luciani and Albera, 2010]. JDC is often used for ICA algorithms or moment matching based algorithms for graphical models when a whitening step is not desirable (see, e.g., Kuleshov et al. [2015a] and references therein). However, neither JDC nor the orthogonal diagonalization-type algorithms [such as, e.g., the tensor power method by Anandkumar et al., 2014] are applicable for the problem (4.49).

To solve the problem (4.49), we use the Jacobi-like non-orthogonal joint diagonalization (NOJD) by similarity algorithms [e.g., Fu and Gao, 2006, Iferroudjene et al., 2009, Luciani and Albera, 2010]. These algorithms are an extension of the orthogonal joint diagonalization algorithms based on Jacobi (=Givens) rotations [Golub and Van Loan, 1996, Bunse-Gerstner et al., 1993, Cardoso and Souloumiac, 1996]. Although these algorithms are quite stable in practice, we are not aware of any theoretical guarantees about their convergence or stability to perturbation. [7]

ED-Based Algorithm. By analogy with the orthogonal case [Section 3.4 ; Cardoso, 1989, Anandkumar et al., 2012a], we can easily extend the idea of the ED-based algorithm to the non-orthogonal one. Indeed, it amounts to performing whitening as before and constructing only one matrix with the diagonal structure, e.g., $\mathbf{B} = \mathbf{W}_1\mathbf{S}_{12}(\mathbf{t})\mathbf{W}_2^\top$ for some $\mathbf{t}$. Then, the matrix $\mathbf{Q}$ is obtained as the matrix of the eigenvectors of $\mathbf{B}$. The vector $\mathbf{t}$ can be, e.g., chosen as $\mathbf{t} = \mathbf{W}\mathbf{u}$, where $\mathbf{W} = [\mathbf{W}_1; \mathbf{W}_2]$ and $\mathbf{u} \in \mathbb{R}^K$ is a vector sampled uniformly at random.

This ED-based algorithm and the NOJD algorithms are closely connected. In particular, when $\mathbf{B}$ has real eigenvectors, the ED-based algorithm is equivalent to NOJD of $\mathbf{B}$. Indeed, in such case, NOJD boils down to an algorithm for a non-symmetric eigenproblem [Eberlein, 1962, Ruhe, 1968]. In practice, however, due to the presence of noise and finite sample errors, $\mathbf{B}$ may have complex eigenvectors. In such case, the ED-based algorithm is different from NOJD. Importantly, the joint diagonalization

7. Note that recently novel perturbation analysis results were obtained for the simultaneous Schur decomposition, which is an algorithms that can be used for the computation of the non-orthogonal joint diagonalization problem by similarity [Colombo and Vlassis, 2016a,b,c]. An experimental comparison and potential extension of these results to the Jacobi-like algorithms could be of interest.

type algorithms are known to be more stable in practice [see, e.g., Bach and Jordan, 2002, Podosinnikova et al., 2015].

While deriving precise theoretical guarantees is beyond the scope of this thesis, the techniques outlined by Anandkumar et al. [2012a] for the ED-based (spectral) algorithm for latent Dirichlet Allocation can potentially be extended. The main difference is obtaining the analogue of the SVD accuracy [Lemma C.3, Anandkumar et al., 2013a] for the eigendecomposition. This kind of analysis can potentially be extended with the techniques outlined by Stewart and Sun [1990, Chapter 4]. Nevertheless, with appropriate parametric assumptions on the sources, we expect that the above described extension of the ED-based algorithm should lead to similar guarantee as the ED-based (spectral) algorithm of Anandkumar et al. [2012a].

4.6 Experiments

In this section, we illustrate the proposed algorithms on both synthetic and real data. In Section 4.6.1, we illustrate the estimation in the DCCA model on synthetic count data. In Section 4.6.2, we illustrate the estimation in the non-Gaussian CCA model on synthetic continuous data. In Section 4.6.3, we illustrate the estimation in the DCCA model on real data. The code for reproducing the experiments described in this section is available at https://github.com/anastasia-podosinnikova/cca.

4.6.1 Synthetic Count Data

Synthetic Data. We first consider multi-view models for count data, which we also sometimes refer to as discrete data, and sample synthetic data to have ground truth information (i.e., matrices $\mathbf{D}_1$ and $\mathbf{D}_2$) for evaluation. We sample data from the linear DCCA model, which is defined as follows

$$\begin{aligned} \boldsymbol{\alpha} &\sim \text{Gamma}(\mathbf{c}, \mathbf{b}), \\ \boldsymbol{\beta}^{(j)} &\sim \text{Gamma}(\mathbf{c}_j, \mathbf{b}_j), \quad j = 1, 2, \\ \mathbf{x}^{(j)} &\sim \text{Poisson}(\mathbf{D}_j \boldsymbol{\alpha} + \mathbf{F}_j \boldsymbol{\beta}^{(j)}), \quad j = 1, 2, \end{aligned} \tag{4.50}$$

where the vector with independent components $\boldsymbol{\alpha}$ corresponds to the *common sources* and vectors with independent components $\boldsymbol{\beta}^{(j)}$, for $j = 1, 2$, are the view-specific *(noise) sources*. Let us define $\mathbf{s}^{(j)} \sim \text{Poisson}(\mathbf{D}_j \boldsymbol{\alpha})$ to be the part of the sample due to the common sources and $\mathbf{n}^{(j)} \sim \text{Poisson}(\mathbf{F}_j \boldsymbol{\beta}^{(j)})$ to be the part of the sample due to the noise (i.e., $\mathbf{x}^{(j)} = \mathbf{s}^{(j)} + \mathbf{n}^{(j)}$). Below we explain the choice of the parameters.

We define the expected sample length due to the common sources and noise sources, respectively :

$$L_s^{(j)} := \mathbb{E}\left[\sum_{m=1}^{M_j} s_m^{(j)}\right], \qquad L_n^{(j)} := \mathbb{E}\left[\sum_{m=1}^{M_j} n_m^{(j)}\right], \qquad j = 1, 2.$$

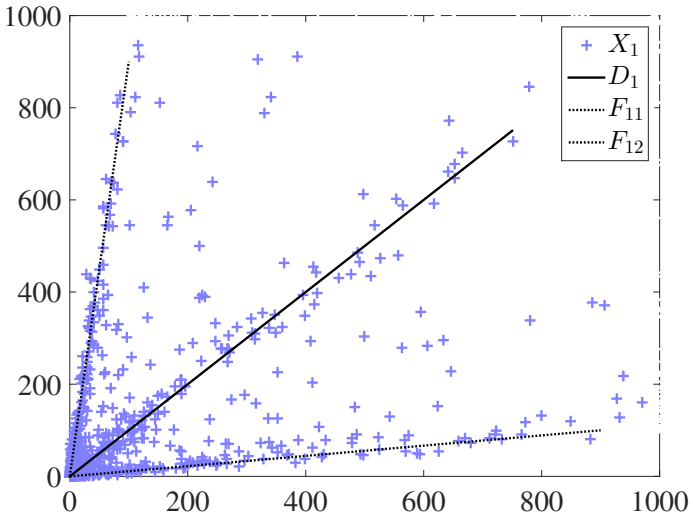


FIGURE 4-2 – An example of a **2D Discrete Synthetic Data** sample drawn from the linear DCCA model (4.50) as explained in Section 4.6.1. The parameters are set to the following values : the dimensions are $M_1 = M_2 = K_1 = K_2 = 2$ and $K = 1$; the scale parameters are $c = c_1 = c_2 = 0.1$; the rate parameters are set to ensure $L_s = L_n = 100$; the matrices are $\mathbf{D}_1 = \mathbf{D}_2$ with $[\mathbf{D}_1]_{11} = [\mathbf{D}_1]_{12} = 0.5$ and $\mathbf{F}_1 = \mathbf{F}_2$ with $[\mathbf{F}_1]_{11} = [\mathbf{F}_1]_{22} = 0.9$ and $[\mathbf{F}_1]_{12} = [\mathbf{F}_1]_{21} = 0.1$.

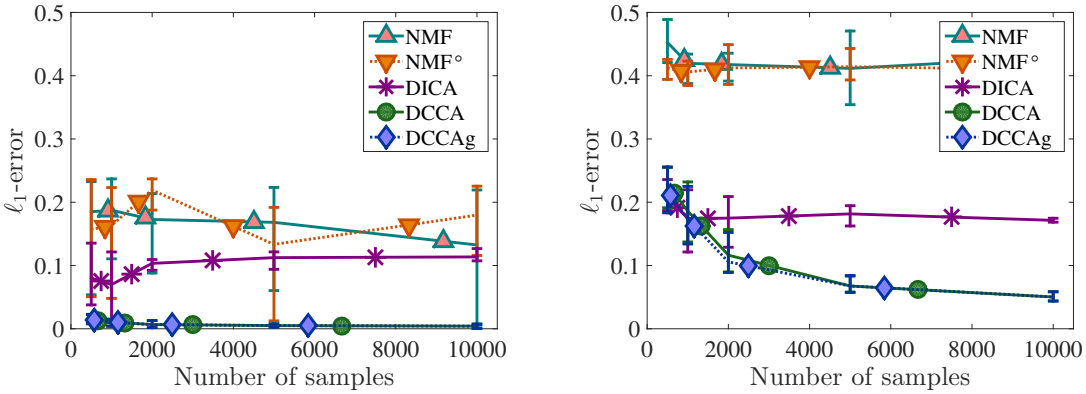
For sampling, we fix the target values $L_s := L_s^{(j)} = L_s^{(2)}$ and $L_n := L_n^{(1)} = L_n^{(2)}$. We assume that the prior parameters are uniform, i.e. $\mathbf{c} = c\mathbf{1}$, $\mathbf{c}_j = c_j\mathbf{1}$, $\mathbf{b} = b\mathbf{1}$, and $\mathbf{b}_j = b_j\mathbf{1}$, for $j = 1, 2$, where $\mathbf{1}$ denotes a vector with all elements equal to 1 of a respective (not always equal) dimension. We set the parameters b and b_j to ensure the chosen values of the document lengths L_s and L_n, i.e. $b = Kc/L_s$ and $b_j = K_j c_j / L_n$ (see below the values of c and c_j). We sample :

(a) *2D Data*, where the dimensions are $M_1 = M_2 = K_1 = K_2 = 2$ and $K = 1$, the matrices are $\mathbf{D}_1 = \mathbf{D}_2$ with $[\mathbf{D}_1]_{11} = [\mathbf{D}_1]_{12} = 0.5$ and $\mathbf{F}_1 = \mathbf{F}_2$ with $[\mathbf{F}_1]_{11} = [\mathbf{F}_1]_{22} = 0.9$ and $[\mathbf{F}_1]_{12} = [\mathbf{F}_1]_{21} = 0.1$, the scale parameters of the sources are $c = c_1 = c_2 = 0.1$, and the rate parameters of the sources are set to ensure the document lengths $L_s = L_n = 100$ (see Figure 4-2).

(b) *20D Data*, where the dimensions are $M_1 = M_2 = K_1 = K_2 = 20$ and $K = 10$, each column of the matrices $\mathbf{D}_j$ and $\mathbf{F}_j$, for $j = 1, 2$, is sampled from the symmetric Dirichlet distribution with the concentration parameter equal to 0.5, the scale parameters of the sources are $c = 0.3$, and $c_1 = c_2 = 0.1$, and the rate parameters of the sources are set to ensure the document lengths $L_s = L_n = 1,000$.

For each experiment, $\mathbf{D}_j$ and $\mathbf{F}_j$, for $j = 1, 2$, are sampled once and, then, the stacked observations $\mathbf{x}_n$, for $n \in [N]$ are sampled for different sample sizes $N = 500$; $1,000$; $2,000$; $5,000$; and $10,000$. For each N, 5 samples are drawn.

Metric. The evaluation is performed on a matrix $\mathbf{D}$ obtained by stacking $\mathbf{D}_1$ and $\mathbf{D}_2$



(a) *2D Data* : the dimensions $M_1 = M_2 = K_1 = K_2 = 2$, and $K = 1$, the scale parameters $c = c_1 = c_2 = 0.1$, and the document lengths $L_s = L_n = 100$.

(b) *20D Data* : the dimensions $M_1 = M_2 = K_1 = K_2 = 20$, $K = 10$, the scale parameters $c = 0.3$, $c_1 = c_2 = 0.1$, and the document lengths $L_s = L_n = 1,000$.

FIGURE 4-3 – Synthetic experiment with count data.

vertically.[8] As in Section 3.5.2, we use as evaluation metric the normalized ℓ_1-error between a recovered matrix $\widehat{\mathbf{D}}$ and the true matrix $\mathbf{D}$ with the best permutation of columns $\mathrm{err}_1(\widehat{\mathbf{D}}, \mathbf{D}) := \min_{\boldsymbol{\pi} \in \mathrm{PERM}} \frac{1}{2K} \sum_k \|\widehat{\mathbf{d}}_{\pi_k} - \mathbf{d}_k\|_1 \in [0, 1]$. The minimization is over the possible permutations $\boldsymbol{\pi} \in \mathrm{PERM}$ of the columns of $\widehat{\mathbf{D}}$ and can be efficiently obtained with the Hungarian algorithm for bipartite matching [Kuhn, 1955]. The (normalized) ℓ_1-error takes the values in [0, 1] and smaller values of this error indicate better performance of an algorithm.

Algorithms. We compare DCCA (implementation with the $\mathbf{S}$- and $\boldsymbol{\mathcal{T}}$-cumulants) and DCCAg (implementation with the generalized $\mathbf{S}$-covariance matrices and the processing points initialized as described in Section 4.5) to DICA and the non-negative matrix factorization (NMF) algorithm with multiplicative updates for divergence [Lee and Seung, 2001]. To run DICA or NMF, we use the stacking trick (4.4). DCCA is set to estimate K components. DICA is set to estimate either $K_0 = K + K_1 + K_2$ or $M = M_1 + M_2$ components (whichever is the smallest, since DICA cannot work in the overcomplete case). NMF is always set to estimate K_0 components. For the evaluation of DICA/NMF, the K columns with the smallest ℓ_1-error are chosen. NMF° stands for NMF initialized with a matrix $\mathbf{D}$ of the form (4.4) with induced zeros ; otherwise NMF is initialized with (uniformly) random non-negative matrices. The running times are discussed in Section 4.6.3.

Synthetic Experiment. We first perform an experiment with discrete synthetic 2D Data (see Figure 4-3a) and then repeat the same experiment when the size of the problem is 10 times larger (see Figure 4-3b). In practice, we observed that for $K < M$ all models work approximately equally well, except for NMF which breaks down in high dimensions. In the overcomplete case as in Figure 4-3, DCCA works

8. Note that the column order of matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ is preserved (see the comment after Theorem 4.3.1).

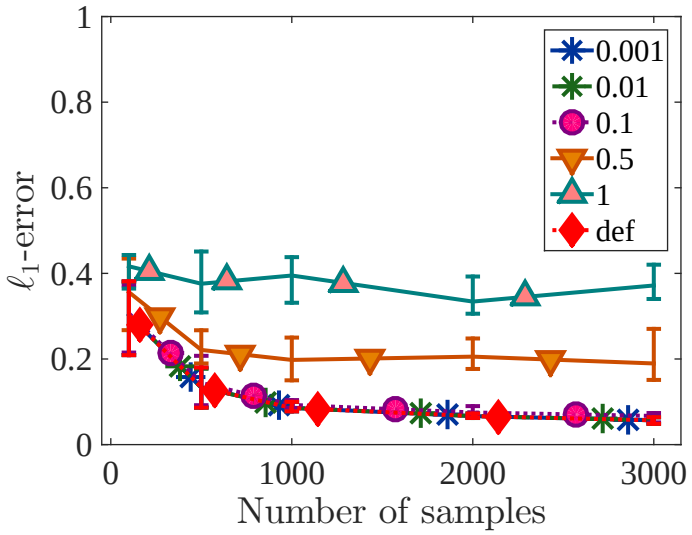


FIGURE 4-4 – An experimental analysis of the performance of DCCAg with generalized covariance matrices using different parameters δ_j for the processing points. The numbers in the legend correspond to the values of δ defining δ_j via (4.47). The default value (def) is $\delta = 0.1$. The experiment is performed on 20D Data (see description in the text).

better.

Sensitivity of the Generalized Covariance Matrices to the Choice of the Processing Points. We experimentally analyze the performance of the DCCAg algorithm based on the generalized **S**-covariance matrices vs. the parameters δ_1 and δ_2. We use the experimental setup of the synthetic count data from this section with $K_1 = K_2 = K = 10$, i.e. 20D Data. The results are presented in Figure 4-4.

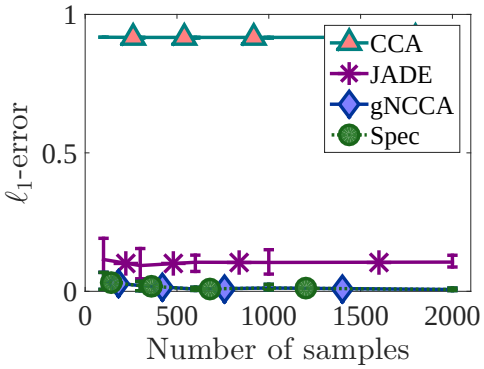
4.6.2 Synthetic Continuous Data

This experiment is essentially a continuous analogue to the synthetic experiment with the discrete data from Section 4.6.1.

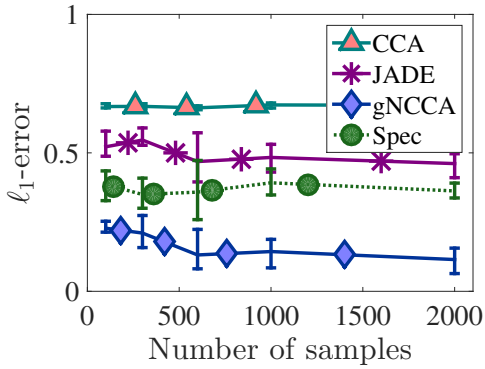
Synthetic Data. We sample synthetic data from the linear non-Gaussian CCA (NCCA) model :

$$\begin{aligned} \boldsymbol{\alpha} &\sim z_{\boldsymbol{\alpha}}\text{Gamma}(\mathbf{c}, \mathbf{b}), \\ \boldsymbol{\beta}_j &\sim z_{\boldsymbol{\beta}_j}\text{Gamma}(\mathbf{c}_j, \mathbf{b}_j), \qquad j = 1, 2, \\ \mathbf{x}^{(j)} &= \mathbf{D}_j\boldsymbol{\alpha} + \mathbf{F}_j\boldsymbol{\beta}^{(j)}, \qquad j = 1, 2, \end{aligned} \tag{4.51}$$

where $z_{\boldsymbol{\alpha}}$ and $z_{\boldsymbol{\beta}_j}$, for $j = 1, 2$, are Rademacher random variables (i.e., they take the values -1 or 1 with the equal probabilities). As in the count data case (see Section 4.6.1), $\boldsymbol{\alpha}$ stands for the *common sources*, and $\boldsymbol{\beta}_j$ stands for the view-specific *(noise) sources*. Note that both vector sources are non-Gaussian. The rate parameters of the gamma distribution are initialized by analogy with the discrete case. The



(a) The number of factors : $K_1 = K_2 = K = 1$.



(b) The number of factors : $K_1 = K_2 = K = 10$.

FIGURE 4-5 – Synthetic experiment with continuous data. For both experiments, the parameters are $M_1 = M_2 = 20$, $c = c_1 = c_2 = 0.1$ and $L_n = L_s = 1000$. The data are synthetic continuous (see the description in Section 4.6.2).

elements of the matrices $\mathbf{D}_j$ and $\mathbf{F}_j$, for $j = 1, 2$, are sampled independently from the uniform distribution in $[-1, 1]$. Each column of $\mathbf{D}_j$ and $\mathbf{F}_j$, for $j = 1, 2$, is normalized to have unit ℓ_1-norm.

Algorithms. We compare gNCCA (the implementation of NCCA with the generalized **S**-covariance matrices with the default values of the parameters δ_1 and δ_2 as described in Section 4.5), the ED-based algorithm for NCCA (also with the generalized **S**-covariance matrices), the JADE algorithm [9] [Cardoso and Souloumiac, 1993] for independent component analysis (ICA), and classical CCA.

Synthetic Experiment. In Figure 4-5, the results of the experiment for the different number of topics are presented. The error of the classical CCA is high due to the mentioned unidentifiability issues.

4.6.3 Real Data Experiment – Translation Topics

Real Data (Translation). Following Vinokourov et al. [2002], we illustrate the performance of DCCA by extracting bilingual topics from the *Hansard collection* [Vinokourov and Girolami, 2002] with aligned English and French proceedings of the 36-th Canadian Parliament. In Tables 4.1–4.5 on pages 115–117, we present the topics extracted after running DCCA with $K = 20$.

For the real data experiment, we estimate the factor loading matrices (topics, in the following) $\mathbf{D}_1$ and $\mathbf{D}_2$ of aligned proceedings of the 36-th Canadian Parliament in English and French languages. [10]

Although going into details of natural language processing (NLP) related problems

9. The code is available at http://perso.telecom-paristech.fr/ cardoso/Algo/Jade/jadeR.m.
10. The data are available at http://www.isi.edu/natural-language/download/hansard/.

is not the goal of this thesis, we do minor preprocessing (see below) of this text data to improve the presentation of the estimated bilingual topics $\mathbf{D}_1$ and $\mathbf{D}_2$.

The 20 topics obtained with DCCA are presented in Tables 4.1–4.5 pages 115–117. For each topic, we display the 20 most frequent words (ordered from top to bottom in the decreasing order). Most of the topics have quite clear interpretation. Moreover, we can often observe the pairs of words which are each others translations in the topics. For example,

- in the topic 10 : the phrase "pension plan" can be translated as "régime de retraite," the word "benefits" as "prestations," and abbreviations "CPP" and "RPC" stand for "Canada Pension Plan" and "Régime de pensions du Canada," respectively ;
- in the topic 3 : "OTAN" is the French abbreviation for "NATO," the word "war" is translated as "guerre," and the word "peace" as "paix ;"
- in the topic 9 : "Nisga" is the name of an Indigenous (or "aboriginal") people in British Columbia, the word "aboriginal" translates to French as "autochtontes," and, e.g., the word "right" can be translated as "droit."

Note also that, e.g., in the topic 10, although the French words "ans" and "années" are present in the French topic, their English translation "year" is not, since it was removed as one of the 15 most frequent words in English (see below).

Data Preprocessing

For the experiment, we use *House Debate Training Set* of the *Hansard collection*. To preprocess this text data, we perform case conversion, stemming, and removal of some stop words. For stemming, we use the *SnowballStemmer* of the *NLTK toolbox* [Bird et al., 2009] for both English and French languages. Although this stemmer has some problems (such as mapping several different forms of a word to a single stem in one language but not in the other), they are left beyond our consideration. Moreover, in addition to the standard stop words of the *NLTK toolbox*, we also removed the following words that we consider to be stop words for our task [11] (and their possible forms) :

- from English : ask, become, believe, can, could, come, cost, cut, do, done, follow, get, give, go, know, let, like, listen, live, look, lost, make, may, met, move, must, need, put, say, see, show, take, think, talk, use, want, will, also, another, back, day, certain, certainly, even, final, finally, first, future, general, good, high, just, last, long, major, many, new, next, now, one, point, since, thing, time, today, way, well, without ;
- from French (translations in brackets) : demander (ask), doit (must), devenir (become), dit (speak, talk), devoir (have to), donner (give), ila (he has), met (put), parler (speak, talk), penser (think), pourrait (could), pouvoir (can),

11. This list of words was obtained by looking at words that appear in the top-20 words of a large number of topics in a first experiment. Removing these words did not change much the content of the topics, but made them more interpretable.

prendre (take), savoir (know), aller (go), voir (see), vouloir (want), actuellement, après (after), aujourd'hui (today), autres (other), bien (good), beaucoup (a lot), besoin (need), cas (case), cause, cela (it), certain, chose (thing), déjà (already), dernier (last), égal (equal), entre (between), façon (way), grand (big), jour (day), lorsque (when), neuf (new), passé (past), plus, point, présent, prêts (ready), prochain (next), quelque (some), suivant (next), unique.

After stemming and removing stop words, several files had different number of documents in each language and had to be removed too. The numbers of these files are : 16, 36, 49 55, 88, 103, 110, 114, 123, 155, 159, 204, 229, 240, 2-17, 2-35.

We also removed the 15 most frequent words from each language. These include :

- in English : Mr, govern, member, speaker, minist(er), Hon, Canadian, Canada, bill, hous(e), peopl(e), year, act, motion, question ;
- in French : gouvern(er), président, loi, députe(é), ministr(e), canadien, Canada, projet, Monsieur, question, part(y), chambr(e), premi(er), motion, Hon.

Removing these words is not necessary, but improves the presentation of the learned topics significantly. Indeed, the most frequent words tend to appear in nearly every topic (often in pairs in both languages as translations of each other, e.g., "member" and "député" or "Canada" in both languages, which confirms one more time the correctness of our algorithm).

Finally, we select $M_1 = M_2 = 5,000$ words for each language to form matrices $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ each containing $N = 11,969$ documents in columns. As stemming removes the words endings, we map the stemmed words to the respective most frequent original words when showing off the topics in Tables 4.1-4.5.

Running Time

For the real data experiment, the runtime of DCCA algorithm is 24 seconds including 22 seconds for SVD at the whitening step. In general, the computational complexity of the D/N/MCCA algorithms is bounded by the time of SVD plus $O(RNK) + O(NK^2)$, where R is the largest number of non-zero components in the stacked vector $\mathbf{x} = [\mathbf{x}^{(1)}; \mathbf{x}^{(2)}]$, plus the time of NOJD for P target matrices of the K-by-K size. In practice, DCCAg is faster than DCCA.

4.7 Conclusion

We have proposed the first identifiable versions of CCA, together with moment matching algorithms which allow the identification of the loading matrices in a semi-parametric framework, where no assumptions are made regarding the distribution of the source or the noise. We also introduced new sets of moments (our generalized covariance matrices), which could prove useful in other settings.

farmers	agriculteurs	division	no	nato	otan	tax	impôts
agriculture	programme	negatived	vote	kosovo	kosovo	budget	budget
program	agricole	paired	rejetée	forces	militaires	billion	enfants
farm	pays	declare	voix	military	guerre	families	économie
country	important	yeas	mise	war	international	income	années
support	problème	divided	pairs	troops	pays	country	dollars
industry	aide	nays	porte	country	réfugiés	debt	pays
trade	agriculture	vote	contre	world	situation	students	finances
provinces	années	order	déclaration	national	paix	children	familles
work	secteur	deputy	suppléant	peace	yougoslavie	money	fiscal
problem	provinces	thibeault	vice	international	milosevic	finance	milliards
issue	gens	mcclelland	lethbridge	conflict	forces	education	libéraux
us	économie	ms	poisson	milosevic	serbes	liberal	jeunes
tax	industrie	oversee	mme	debate	intervention	fund	gens
world	dollars	rise	plantes	support	troupes	care	important
help	mesure	past	harvey	action	humanitaire	poverty	revenu
federal	faut	army	perdront	refugees	nations	jobs	mesure
producers	situation	peterson	sciences	ground	conflit	benefits	argent
national	réformiste	heed	liberté	happen	ethnique	child	santé
business	accord	moral	prière	issue	monde	pay	payer

TABLE 4.1 – The real data (translation) experiment. Topics 1 to 4.

work	travail	justice	jeunes	business	entreprises	board	commission
workers	négociations	young	justice	small	petites	wheat	blé
strike	travailleurs	crime	victimes	loans	programme	farmers	agriculteurs
legislation	grève	offenders	systéme	program	banques	grain	administration
union	emploi	victims	crime	bank	finances	producers	producteurs
agreement	droit	system	mesure	money	important	amendment	grain
labour	syndicat	legislation	criminel	finance	économie	market	conseil
right	services	sentence	contrevenants	access	secteur	directors	ouest
services	accord	youth	peine	jobs	argent	western	amendement
negotiations	voix	criminal	ans	economy	emplois	election	comité
chairman	adopter	court	juge	industry	assurance	support	réformiste
public	réglement	issue	enfants	financial	financiére	party	propos
party	article	law	important	billion	appuyer	farm	important
employees	retour	community	gens	support	créer	agriculture	compte
collective	gens	right	tribunaux	ovid	choc	clause	prix
agreed	conseil	reform	droit	merger	accés	ottawa	no
board	collectivités	country	problème	information	milliards	us	dispositions
arbitration	postes	problem	réformiste	size	propos	vote	information
grain	grain	person	traité	korea	pme	cwb	mesure
order	trésor	support	faut	companies	obtenir	states	produits

TABLE 4.2 – The real data (translation) experiment. Topics 5 to 8.

nisga	nisga	pension	régime	newfoundland	terre	health	santé
treaty	autochtones	plan	pensions	amendment	droit	research	recherche
aboriginal	traité	fund	cotisations	school	modifications	care	fédéral
agreement	accord	benefits	prestations	education	provinces	federal	provinces
right	droit	public	retraite	right	école	provinces	soins
land	nations	investment	emploi	constitution	comité	budget	budget
reserve	britannique	money	assurance	provinces	éducation	billion	dollars
national	indiennes	contribution	investissement	committee	enseignement	social	systéme
british	terre	cpp	fonds	system	systéme	money	finances
columbia	colombie	retirement	années	reform	enfants	tax	transfert
indian	réserves	pay	ans	minority	vote	system	milliards
court	non	billion	argent	denominational	amendement	provincial	domaine
party	affaires	change	important	referendum	constitution	fund	sociale
law	négociations	liberal	administration	children	religieux	country	années
native	bande	legislation	dollars	quebec	référendum	quebec	maladie
non	réformiste	board	propos	parents	article	transfer	important
constitution	constitution	employment	milliards	students	réformiste	debt	programme
development	application	tax	gens	change	québec	liberal	libéraux
reform	user	rate	taux	party	constitutionnelle	services	environnement
legislation	gestion	amendment	rpc	labrador	confessionnelles	issue	assurance

TABLE 4.3 – The real data (translation) experiment. Topics 9 to 12.

party	pays	tax	agence	quebec	québec	court	pêches
country	politique	provinces	provinces	federal	québécois	right	droit
issue	important	agency	revenu	information	fédéral	fisheries	juge
us	comité	federal	impôts	provinces	provinces	decision	cours
debate	libéraux	revenue	fiscal	protection	protection	fish	gens
liberal	réformiste	taxpayers	fédéral	right	renseignements	issue	décision
committee	gens	equalization	contribuables	legislation	droit	law	important
work	débat	system	payer	provincial	personnel	work	pays
order	accord	services	taxe	person	privé	us	traité
support	démocratique	accountability	péréquation	law	protéger	party	conservateur
reform	québécois	amendment	argent	constitution	électronique	debate	région
election	réglement	billion	services	privacy	article	justice	problème
world	propos	money	fonction	country	commerce	problem	supréme
quebec	collégue	party	modifier	electronic	provinciaux	community	tribunaux
standing	parlementaire	provincial	article	court	bloc	supreme	faut
national	appuyer	public	ministére	bloc	vie	country	situation
interest	opposition	business	administration	students	application	area	victimes
important	élections	reform	déclaration	section	citoyens	case	appuyer
right	bloc	office	tps	clear	non	order	mesure
public	industrie	support	provinciaux	states	nationale	parliament	trouve

TABLE 4.4 – The real data (translation) experiment. Topics 13 to 16.

legislation	important	national	important	vote	voix	water	eau
issue	environnement	area	gens	yeas	no	trade	ressources
amendment	mesure	parks	environnement	division	adopter	resources	accord
committee	enfants	work	parcs	nays	vote	country	environnement
support	comité	country	pays	agreed	non	agreement	important
protection	propos	us	marine	deputy	contre	provinces	industrie
information	pays	development	mesure	paired	dépenses	industry	américains
industry	appuyer	support	propos	responsible	accord	protection	pays
concerned	protection	community	fédéral	treasury	conseil	export	provinces
right	article	federal	jeunes	divided	budget	environmental	exportations
important	droit	issue	appuyer	order	crédit	us	échange
change	accord	legislation	années	fiscal	trésor	freshwater	conservateur
world	gens	help	assurance	amount	oui	federal	responsabilité
law	amendement	liberal	gestion	pleased	mise	world	effet
families	adopter	world	conservateur	budget	propos	issue	quantité
work	industrie	responsible	accord	ms	porte	legislation	traité
children	non	concerned	région	infrastructure	lib	environment	commerce
order	société	committee	problème	board	pairs	responsible	unis
national	porte	problem	nationale	consent	veuillent	development	économie
states	no	important	québec	estimates	vice	culture	alena

TABLE 4.5 – The real data (translation) experiment. Topics 17 to 20.

Chapitre 5

Conclusion and Future Work

5.1 Algorithms for the CP Decomposition

In Sections 2.2.2, 2.2.3, 3.3.2, and 4.4.1, we saw that population higher-order statistics of many latent linear models admit representation in the form of (often symmetric) CP decomposition (sometimes also non-negative) with the linear transformation matrix as the factor matrix. This clearly poses a question of an optimal algorithm for the CPD computation. Importantly, computation of the symmetric, rather than non-symmetric, CPD is a more challenging task.

The Symmetric CPD

Orthogonal Approaches. In the literature, numerous approaches were proposed to this problem. One approach—the approach chosen in this thesis—is based on the *prewhitening* step, which leads to the problem of (approximately) *orthogonal symmetric CPD*. The popularity of this approach can be explained by : (a) relatively easy theoretical analysis due to the orthogonality and availability of the global solution guarantees in the idealized population case, (b) reduced dimension of the target tensor due to the prewhitening step leading, in particular, to relatively low computational complexity. Several algorithms are readily available for the approximation of the orthogonal symmetric CPD (see Section 2.3, plus gradient-based algorithms), however, the choice of one or another algorithm is not obvious in practical applications and more extensive theoretical and experimental analysis is of interest.

We saw in Section 3.5 that the tensor power method does not always finds accurate approximations of $\mathbf{q}_k$ when applied to the estimation of latent linear models (see, e.g., Section 3.5). A possible explanation is the error introduced at the prewhitening step since a sample estimate of the matrix $\mathbf{S}^{DICA}$ is also not exactly in the diagonal form (4.21). This is a known problem of *orthogonal diagonalization methods* based on prewhitening [see, e.g., Cardoso, 1994b, Souloumiac, 2009b]. Another potential problem is the deflation procedure. It is well known, that an error obtained at a deflation step propagates to all consecutive steps. Moreover, Kolda [2001] provides

an example where the best rank-one approximation of a cubic tensor is not a factor in the best rank-two approximation, which could potentially be an issue in the finite sample case, where the target tensor is not exactly orthogonally decomposable. These and related questions still require further investigation.

Non-Orthogonal Approaches. The well known issue of the prewhitening-based approach is the propagation of the whitening error [Cardoso, 1994b] in the non-idealized case of finite sample estimates of higher order statistics. Therefore, multiple attempts to develop algorithms without prewhitening were made. In the symmetric case, these algorithms are mostly based on the idea of *non-orthogonal joint matrix diagonalization* : which include both gradient-based algorithms [Yeredor, 2002, Yeredor et al., 2004, Vollgraf and Obermayer, 2006] as well as multiplicative updates-based methods [Ziehe et al., 2004, Afsari, 2006, Souloumiac, 2009a, Mesloub et al., 2012]. However, these approaches are more difficult than in the orthogonal case [Afsari, 2007]. Although some comparison of these algorithms among each other and with some orthogonal-type methods is available in the literature [Dégerine and Kane, 2007, Souloumiac, 2009b, Chabriel et al., 2014], a more comprehensive study (especially gradient-based vs. multiplicative update-based approaches) is needed, especially, in the context when these algorithms are applied to the problem of estimation in latent linear models.

As opposed to the orthogonal case, non-orthogonal joint diagonalization does not perform preliminary dimensionality reduction. This means that all computations have to be performed with the original non-reduced tensor (of size M, where M is the number of words in the vocabulary and not the number of topics K), which can be often significantly larger especially in the undercomplete case with $K \ll M$ making non-orthogonal approaches computationally unattractive. One way to address this problem is the so called *sketching* technique [Wang et al., 2015, Keriven et al., 2016a,b], which is becoming more and more popular in the context of large-scale learning. Another approach is adapting stochastic optimization methods [e.g., similar to Vu et al., 2015, Ge et al., 2015].

Non-Negative Approaches. The estimation problem in many latent linear models actually reduces to the joint *non-negative* symmetric CPD of second- and third-order statistics, with emphasis on non-negativity. The approaches widely known in the literature, including the ones of this thesis, do not explicitly handle the non-negativity constraint. Explicit integration of this constraint would necessary lead to an improved accuracy of the estimation given the respective algorithms can be solved efficiently. Indeed, we performed multiple experiments for the different orthogonal-type diagonalization algorithms in the context of topic models (Chapter 3), where we computed the so called *Amari metric* [Amari et al., 1996], which measures the quality of the joint diagonalization type algorithms, before and after the heuristic truncation (of negative values) step which ensures the non-negativity (see Section 3.4). We saw significant increase of the error after such truncation step.

A potential approach to resolving this issue is performing the non-negative symmetric joint CP decomposition of second- and third-order statistics. Since the low rank ten-

sor approximation in the non-negative setting is well-posed [Lim and Comon, 2009], it makes sense to investigate this direction. In the non-symmetric non-negative case, numerous algorithms, often based on alternating minimization, are known [Krijnen and Ten Berge, 1991, Bro and Jong, 1997, Bro and Sidiropoulos, 1998, Welling and Weber, 2001, Hazan et al., 2005, Shashua and Hazan, 2005, Cichocki et al., 2009, Lim and Comon, 2009, Royer et al., 2011, Zhou et al., 2014, Kim et al., 2014]. Extensions and comparisons of these algorithms to the symmetric case is therefore of interest.

The Non-Symmetric CPD

Gradient-Based Methods and Alternating Least Squares. Similar questions of theoretical and experimental comparison of the CPD algorithms naturally arise in the *non-symmetric* case. An extremely widely-used approach in practice is the one based on the low-rank approximation formulation and alternating least squares type of methods [Comon et al., 2009a, Kolda and Bader, 2009]. Major drawback of all these algorithms are different *CP degeneracies* : bottlenecks, swamps, and general ill-posedness of the problem, which do cause problems in practice [Comon et al., 2009a]. Hence, understanding whether joint diagonalization-based approaches are more preferable in this case is another important question.

Simultaneous Schur Decomposition. In addition to the non-orthogonal joint diagonalization (NOJD) by similarity algorithms (see Section 2.3.2), simultaneous Schur decomposition is another important approach to the non-symmetric CPD approximation problem. New perturbation analysis results have just became available for these methods [De Lathauwer et al., 2004, Colombo and Vlassis, 2016a,b,c]. The most notable results are the so called *a posteriori* bounds on the quality of the approximation, which take into account an approximate solution (an output of the algorithm) rather than a global minimizer as in *a priori* bounds [Cardoso, 1994a]. Comparison of these algorithms to NOJD by similarity as well as potential extension of this perturbation analysis is of interest.

Joint Diagonalization Algorithms

(Orthogonal and non-orthogonal) joint diagonalization algorithms are of interest on their own, e.g., in the *generalized covariance matrices* framework. Note that the latter is an important area of research on its own and further results, e.g., on the sample complexity of the generalized covariance matrices or showing asymptotic equivalence (which is observed experimentally) of the higher-order cumulants-based algorithms with the generalized covariance matrices-based algorithms, are of interest.

An important question to answer for joint diagonalization methods is the choice of the number of target matrices or contraction vectors. Indeed, it is experimentally observed that choosing the number of matrices of the order of the logarithm of the dimension is sufficient. However, no theoretical results in this respect are known. Moreover, the results on the convergence rate and global convergence properties of

joint diagonalization algorithms are very limited. It would be interesting, e.g., to perform analysis similar to the one of [Colombo and Vlassis, 2016a,b,c] to better understand these properties.

The Overcomplete Case

The estimation in overcomplete models is a much more difficult task, but, nevertheless, it is still possible in some special cases or under additional assumptions [Cardoso, 1991, Yeredor, 2002, De Lathauwer et al., 2007, Anandkumar et al., 2015c]. Extension of the ideas presented in this thesis to the overcomplete case is another important direction for the future research.

5.2 Inference for Semiparametric Models

The models introduced in Chapters 3 and 4 of this thesis are semiparametric and supposed to adapt to the densities of the latent sources. In the signal processing literature, some methods are known [see, e.g. Comon and Jutten, 2010] for density approximation and can be directly applied for the inference in the introduced semiparametric models. Therefore, developing inference methods for the discrete ICA and discrete, non-Gaussian, and mixed CCA models is another interesting direction for the future research.

Annexe A

Notation

In Appendix A, we outline the list of probability distributions which are widely used in this thesis.

A.1 The List of Probability Distributions

In this section, we recall the probability distributions used in this thesis.

Multivariate Gaussian. The probability density function of the *multivariate Gaussian* distribution with the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ of an $\mathbb{R}^M$-valued continuous random variable $\mathbf{x}$ is :

$$\mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^M |\boldsymbol{\Sigma}|}} \exp\left[\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right], \tag{A.1}$$

where the covariance matrix $\boldsymbol{\Sigma}$ is strictly positive definite and $|\boldsymbol{\Sigma}|$ is its determinant. We write $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to denote that $\mathbf{x}$ is a Gaussian random variable with the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$.

Dirichlet. The probability distribution function of the *Dirichlet* distribution is :

$$p(\boldsymbol{\theta}|\mathbf{c}) := \frac{\Gamma(c_0)}{\prod_{k=1}^K \Gamma(c_k)} \prod_{k=1}^K \theta_k^{c_k - 1} \mathbb{I}(\boldsymbol{\theta} \in \boldsymbol{\Delta}_K), \tag{A.2}$$

where the parameter $\mathbf{c} \in \mathbb{R}^K_{++}$, the parameter $c_0 := \sum_{k=1}^K c_k$, $\Gamma(\cdot)$ is the Gamma function,[1] and $\mathbb{I}(\cdot)$ is the indicator function (it is 1 if the condition in the brackets is true and 0 otherwise). The Dirichlet distribution is supported on the $(K-1)$-simplex. We write $\boldsymbol{\theta} \sim \text{Dirichlet}(\mathbf{c})$ to denote that $\boldsymbol{\theta}$ is a Dirichlet variable with the parameter $\mathbf{c}$.

1. For complex numbers with a positive real part, the *gamma function* is defined via a convergent improper integral $\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx$.

When all elements of $\mathbf{c}$ are the same, $c_1 = c_2 = \cdots = c_K$, the Dirichlet distribution is *symmetric*. When $\mathbf{c} = \mathbf{1}$, the Dirichlet distribution is equivalent to the *uniform* distribution on the $(K-1)$-simplex. When every $c_k > 1$ (or $c_0 > K$), then the mode of the density function is somewhere in the middle of the simplex, therefore, most or all elements in a sampled vector $\boldsymbol{\theta}$ are likely to be significantly larger than zero. When every $c_k < 1$ (or $c_0 < K$), then the mode of the density function is almost at the vertices of the simplex, therefore, only few elements in a sampled vector $\boldsymbol{\theta}$ are likely to be significantly large than zero. In the limit $c_0 \to 0$, only one element of a sampled vector is significantly larger than zero (and nearly equal to one). It is useful sometimes to write the Dirichlet parameter $\mathbf{c}$ as a product $\mathbf{c} = c\mathbf{n}$, where c is called the *concentration parameter* and $\mathbf{n}$ is the *base measure*. See also Frigyik et al. [2010].

Multinomial and Discrete. The *multinomial* distribution models the trials for rolling M-sided dice L times. A multinomial random vector $\mathbf{x}$, which takes non-negative discrete values which sum to L, i.e. $\sum_{m=1}^{M} x_m = L$, has the following probability mass function :

$$p(\mathbf{x}|L, \mathbf{y}) = \frac{L!}{\prod_{m=1}^{M} x_m!} \prod_{m=1}^{M} y_m^{x_m}, \tag{A.3}$$

where the parameter $\mathbf{y} \in \boldsymbol{\Delta}_M$. We write $\mathbf{x} \sim \text{Mult}(L; \mathbf{y})$ to denote that $\mathbf{x}$ is a multinomial variable for L trials with the parameter $\mathbf{y}$. When $L = 1$, the multinomial distribution is equivalent to the *discrete* distribution :

$$p(\mathbf{x}|\mathbf{y}) = \prod_{m=1}^{M} y_m^{x_m}, \tag{A.4}$$

and we denote $\mathbf{x} \sim \text{Mult}(1, \mathbf{y})$. For such a vector $\mathbf{x}$, which has only one non-negative m-th element equal to 1 and the rest is zero, we say that it is represented using *one-hot* encoding.

Poisson. The probability mass function of the *Poisson* distribution with the parameter λ of a random variable x, which takes positive discrete values, is

$$p(x|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}, \tag{A.5}$$

where $\lambda \in \mathbb{R}_{++}$. We write $x \sim \text{Poisson}(\lambda)$ to denote that x is a Poisson random variable with the parameter λ. We also write $\mathbf{x} \sim \text{Poisson}(\boldsymbol{\lambda})$ to denote that each element x_m of $\mathbf{x}$ is a Poisson random variable with the parameter λ_m. In Chapter 3, we use the fact that all cumulants of a Poisson random variable are equal to each other and equal to the parameter λ (see also Section 2.2).

Gamma. The density function of the *gamma* distribution with the *shape* c and *rate* b

parameters of an $\mathbb{R}_{++}$-valued random variable x is

$$p(x|c,b) = \frac{b^c}{\Gamma(c)} x^{c-1} e^{-bx}, \tag{A.6}$$

where the parameters $c \in \mathbb{R}_{++}$ and $b \in \mathbb{R}_{++}$. We write $x \sim \text{Gamma}(c,b)$ to denote that x is a gamma random variable with the shape and rate parameters c and b respectively. We write $\mathbf{x} \sim \text{Gamma}(\mathbf{c},\mathbf{b})$ to denote that each element x_m of $\mathbf{x}$ is a gamma random variable with the shape and rate parameters c_m and b_m, respectively. Note that this definition is preferable to the one with the scale parameter (equal to $1/b$) since the density (A.6) leads to a convex optimization problem for the maximum likelihood estimation as opposed to the other one.

Note that we mostly omit the dependence on parameters in the notation, e.g. instead on $p(\mathbf{x}|\boldsymbol{\theta})$ we write $p(\mathbf{x})$, and we use conditional dependence to show the dependence on latent variables only, e.g. $p(\mathbf{x}|\mathbf{z})$, where $\mathbf{z}$ is a latent variable.

Annexe B

Discrete ICA

Appendix B is organized as follows :

- In Appendix B.1, we derive the symmetric canonical polyadic (a.k.a. diagonal) form of the population third-order cumulant of the DICA model.
- In Appendix B.2, we outline the proof of the sample complexity result for the DICA model (Proposition 3.3.1).

B.1 The Order-Three DICA Cumulant

In this section, we derive the order-3 DICA (and GP) cumulant. See Section 2.2 for the definition and properties of cumulants.

We also use the property that all cumulants of a Poisson random variable with a parameter λ are equal to this parameter.

For a Poisson random variable with the parameter λ, all cumulants are equal to this parameter (see Section 2.2). Therefore the order-3 cumulant of a univariate Poisson random variable x_m with the parameter y_m is

$$\mathbb{E}((x_m - \mathbb{E}(x_m))^3 \,|\, y_m) = y_m.$$

By the independence property of cumulants, the order-3 cumulant of $\mathbf{x} \,|\, \mathbf{y}$ is a diagonal tensor with the (m_1, m_2, m_3)-th element equal to

$$\operatorname{cum}(x_{m_1}, x_{m_2}, x_{m_3} \,|\, \mathbf{y}) = \delta(m_1, m_2, m_3)\, y_{m_1}, \tag{B.1}$$

where δ is the Kronecker delta. Recall that the low of total cumulance reads as

$$\begin{aligned}
&\operatorname{cum}(x_{m_1}, x_{m_2}, x_{m_3}) = \mathbb{E}\left[\operatorname{cum}(x_{m_1}, x_{m_2}, x_{m_3} \,|\, \mathbf{y})\right] \\
&+ \operatorname{cum}\left[\mathbb{E}(x_{m_1} \,|\, \mathbf{y}), \mathbb{E}(x_{m_2} \,|\, \mathbf{y}), \mathbb{E}(x_{m_3} \,|\, \mathbf{y})\right] + \operatorname{cov}\left[\mathbb{E}(x_{m_1} \,|\, \mathbf{y}), \operatorname{cov}(x_{m_2}, x_{m_3} \,|\, \mathbf{y})\right] \\
&+ \operatorname{cov}\left[\mathbb{E}(x_{m_2} \,|\, \mathbf{y}), \operatorname{cov}(x_{m_1}, x_{m_3} \,|\, \mathbf{y})\right] + \operatorname{cov}\left[\mathbb{E}(x_{m_3} \,|\, \mathbf{y}), \operatorname{cov}(x_{m_1}, x_{m_2} \,|\, \mathbf{y})\right].
\end{aligned}$$

Substituting the cumulant (B.1) of $\mathbf{x}\,|\,\mathbf{y}$ into this law of total cumulance, we obtain

$$\begin{aligned}
&\mathrm{cum}(x_{m_1},\, x_{m_2},\, x_{m_3}) = \delta(m_1,\, m_2,\, m_3)\mathbb{E}(y_{m_1}) + \mathrm{cum}(y_{m_1},\, y_{m_2},\, y_{m_3}) \\
&\quad + \delta(m_2,\, m_3)\mathrm{cov}(y_{m_1},\, y_{m_2}) + \delta(m_1,\, m_3)\mathrm{cov}(y_{m_1},\, y_{m_2}) + \delta(m_1,\, m_2)\mathrm{cov}(y_{m_1},\, y_{m_3}) \\
&= \mathrm{cum}(y_{m_1},\, y_{m_2},\, y_{m_3}) - 2\delta(m_1,\, m_2,\, m_3)\mathbb{E}(x_{m_1}) \\
&\quad + \delta(m_2,\, m_3)\mathrm{cov}(x_{m_1},\, x_{m_2}) + \delta(m_1,\, m_3)\mathrm{cov}(x_{m_1},\, x_{m_2}) + \delta(m_1,\, m_2)\mathrm{cov}(x_{m_1},\, x_{m_3}),
\end{aligned}$$

where we used the previous result from (3.13) that $\mathrm{cov}(\mathbf{y},\, \mathbf{y}) = \mathrm{cov}(\mathbf{x},\, \mathbf{x}) - \mathrm{Diag}(\mathbb{E}(\mathbf{x}))$. Finally, by the multilinearity property for $\mathrm{cum}(y_{m_1},\, y_{m_2},\, y_{m_3})$, we obtain

$$\begin{aligned}
\mathrm{cum}(x_{m_1},\, x_{m_2},\, x_{m_3}) &= \left[\mathrm{cum}(\boldsymbol{\alpha}, \boldsymbol{\alpha}, \boldsymbol{\alpha}) \times_1 \mathbf{D}^\top \times_2 \mathbf{D}^\top \times_3 \mathbf{D}^\top)\right]_{m_1 m_2 m_3} \\
&\quad - 2\delta(m_1,\, m_2,\, m_3)\mathbb{E}(x_{m_1}) \quad + \delta(m_2,\, m_3)\mathrm{cov}(x_{m_1},\, x_{m_2}) \\
&\quad + \delta(m_1,\, m_3)\mathrm{cov}(x_{m_1},\, x_{m_2}) \quad + \delta(m_1,\, m_2)\mathrm{cov}(x_{m_1},\, x_{m_3}),
\end{aligned} \tag{B.2}$$

where, in the third equality, we used the previous result from (3.13) that $\mathrm{cov}(\mathbf{y},\, \mathbf{y}) = \mathrm{cov}(\mathbf{x},\, \mathbf{x}) - \mathrm{Diag}(\mathbb{E}(\mathbf{x}))$.

B.2 The Sketch of the Proof for Proposition 3.3.1

B.2.1 Expected Squared Error for the Sample Expectation

The sample expectation is $\widehat{\mathbb{E}}(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$ is an unbiased estimator of the expectation and its squared error is :[1]

$$\begin{aligned}
&\mathbb{E}\left(\|\widehat{\mathbb{E}}(\mathbf{x}) - \mathbb{E}(\mathbf{x})\|_2^2\right) = \sum_{m=1}^{M}\mathbb{E}\left[\left(\widehat{\mathbb{E}}(x_m) - \mathbb{E}(x_m)\right)^2\right] \\
&= N^{-2}\sum_{m=1}^{M}\left[\mathbb{E}\left(\sum_{n=1}^{N}\left(x_{nm} - \mathbb{E}(x_m)\right)^2\right)\right. \\
&\left. + \mathbb{E}\left(\sum_{n=1}^{N}\sum_{\substack{n'=1 \\ n'\neq n}}^{N}\left(x_{nm} - \mathbb{E}(x_m)\right)\left(x_{n'm} - \mathbb{E}(x_m)\right)\right)\right] \\
&= N^{-1}\sum_{m=1}^{M}\mathbb{E}\left[\left(x_m - \mathbb{E}(x_m)\right)^2\right] = N^{-1}\sum_{m=1}^{M}\mathrm{var}(x_m).
\end{aligned}$$

1. Note that these derivations are partially based on the lecture notes to the "Machine Learning" course read at Saarland University in winter semester of 2011/2012 by Prof. M. Hein.

Further, by the law of total variance :

$$\mathbb{E}\left(\|\widehat{\mathbb{E}}(\mathbf{x}) - \mathbb{E}(\mathbf{x})\|_2^2\right) = N^{-1}\sum_{m=1}^{M}\left[\mathbb{E}(\mathrm{var}(x_m|\mathbf{y})) + \mathrm{var}(\mathbb{E}(x_m|\mathbf{y}))\right]$$
$$= N^{-1}\sum_{m=1}^{M}\left[\mathbb{E}(y_m) + \mathrm{var}(y_m)\right] = N^{-1}\left[\sum_{k=1}^{K}\mathbb{E}(\alpha_k) + \sum_{k=1}^{K}\langle \mathbf{d}_k, \mathbf{d}_k\rangle \mathrm{var}(\alpha_k)\right],$$

using the fact that $\sum_{m=1}^{M} D_{mk} = 1$ for any k.

B.2.2 Expected Squared Error for the Sample Covariance

The following finite sample estimator of the covariance matrix, defined as $\mathrm{cov}(\mathbf{x}, \mathbf{x}) = \mathbb{E}(\mathbf{x}\mathbf{x}^\top) - \mathbb{E}(\mathbf{x})\mathbb{E}(\mathbf{x})^\top$,

$$\begin{aligned}\widehat{\mathrm{cov}}(\mathbf{x}, \mathbf{x}) &= (N-1)^{-1}\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top - \widehat{\mathbb{E}}(\mathbf{x})\widehat{\mathbb{E}}(\mathbf{x})^\top \\ &= (N-1)^{-1}\sum_{n=1}^{N}\left(\mathbf{x}_n\mathbf{x}_n^\top - N^{-2}\sum_{n'=1}^{N}\sum_{n''=1}^{N}\mathbf{x}_{n'}\mathbf{x}_{n''}^\top\right) \\ &= N^{-1}\sum_{n=1}^{N}\left(\mathbf{x}_n\mathbf{x}_n^\top - (N-1)^{-1}\mathbf{x}_n\sum_{\substack{n'=1\\ n'\neq n}}^{N}\mathbf{x}_{n'}^\top\right)\end{aligned} \tag{B.3}$$

is unbiased, i.e., $\mathbb{E}(\widehat{\mathrm{cov}}(\mathbf{x}, \mathbf{x})) = \mathrm{cov}(\mathbf{x}, \mathbf{x})$. Its squared error is equal to

$$\mathbb{E}\left(\|\widehat{\mathrm{cov}}(\mathbf{x}, \mathbf{x}) - \mathrm{cov}(\mathbf{x}, \mathbf{x})\|_F^2\right) = \sum_{m=1}^{M}\sum_{m'=1}^{M}\mathbb{E}\left[\left(\widehat{\mathrm{cov}}(x_m, x_{m'}) - \mathbb{E}[\widehat{\mathrm{cov}}(x_m, x_{m'})]\right)^2\right].$$

The (m, m')-th element of the sum above is equal to :

$$\frac{1}{N^2}\sum_{n,n'}\mathrm{cov}\left(x_{nm}x_{nm'} - \frac{1}{N-1}x_{nm}\sum_{n''\neq n}x_{n''m'} \quad , \quad x_{n'm}x_{n'm'} - \frac{1}{N-1}x_{n'm}\sum_{n'''\neq n'}x_{n'''m'}\right)$$
$$= \frac{1}{N^2}\sum_{n,n'}\mathrm{cov}\left(x_{nm}x_{nm'}, x_{n'm}x_{n'm'}\right) - \frac{2}{N^2(N-1)}\sum_{n,n'}\mathrm{cov}\left(x_{nm}\sum_{n''\neq n}x_{n''m'}, x_{n'm}x_{n'm'}\right)$$
$$+ \frac{1}{N^2(N-1)^2}\sum_{n,n'}\mathrm{cov}\left(x_{nm}\sum_{n''\neq n}x_{n''m'}, x_{n'm}\sum_{n'''\neq n'}x_{n'''m'}\right);$$

this (m, m')-th element is further equal to :

$$\frac{1}{N^2}\sum_{n=1}^{N}\operatorname{cov}\left(x_{nm}x_{nm'}, x_{nm}x_{nm'}\right)$$
$$-\frac{2}{N^2(N-1)}\left[\sum_{n}\sum_{n''}\operatorname{cov}\left(x_{nm}x_{n''m'},\, x_{nm}x_{nm'}\right)+\sum_{n}\sum_{n'}\operatorname{cov}\left(x_{nm}x_{n'm'},\, x_{n'm}x_{n'm'}\right)\right.$$
$$+\sum_{n}\sum_{n''}\sum_{n'''}\operatorname{cov}\left(x_{nm}x_{n''m'},\, x_{nm}x_{n'''m'}\right)+\sum_{n'}\sum_{n}\sum_{n''}\operatorname{cov}\left(x_{nm}x_{n''m'},\, x_{n'm}x_{nm'}\right)$$
$$\left.+\sum_{n'}\sum_{n}\sum_{n'''}\operatorname{cov}\left(x_{nm}x_{n'm'},\, x_{n'm}x_{n'''m'}\right)+\sum_{n'}\sum_{n}\sum_{n''}\operatorname{cov}\left(x_{nm}x_{n''m'},\, x_{n'm}x_{n''m'}\right)\right],$$

where the summations of the form $\sum_n \sum_{n'}$ denote $\sum_{n=1}^{N}\sum_{n'=1,\, n'\neq n}^{N}$ and the summations of the form $\sum_n \sum_{n'} \sum_{n''}$ denote $\sum_{n=1}^{N}\sum_{n'=1,\, n'\neq n}^{N}\sum_{n''=1,\, n''\neq n', n''\neq n}^{N}$. We also used mutual independence of the observations $\mathbf{x}_n$ in a sample $\{\mathbf{x}_n\}_{n=1}^{N}$ to conclude that the covariance between the two expressions involving only independent variables is zero. Substituting these elements back to the sum, we get :

$$\mathbb{E}\left(\|\widehat{\operatorname{cov}}(\mathbf{x},\mathbf{x})-\operatorname{cov}(\mathbf{x},\mathbf{x})\|_F^2\right)=\frac{1}{N^2}\sum_{m,m'}N\left(\mathbb{E}(x_m^2x_{m'}^2)-\left[\mathbb{E}(x_mx_{m'})\right]^2\right)$$
$$-\frac{4}{N^2(N-1)}\sum_{m,m'}N(N-1)\left(\mathbb{E}(x_m^2x_{m'})\mathbb{E}(x_{m'})-\mathbb{E}(x_mx_{m'})\mathbb{E}(x_m)\mathbb{E}(x_{m'})\right)$$
$$+\frac{2}{N^2(N-1)^2}\sum_{m,m'}N(N-1)(N-2)\left(\mathbb{E}(x_m^2)\left[\mathbb{E}(x_{m'})\right]^2-\left[\mathbb{E}(x_m)\right]^2\left[\mathbb{E}(x_{m'})\right]^2\right)$$
$$+\frac{2}{N^2(N-1)^2}\sum_{m,m'}N(N-1)(N-2)\left(\mathbb{E}(x_mx_{m'})\mathbb{E}(x_m)\mathbb{E}(x_{m'})-\left[\mathbb{E}(x_m)\right]^2\left[\mathbb{E}(x_{m'})\right]^2\right)$$
$$+O\left(N^{-2}\right),$$

where some terms with N are left to emphasize that we used the fact that each document is independent and identically distributed. The summation of the form $\sum_{m,m'}$ denote $\sum_{m=1}^{M}\sum_{m'=1}^{M}$. Subsequent simplification gives :

$$\mathbb{E}\left(\|\widehat{\operatorname{cov}}(\mathbf{x},\mathbf{x})-\operatorname{cov}(\mathbf{x},\mathbf{x})\|_F^2\right)=\frac{1}{N}\sum_{m,m'}\left[\operatorname{var}(x_mx_{m'})+2\left[\mathbb{E}(x_m)\right]^2\operatorname{var}(x_{m'})\right]$$
$$+\frac{1}{N}\sum_{m,m'}\left[2\mathbb{E}(x_m)\mathbb{E}(x_{m'})\operatorname{cov}(x_m,x_{m'})-4\mathbb{E}(x_m)\operatorname{cov}(x_mx_{m'},x_{m'})\right]+O\left(N^{-2}\right),$$

where in the last equality, by symmetry, the summation indexes m and m' can be exchanged. As $x_m \sim \text{Poisson}(y_m)$, by the law of total expectation and law of total covariance, it follows, for $m \neq m'$ (and using the auxiliary expressions from Sec-

tion B.2.4) :

$$\begin{aligned}\operatorname{var}(x_m x_{m'}) &= \mathbb{E}(x_m^2 x_{m'}^2) - [\mathbb{E}[x_m x_{m'}]]^2 = \mathbb{E}\left[\mathbb{E}(x_m^2 x_{m'}^2|\mathbf{y})\right] - [\mathbb{E}\left[\mathbb{E}(x_m x_{m'}|\mathbf{y})\right]]^2 \\ &= \mathbb{E}\left[y_m^2 y_{m'}^2 + y_m^2 y_{m'} + y_m y_{m'}^2 + y_m y_{m'}\right] - \left[\mathbb{E}(y_m y_{m'})\right]^2,\end{aligned}$$

$$\begin{aligned}\left[\mathbb{E}(x_m)\right]^2 \operatorname{var}(x_{m'}) &= \left[\mathbb{E}(y_m)\right]^2 \mathbb{E}(y_{m'}) + \left[\mathbb{E}(y_m)\right]^2 \mathbb{E}(y_{m'}^2) - \left[\mathbb{E}(y_m)\right]^2 \left[\mathbb{E}(y_{m'})\right]^2, \\ \mathbb{E}(x_m)\mathbb{E}(x_{m'})\operatorname{cov}(x_m, x_{m'}) &= \mathbb{E}(y_m y_{m'})\mathbb{E}(y_m)\mathbb{E}(y_{m'}) - \left[\mathbb{E}(y_m)\right]^2 \left[\mathbb{E}(y_{m'})\right]^2, \\ \mathbb{E}(x_m)\operatorname{cov}(x_m x_{m'}, x_{m'}) &= \mathbb{E}(y_m)\left[\mathbb{E}(y_m y_{m'}) + \mathbb{E}(y_m y_{m'}^2) - \mathbb{E}(y_m y_{m'})\mathbb{E}(y_{m'})\right].\end{aligned}$$

Now, considering the $m = m'$ case, we have :

$$\begin{aligned}\operatorname{var}(x_m^2) &= \mathbb{E}[\mathbb{E}(x_m^4|\mathbf{y})] - \left[\mathbb{E}[\mathbb{E}(x_m^2|\mathbf{y})]\right]^2 \\ &= \mathbb{E}\left[y_m^4 + 6y_m^3 + 7y_m^2 + y_m\right] - \left[\mathbb{E}\left[y_m^2 + y_m\right]\right]^2, \\ \mathbb{E}(x_m)\mathbb{E}(x_m)\operatorname{cov}(x_m, x_m) &= \mathbb{E}(y_m)^2\left[\mathbb{E}(y_m^2) + \mathbb{E}(y_m) - \left[\mathbb{E}(y_m)\right]^2\right], \\ \mathbb{E}(x_m)\operatorname{cov}(x_m^2, x_m) &= \mathbb{E}(y_m)\left[\mathbb{E}(y_m^3) + 3\mathbb{E}(y_m^2) + \mathbb{E}(y_m) - \mathbb{E}(y_m)\left[\mathbb{E}(y_m^2) + \mathbb{E}(y_m)\right]\right].\end{aligned}$$

Substitution of $y_m = \sum_{k=1}^K D_{mk}\alpha_k$ gives the following

$$\begin{aligned}\mathbb{E}\left(\|\widehat{\operatorname{cov}}(\mathbf{x}, \mathbf{x}) - \operatorname{cov}(\mathbf{x}, \mathbf{x})\|_F^2\right) &= N^{-1} \sum_{k,k',k'',k'''} \langle \mathbf{d}_k, \mathbf{d}_{k'}\rangle\langle \mathbf{d}_{k''}, \mathbf{d}_{k'''}\rangle \mathcal{A}_{kk'k''k'''} \\ &+ N^{-1} \sum_{k,k',k''} \left[\langle \mathbf{d}_k, \mathbf{d}_{k'}\rangle\langle \mathbf{d}_{k''}, \mathbf{1}\rangle \mathcal{B}_{kk'k''} + \langle \mathbf{d}_k \circ \mathbf{d}_{k'}, \mathbf{d}_{k''}\rangle \mathcal{E}_{kk'k''}\right] \\ &+ N^{-1} \sum_{k,k'} \left[\langle \mathbf{d}_k, \mathbf{1}\rangle\langle \mathbf{d}_{k'}, \mathbf{1}\rangle \mathbb{E}(\alpha_k\alpha_{k'}) + \langle \mathbf{d}_k, \mathbf{d}_{k'}\rangle \mathcal{F}_{kk'}\right] + \sum_k \langle \mathbf{d}_k, \mathbf{1}\rangle \mathbb{E}(\alpha_k) + O\left(N^{-2}\right),\end{aligned}$$

where $\mathbf{1}$ is the vector with all the elements equal to 1, the sign $\circ$ is the element-wise Hadamard product, and

$$\begin{aligned}\mathcal{A}_{kk'k''k'''} &= \mathbb{E}(\alpha_k\alpha_{k'}\alpha_{k''}\alpha_{k'''}) - \mathbb{E}(\alpha_k\alpha_{k''})\mathbb{E}(\alpha_{k'}\alpha_{k'''}) + 2\mathbb{E}(\alpha_k)\mathbb{E}(\alpha_{k'})\mathbb{E}(\alpha_{k''}\alpha_{k'''}) \\ &- 2\mathbb{E}(\alpha_k)\mathbb{E}(\alpha_{k'})\mathbb{E}(\alpha_{k''})\mathbb{E}(\alpha_{k'''}) + 2\mathbb{E}(\alpha_k\alpha_{k''})\mathbb{E}(\alpha_{k'})\mathbb{E}(\alpha_{k'''}) - 2\mathbb{E}(\alpha_k)\mathbb{E}(\alpha_{k'})\mathbb{E}(\alpha_{k''})\mathbb{E}(\alpha_{k'''}) \\ &- 4\mathbb{E}(\alpha_k)\mathbb{E}(\alpha_{k'}\alpha_{k''}\alpha_{k'''}) + 4\mathbb{E}(\alpha_k)\mathbb{E}(\alpha_{k'}\alpha_{k''})\mathbb{E}(\alpha_{k'''}),\end{aligned}$$

$$\begin{aligned}\mathcal{B}_{kk'k''} &= 2\mathbb{E}(\alpha_k\alpha_{k'}\alpha_{k''}) + 2\mathbb{E}(\alpha_k)\mathbb{E}(\alpha_{k'})\mathbb{E}(\alpha_{k''}) - 4\mathbb{E}(\alpha_k)\mathbb{E}(\alpha_{k'}\alpha_{k''}), \\ \mathcal{E}_{kk'k''} &= 4\mathbb{E}(\alpha_k\alpha_{k'}\alpha_{k''}) + 6\mathbb{E}(\alpha_k)\mathbb{E}(\alpha_{k'})\mathbb{E}(\alpha_{k''}) - 10\mathbb{E}(\alpha_k\alpha_{k'})\mathbb{E}(\alpha_{k''}), \\ \mathcal{F}_{kk'} &= 6\mathbb{E}(\alpha_k\alpha_{k'}) - 5\mathbb{E}(\alpha_k)\mathbb{E}(\alpha_{k'}),\end{aligned}$$

where we used the expressions from Section B.2.4.

B.2.3 Expected Squared Error of the Estimator $\widehat{\mathbf{S}}$ for the GP/DICA Cumulants

As the estimator $\widehat{S}$ (3.19) of S (3.14) is unbiased, its expected squared error is

$$\begin{aligned}\mathbb{E}\left[\|\widehat{\mathbf{S}}-\mathbf{S}\|_F^2\right] &= \mathbb{E}\left[\left\|(\widehat{\mathrm{cov}}(\mathbf{x},\mathbf{x})-\mathrm{cov}(\mathbf{x},\mathbf{x}))+\left(\mathrm{Diag}[\widehat{\mathbb{E}}(\mathbf{x})]-\mathrm{Diag}\left[\mathbb{E}(\mathbf{x})\right]\right)\right\|_F^2\right] \\ &= \mathbb{E}\left[\|\widehat{\mathbb{E}}(\mathbf{x})-\mathbb{E}(\mathbf{x})\|_F^2\right]+\mathbb{E}\left[\|\widehat{\mathrm{cov}}(\mathbf{x},\mathbf{x})-\mathrm{cov}(\mathbf{x},\mathbf{x})\|_F^2\right] \\ &\quad + 2\sum_{m=1}^{M}\mathbb{E}\left[\left(\widehat{\mathbb{E}}(x_m)-\mathbb{E}(x_m)\right)\left(\widehat{\mathrm{cov}}(x_m,x_m)-\mathrm{cov}(x_m,x_m)\right)\right].\end{aligned} \tag{B.4}$$

As $\widehat{\mathbb{E}}(x_m)$ and $\widehat{\mathrm{cov}}(x_m,x_m)$ are unbiased, the m-th element of the last sum is equal to

$$\begin{aligned}&\mathrm{cov}\left[\widehat{\mathbb{E}}(x_m),\widehat{\mathrm{cov}}(x_m,x_m)\right] \\ &= N^{-2}\sum_{n,n'}\mathrm{cov}\left[x_{nm},x_{n'm}^2\right]-N^{-2}(N-1)^{-1}\sum_{n,n',n''\neq n'}\mathrm{cov}\left[x_{nm},x_{n'm}x_{n''m}\right] \\ &= N^{-2}\sum_{n}\mathrm{cov}\left[x_{nm},x_{nm}^2\right]-2N^{-2}(N-1)^{-1}\sum_{n,n'\neq n}\mathrm{cov}\left[x_{nm},x_{n'm}x_{nm}\right]+O\left(N^{-2}\right) \\ &= N^{-1}\mathbb{E}(x_m^3)-2N^{-1}\left(\mathbb{E}(x_m^2)\mathbb{E}(x_m)-\left[\mathbb{E}(x_m)\right]^3\right)+O\left(N^{-2}\right) \\ &\leq N^{-1}\mathbb{E}(x_m^3)+2N^{-1}\left[\mathbb{E}(x_m)\right]^3+O\left(N^{-2}\right) \\ &= N^{-1}\left[\mathbb{E}(y_m^3)+3\mathbb{E}(y_m^2)+\mathbb{E}(y_m)+2\left[\mathbb{E}(y_m)\right]^3\right]+O\left(N^{-2}\right),\end{aligned}$$

where we neglected the negative term $-\mathbb{E}(x_m^2)\mathbb{E}(x_m)$ for the inequality, and the last equality follows from the expressions in Section B.2.4. Further, the fact that $y_m = \sum_{k=1}^{K} D_{mk}\alpha_k$ gives

$$\begin{aligned}\sum_{m=1}^{M}\mathrm{cov}\left[\widehat{\mathbb{E}}(x_m),\widehat{\mathrm{cov}}(x_m,x_m)\right] &= N^{-1}\sum_{k,k',k''}\langle\mathbf{d}_k\circ\mathbf{d}_{k'},\mathbf{d}_{k''}\rangle\mathcal{C}_{kk'k''} \\ &\quad + 3N^{-1}\sum_{k,k'}\langle\mathbf{d}_k,\mathbf{d}_{k'}\rangle\mathbb{E}(\alpha_k\alpha_{k'})+N^{-1}\sum_{k}\langle\mathbf{d}_k,\mathbf{1}\rangle\mathbb{E}(\alpha_k)+O\left(N^{-2}\right),\end{aligned}$$

where $\circ$ denotes the element-wise Hadamard product and

$$\mathcal{C}_{kk'k''} = \mathbb{E}(\alpha_k\alpha_{k'}\alpha_{k''})+2\mathbb{E}(\alpha_k)\mathbb{E}(\alpha_{k'})\mathbb{E}(\alpha_{k''}).$$

Plugging this and the expressions for $\mathbb{E}(\|\widehat{\mathbb{E}}(\mathbf{x})-\mathbb{E}(\mathbf{x})\|_F^2)$ and $\mathbb{E}(\|\widehat{\mathrm{cov}}(\mathbf{x},\mathbf{x})-\mathrm{cov}(\mathbf{x},\mathbf{x})\|_F^2)$ from Sections B.2.1 and B.2.2, respectively, into (B.4) gives

$$\begin{aligned}
\mathbb{E}\left[\|\widehat{\mathbf{S}}-\mathbf{S}\|_F^2\right] &= N^{-1}\sum_k \langle \mathbf{d}_k,\mathbf{d}_k\rangle \mathrm{var}(\alpha_k) + N^{-1}\sum_k \mathbb{E}(\alpha_k) \\
&+ N^{-1}\sum_{k,k',k'',k'''} \langle \mathbf{d}_k,\mathbf{d}_{k'}\rangle\langle \mathbf{d}_{k''},\mathbf{d}_{k'''}\rangle \mathcal{A}_{kk'k''k'''} \\
&+ N^{-1}\sum_{k,k',k''} \left[\langle \mathbf{d}_k,\mathbf{d}_{k'}\rangle \mathcal{B}_{kk'k''} + 2\langle \mathbf{d}_k\circ\mathbf{d}_{k'},\mathbf{d}_{k''}\rangle \mathcal{C}_{kk'k''}\right] \\
&+ N^{-1}\sum_{k,k'} \left(1+6\,\langle \mathbf{d}_k,\mathbf{d}_{k'}\rangle\right)\mathbb{E}(\alpha_k\alpha_{k'}) + 2N^{-1}\sum_k \mathbb{E}(\alpha_k) + O\left(N^{-2}\right),
\end{aligned}$$

where we used that, by the simplex constraint on the topics, $\langle \mathbf{d}_k,\mathbf{1}\rangle = 1$ for all k. To analyze this expression in more details, let us now consider the GP model, i.e., $\alpha_k \sim \mathrm{Gamma}(c_k,b)$:

$$\sum_{k,k',k'',k'''} \mathcal{A}_{kk'k''k'''} \leq \frac{30c_0^4+23c_0^3+14c_0^2+8c_0}{b^4}, \quad \text{and} \quad \sum_{k,k',k''} \mathcal{B}_{kk'k''} \leq \frac{6c_0^3+10c_0^2+4c_0}{b^3},$$
$$\sum_{k,k',k''} \mathcal{C}_{kk'k''} \leq \frac{7c_0^3+6c_0^2+2c_0}{b^3}, \quad \text{and} \quad \sum_{k,k',k''} \mathcal{E}_{kk'k''} \leq \frac{12c_0^3+10c_0^2+8c_0}{b^3},$$
$$\sum_{k,k'} \mathcal{F}_{kk'} \leq \frac{2c_0^2+c_0}{b^2} \quad \text{and} \quad \sum_{k,k'} \mathbb{E}(\alpha_k\alpha_{k'}) \leq \frac{2c_0^2+c_0}{b^2},$$

where we used the expressions from Section B.2.4, which gives

$$\begin{aligned}
\mathbb{E}\left[\|\widehat{\mathbf{S}}-\widehat{\mathbf{S}}\|_F^2\right] &\leq \nu N^{-1}\left[\max_k \|\mathbf{d}_k\|_2^2\frac{c_0}{b^2} + \frac{c_0}{b} + \left(\max_{k,k'}\langle \mathbf{d}_k,\mathbf{d}_{k'}\rangle\right)^2 \max\left[\frac{c_0^4}{b^4},\frac{c_0}{b^4}\right]\right] \\
&+ \nu N^{-1}\left[\max_{k,k'}\langle \mathbf{d}_k,\mathbf{d}_{k'}\rangle \max\left[\frac{c_0^3}{b^3},\frac{c_0}{b^3}\right] + \left(\max_{k,k',k''}\langle \mathbf{d}_k\circ\mathbf{d}_{k'},\mathbf{d}_{k''}\rangle\right)\max\left[\frac{c_0^3}{b^3},\frac{c_0}{b^3}\right]\right] \\
&+ \nu N^{-1}\left[\left(1+\max_{k,k'}\langle \mathbf{d}_k,\mathbf{d}_{k'}\rangle\right)\max\left[\frac{c_0^2}{b^2},\frac{c_0}{b^2}\right]\right] + O\left(N^{-2}\right),
\end{aligned}$$

where $\nu \leq 30$ is a universal constant. As, by the Cauchy-Schwarz inequality,

$$\max_{k,k'}\langle \mathbf{d}_k,\mathbf{d}_{k'}\rangle \leq \max_k \|\mathbf{d}_k\|_2^2 =: \Delta_1,$$
$$\max_{k,k',k''}\langle \mathbf{d}_k\circ\mathbf{d}_{k'},\mathbf{d}_{k''}\rangle \leq \max_k \|\mathbf{d}_k\|_\infty \|\mathbf{d}_k\|_2^2 \leq \max_k \|\mathbf{d}_k\|_2^3 =: \Delta_2.$$

(note that for the topics in the simplex, $\Delta_2 \leq \Delta_1$ as well as $\Delta_1^2 \leq \Delta_1$), it follows that

$$\mathbb{E}\left[\|\widehat{\mathbf{S}} - \mathbf{S}\|_F^2\right] \leq \nu N^{-1}\left[\Delta_1\left(\frac{L^2}{\bar{c}_0} + \frac{L^3}{\bar{c}_0^2}\right) + L + \Delta_1^2\frac{L^4}{\bar{c}_0^3} + \frac{L^2}{\bar{c}_0^2} + \Delta_2\frac{L^3}{\bar{c}_0^2}\right] + O\left(N^{-2}\right)$$
$$\leq 2\nu N^{-1}(\bar{c}_0)^{-3}\left[\Delta_1^2 L^4 + \bar{c}_0\Delta_1 L^3 + \bar{c}_0^2 L^2 + \bar{c}_0^3 L\right] + O\left(N^{-2}\right),$$

where $\bar{c}_0 = \min(1, c_0) \leq 1$ and, from Section 3.3, $c_0 = bL$ where L is the expected document length. The second term $\bar{c}_0\Delta_1 L^3$ cannot be dominant as the system $\bar{c}_0\Delta_1 L^3 > \bar{c}_0^2 L^2$ and $\bar{c}_0\Delta_1 L^3 > \Delta_1^2 L^4$ is infeasible. Also, with the reasonable assumption that $L \geq 1$, we also have that the 4th term $\bar{c}_0^3 L \leq \bar{c}_0^2 L^2$. Therefore,

$$\mathbb{E}\left[\|\widehat{\mathbf{S}} - \mathbf{S}\|_F^2\right] \leq 3\nu N^{-1}\max\left[\Delta_1^2 L^4,\, \bar{c}_0^2 L^2\right] + O\left(N^{-2}\right).$$

B.2.4 Auxiliary Expressions

As $\{x_m\}_{m=1}^M$ are conditionally independent given y in the DICA model (3.4), we have the following expressions by using the law of total expectation for $m \neq m'$ and using the moments of the Poisson distribution with parameter y_m :

$$\begin{aligned}
\mathbb{E}(x_m) &= \mathbb{E}[\mathbb{E}(x_m|y_m)] = \mathbb{E}(y_m),\\
\mathbb{E}(x_m^2) &= \mathbb{E}[\mathbb{E}(x_m^2|y_m)] = \mathbb{E}(y_m^2) + \mathbb{E}(y_m),\\
\mathbb{E}(x_m^3) &= \mathbb{E}[\mathbb{E}(x_m^3|y_m)] = \mathbb{E}(y_m^3) + 3\mathbb{E}(y_m^2) + \mathbb{E}(y_m),\\
\mathbb{E}(x_m^4) &= \mathbb{E}[\mathbb{E}(x_m^4|y_m)] = \mathbb{E}(y_m^4) + 6\mathbb{E}(y_m^3) + 7\mathbb{E}(y_m^2) + \mathbb{E}(y_m),
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}(x_m x_{m'}) &= \mathbb{E}[\mathbb{E}(x_m x_{m'}|\mathbf{y})] = \mathbb{E}[\mathbb{E}(x_m|y_m)\mathbb{E}(x_{m'}|y_{m'})] = \mathbb{E}(y_m y_{m'}),\\
\mathbb{E}(x_m x_{m'}^2) &= \mathbb{E}[\mathbb{E}(x_m x_{m'}^2|\mathbf{y})] = \mathbb{E}[\mathbb{E}(x_m|y_m)\mathbb{E}(x_{m'}^2|y_{m'})] = \mathbb{E}(y_m y_{m'}^2) + \mathbb{E}(y_m y_{m'}),\\
\mathbb{E}(x_m^2 x_{m'}^2) &= \mathbb{E}[\mathbb{E}(x_m^2|y_m)\mathbb{E}(x_{m'}^2|y_{m'})] = \mathbb{E}(y_m^2 y_{m'}^2) + \mathbb{E}(y_m^2 y_{m'}) + \mathbb{E}(y_m y_{m'}^2) + \mathbb{E}(y_m y_{m'}).
\end{aligned}$$

Moreover, the moments of $\alpha_k \sim \text{Gamma}(c_k, b)$ are

$$\mathbb{E}(\alpha_k) = \frac{c_k}{b}, \quad \mathbb{E}(\alpha_k^2) = \frac{c_k^2 + c_k}{b^2}, \quad \mathbb{E}(\alpha_k^3) = \frac{c_k^3 + 3c_k^2 + 2c_k}{b^3},$$

$$\mathbb{E}(\alpha_k^4) = \frac{c_k^4 + 6c_k^3 + 11c_k^2 + 6c_k}{b^4}, \quad \text{etc.}$$

Annexe C

Implementation

Appendix C is organized as follows :

- In Appendix C.1, we derive expressions for scalable and memory-efficient implementation of the LDA moments (see Section 2.2.3) and DICA cumulants (see Section 3.3.2).
- In Appendix C.2, we derive expressions for scalable and memory-efficient implementation of the DCCA cumulants (see Section 4.4.1).

C.1 Implementation of Finite Sample Estimators

Since both population statistics $\boldsymbol{\mathcal{T}}^{LDA}$ and $\boldsymbol{\mathcal{T}}^{DICA}$ are tensors in the CP form with $\mathbf{D}$ being the CP factors, the same algorithms are used for the estimation of the topic matrix $\mathbf{D}$ from finite sample estimators of these tensors. Clearly, working with tensors directly is prohibitive due to the computation and storage issues. Most if not all algorithms, however, do not require construction of these tensors. Instead, the matrices of the form $\widehat{\mathbf{W}}\,\widehat{\boldsymbol{\mathcal{T}}}(\mathbf{v})\widehat{\mathbf{W}}^{\top}$ are of interest. In practice, these matrices can be computed from a finite sample directly (avoiding the construction of tensors) in $O(M_s NK) + O(NK^2) + O(NK)$ flops, where M_s is the largest number of non-zero counts (unique words) in a document over the corpus : $M_s = \max_{n=1,2,\ldots,N} \|\mathbf{x}_n\|_0$, where $\|\cdot\|_0$ counts the number of non-zero elements of a vector.

In this appendix, we derive the formulas for computation of $\widehat{\mathbf{W}}\,\widehat{\boldsymbol{\mathcal{T}}}(\mathbf{v})\widehat{\mathbf{W}}^{\top}$, for both LDA and DICA tensors, in the claimed number of flops. The derivations are straightforward, but quite tedious.

C.1.1 Expressions for Fast Implementation of the LDA Moments Finite Sample Estimators

Finite Sample Estimators for the First Three LDA Moments. The unbiased estimators $\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}$, $\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)}$, and $\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(3)}$ were defined in Equations (2.47)-(2.49). We rewrite

these expressions as :

$$\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)} = N^{-1}\sum_{n=1}^{N}\delta_{1n}\mathbf{x}_n = N^{-1}\mathbf{X}\boldsymbol{\delta}_1, \tag{C.1}$$

$$\begin{aligned}\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)} &= N^{-1}\sum_{n=1}^{N}\delta_{2n}\left[\mathbf{x}_n\otimes\mathbf{x}_n - \sum_{\ell=1}^{L_n}\mathbf{w}_{n\ell}\otimes\mathbf{w}_{n\ell}\right] = N^{-1}\sum_{n=1}^{N}\delta_{2n}\left(\mathbf{x}_n\otimes\mathbf{x}_n - \mathrm{Diag}(\mathbf{x}_n)\right) \\ &= N^{-1}\left[\mathbf{X}\,\mathrm{Diag}(\boldsymbol{\delta}_2)\mathbf{X}^\top - \mathrm{Diag}(\mathbf{X}\boldsymbol{\delta}_2)\right],\end{aligned} \tag{C.2}$$

$$\begin{aligned}\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(3)} &= N^{-1}\sum_{n=1}^{N}\delta_{3n}\left[\mathbf{x}_n\otimes\mathbf{x}_n\otimes\mathbf{x}_n - \sum_{\ell=1}^{L_n}\mathbf{w}_{n\ell}\otimes\mathbf{w}_{n\ell}\otimes\mathbf{w}_{n\ell}\right. \\ &\quad\left. - \sum_{\ell_1=1}^{L_n}\sum_{\substack{\ell_2=1\\ \ell_2\neq\ell_1}}^{L_n}\left(\mathbf{w}_{n\ell_1}\otimes\mathbf{w}_{n\ell_1}\otimes\mathbf{w}_{n\ell_2} + \mathbf{w}_{n\ell_1}\otimes\mathbf{w}_{n\ell_2}\otimes\mathbf{w}_{n\ell_1} + \mathbf{w}_{n\ell_1}\otimes\mathbf{w}_{n\ell_2}\otimes\mathbf{w}_{n\ell_2}\right)\right] \\ &= N^{-1}\sum_{n=1}^{N}\delta_{3n}\left[\mathbf{x}_n\otimes\mathbf{x}_n\otimes\mathbf{x}_n + 2\sum_{m=1}^{M}x_{nm}(\mathbf{e}_m\otimes\mathbf{e}_m\otimes\mathbf{e}_m)\right. \\ &\quad\left. - \sum_{m_1=1}^{M}\sum_{m_2=1}^{M}x_{nm_1}x_{nm_2}(\mathbf{e}_{m_1}\otimes\mathbf{e}_{m_1}\otimes\mathbf{e}_{m_2} + \mathbf{e}_{m_1}\otimes\mathbf{e}_{m_2}\otimes\mathbf{e}_{m_1} + \mathbf{e}_{m_1}\otimes\mathbf{e}_{m_2}\otimes\mathbf{e}_{m_2})\right].\end{aligned} \tag{C.3}$$

Finite Sample Estimators for the Matrix $\widehat{\mathbf{W}}\widehat{\boldsymbol{\mathcal{T}}}^{LDA}(\mathbf{v})\widehat{\mathbf{W}}^\top$. A finite sample estimate of $\widehat{\boldsymbol{\mathcal{T}}}^{LDA}$ was defined in Equation (2.46). The contraction with (a.k.a. projection onto) some vector $\mathbf{v}\in\mathbb{R}^M$ of this $M\times M\times M$-tensor can be written element-wise as :

$$\begin{aligned}\left[\widehat{\boldsymbol{\mathcal{T}}}^{LDA}(\mathbf{v})\right]_{m_1m_2} &= \sum_{m_3=1}^{M}\left[\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(3)}\right]_{m_1m_2m_3}v_{m_3} + C_2\sum_{m_3=1}^{M}[\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}]_{m_1}[\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}]_{m_2}[\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}]_{m_3}v_{m_3} \\ &- C_1\sum_{m_3=1}^{M}\left[\widehat{\mathbb{E}}(\mathbf{w}_{\ell_1}\otimes\mathbf{w}_{\ell_2}\otimes\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}) + \widehat{\mathbb{E}}(\mathbf{w}_{\ell_1}\otimes\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}\otimes\mathbf{w}_{\ell_3}) + \widehat{\mathbb{E}}(\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}\otimes\mathbf{w}_{\ell_2}\otimes\mathbf{w}_{\ell_3})\right]_{m_1m_2m_3}v_{m_3},\end{aligned}$$

where $C_1 = c_0(c_0+2)^{-1}$ and $C_2 = 2c_0^2\left[(c_0+1)(c_0+2)\right]^{-1}$. Plugging into this expres-

sion the expression (C.3) for the estimate $\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(3)}$, we get

$$\begin{aligned}
\left[\widehat{\boldsymbol{\mathcal{T}}}^{LDA}(\mathbf{v})\right]_{m_1m_2} &= N^{-1}\sum_{n=1}^{N}\delta_{3n}\left[x_{nm_1}x_{nm_2}\left\langle\mathbf{x}_n,\mathbf{v}\right\rangle + 2\sum_{m_3}\delta(m_1,m_2,m_3)x_{nm_3}v_{m_3}\right] \\
&- N^{-1}\sum_{n=1}^{N}\delta_{3n}\sum_{m_3=1}^{M}\left[\sum_{i,j=1}^{M}x_{ni}x_{nj}\left(\mathbf{e}_i\otimes\mathbf{e}_i\otimes\mathbf{e}_j+\mathbf{e}_i\otimes\mathbf{e}_j\otimes\mathbf{e}_i+\mathbf{e}_i\otimes\mathbf{e}_j\otimes\mathbf{e}_j\right)\right]_{m_1m_2m_3} v_{m_3} \\
&+ C_2\,[\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}]_{m_1}[\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}]_{m_2}\langle\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)},\mathbf{v}\rangle \\
&- C_1\left[[\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)}]_{m_1m_2}\langle\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)},\mathbf{v}\rangle + \sum_{m_3=1}^{M}\left([\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)}]_{m_1m_3}[\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}]_{m_2}v_{m_3} + [\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)}]_{m_2m_3}[\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}]_{m_1}v_{m_3}\right)\right],
\end{aligned}$$

where $\mathbf{e}_1$, $\mathbf{e}_2$, …, $\mathbf{e}_M$ denote the canonical basis of $\mathbb{R}^M$ (i.e., the columns of the $M\times M$ identity matrix $\mathbf{I}$). This further gives :

$$\begin{aligned}
&\left[\widehat{\mathbf{W}}\,\widehat{\boldsymbol{\mathcal{T}}}^{LDA}(\mathbf{v})\widehat{\mathbf{W}}^\top\right]_{k_1k_2} \\
&= N^{-1}\sum_{n=1}^{N}\delta_{3n}\left[\langle\mathbf{x}_n,\mathbf{v}\rangle\langle\mathbf{x}_n,\widehat{\mathbf{W}}_{k_1}\rangle\langle\mathbf{x}_n,\widehat{\mathbf{W}}_{k_2}\rangle + 2\sum_{m=1}^{M}x_{nm}v_m\widehat{W}_{k_1m}\widehat{W}_{k_2m}\right] \\
&- N^{-1}\sum_{n=1}^{N}\delta_{3n}\sum_{i,j=1}^{M}x_{ni}x_{nj}\left(\widehat{W}_{k_1i}\widehat{W}_{k_2i}v_j + \widehat{W}_{k_1i}\widehat{W}_{k_2j}v_i + \widehat{W}_{k_1i}\widehat{W}_{k_2j}v_j\right) \\
&- C_1\left[\langle\widehat{\mathbf{W}}_{k_1},\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)}\widehat{\mathbf{W}}_{k_2}\rangle + \langle\widehat{\mathbf{W}}_{k_1},\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)}\mathbf{v}\rangle\langle\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)}\widehat{\mathbf{W}}_{k_2}\rangle + \langle\widehat{\mathbf{W}}_{k_2},\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)}\mathbf{v}\rangle\langle\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)},\widehat{\mathbf{W}}_{k_1}\rangle\right] \\
&+ C_2\langle\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)},\widehat{\mathbf{W}}_{k_1}\rangle\langle\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)},\widehat{\mathbf{W}}_{k_2}\rangle\langle\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)},\mathbf{v}\rangle,
\end{aligned}$$

where $\widehat{\mathbf{W}}_k$ denotes the k-th row of $\widehat{\mathbf{W}}$ as a column vector. Introducing the counts matrix $\mathbf{X}\in\mathbb{R}^{M\times N}$ where each element X_{mn} is the count of the m-th word in the n-th document, this further simplifies to :

$$\begin{aligned}
\widehat{\mathbf{W}}\,\widehat{\boldsymbol{\mathcal{T}}}^{LDA}(\mathbf{v})\widehat{\mathbf{W}}^\top &= N^{-1}(\widehat{\mathbf{W}}\mathbf{X})\,\mathrm{Diag}\left[(\mathbf{X}^\top\mathbf{v})\circ\boldsymbol{\delta}_3\right](\widehat{\mathbf{W}}\mathbf{X})^\top \\
&+ N^{-1}\widehat{\mathbf{W}}\,\mathrm{Diag}\left[2[(\mathbf{X}\boldsymbol{\delta}_3)\circ\mathbf{v}] - \mathbf{X}[(\mathbf{X}^\top\mathbf{v})\circ\boldsymbol{\delta}_3]\right]\widehat{\mathbf{W}}^\top \\
&- N^{-1}(\widehat{\mathbf{W}}\mathrm{Diag}[\mathbf{v}]\mathbf{X})\,\mathrm{Diag}[\boldsymbol{\delta}_3](\widehat{\mathbf{W}}\mathbf{X})^\top - N^{-1}(\widehat{\mathbf{W}}\mathbf{X})\mathrm{Diag}[\boldsymbol{\delta}_3](\widehat{\mathbf{W}}\mathrm{Diag}[\mathbf{v}]\mathbf{X})^\top \\
&- C_1\left[\langle\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)},\mathbf{v}\rangle(\widehat{\mathbf{W}}\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)}\widehat{\mathbf{W}}^\top) + (\widehat{\mathbf{W}}(\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)}\mathbf{v}))(\widehat{\mathbf{W}}\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)})^\top + (\widehat{\mathbf{W}}\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)})(\widehat{\mathbf{W}}(\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)}\mathbf{v}))^\top\right] \\
&+ C_2\langle\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)},\mathbf{v}\rangle(\widehat{\mathbf{W}}\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)})(\widehat{\mathbf{W}}\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)})^\top.
\end{aligned}$$

Rewriting the last expression in a more compact form, we get :

$$\begin{aligned}
\widehat{\mathbf{W}}\widehat{\boldsymbol{\mathcal{T}}}^{LDA}(\mathbf{v})\widehat{\mathbf{W}}^\top &= N^{-1}\left[\mathbf{T}_1+\mathbf{T}_2-\mathbf{T}_3-\mathbf{T}_3^\top\right] + C_2\,\langle\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)},\mathbf{v}\rangle(\widehat{\mathbf{W}}\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)})(\widehat{\mathbf{W}}\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)})^\top \\
&- C_1\left[\langle\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)},\mathbf{v}\rangle(\widehat{\mathbf{W}}\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)}\widehat{\mathbf{W}}^\top) + \mathbf{T}_4 + \mathbf{T}_4^\top\right], \qquad \text{(C.4)}
\end{aligned}$$

where $C_1 = c_0(c_0+2)^{-1}$, $C_2 = 2c_0^2\left[(c_0+1)(c_0+2)\right]^{-1}$ and

$$\begin{aligned}
\mathbf{T}_1 &= (\widehat{\mathbf{W}}\mathbf{X})\mathrm{Diag}\left[(\mathbf{X}^\top\mathbf{v})\circ\boldsymbol{\delta}_3\right](\widehat{\mathbf{W}}\mathbf{X})^\top,\\
\mathbf{T}_2 &= \widehat{\mathbf{W}}\mathrm{Diag}\left[2[(\mathbf{X}\boldsymbol{\delta}_3)\circ\mathbf{v}] - \mathbf{X}[(\mathbf{X}^\top\mathbf{v})\circ\boldsymbol{\delta}_3]\right]\widehat{\mathbf{W}}^\top,\\
\mathbf{T}_3 &= [\widehat{\mathbf{W}}\mathrm{Diag}(\mathbf{v})\mathbf{X}]\mathrm{Diag}(\boldsymbol{\delta}_3)(\widehat{\mathbf{W}}\mathbf{X})^\top,\\
\mathbf{T}_4 &= [\widehat{\mathbf{W}}(\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(2)}\mathbf{v})](\widehat{\mathbf{W}}\widehat{\boldsymbol{\mu}}_{\mathbf{x}}^{(1)})^\top.
\end{aligned}$$

C.1.2 Expressions for Fast Implementation of the DICA Cumulants Finite Sample Estimators

The finite sample estimator $\widehat{\boldsymbol{\mathcal{T}}}^{DICA}$ was defined in Equation (3.20). The contraction with (a.k.a. projection onto) some vector $\mathbf{v}\in\mathbb{R}^M$ of this $M\times M\times M$-tensor can be written element-wise as :

$$\begin{aligned}
\left[\widehat{\boldsymbol{\mathcal{T}}}^{DICA}(\mathbf{v})\right]_{m_1m_2} &= \sum_{m_3=1}^{M}\widehat{\mathrm{cum}}(x_{m_1},x_{m_2},x_{m_3})v_{m_3} + 2\sum_{m_3=1}^{M}\delta(m_1,m_2,m_3)\widehat{\mathbb{E}}(x_{m_3})v_{m_3}\\
&- \sum_{m_3=1}^{M}\delta(m_2,m_3)\widehat{\mathrm{cov}}(x_{m_1},x_{m_2})v_{m_3}\\
&- \sum_{m_3=1}^{M}\delta(m_1,m_3)\widehat{\mathrm{cov}}(x_{m_1},x_{m_2})v_{m_3}\\
&- \sum_{m_3=1}^{M}\delta(m_1,m_2)\widehat{\mathrm{cov}}(x_{m_1},x_{m_3})v_{m_3}\\
&= \sum_{m_3=1}^{M}\widehat{\mathrm{cum}}(x_{m_1},x_{m_2},x_{m_3})v_{m_3} + 2\delta(m_1,m_2)\widehat{\mathbb{E}}(x_{m_1})v_{m_1}\\
&- \widehat{\mathrm{cov}}(x_{m_1},x_{m_2})v_{m_2} - \widehat{\mathrm{cov}}(x_{m_1},x_{m_2})v_{m_1} - \delta(m_1,m_2)\sum_{m_3=1}^{M}\widehat{\mathrm{cov}}(x_{m_1},x_{m_3})v_{m_3}.
\end{aligned}$$

This further gives :

$$\begin{aligned}
\left[\widehat{\mathbf{W}}\widehat{\boldsymbol{\mathcal{T}}}^{DICA}(\mathbf{v})\widehat{\mathbf{W}}^{\top}\right]_{k_1k_2} &= \widehat{\mathbf{W}}_{k_1}^{\top}\widehat{\boldsymbol{\mathcal{T}}}^{DICA}(\mathbf{v})\widehat{\mathbf{W}}_{k_2} \\
&= \sum_{m_1=1}^{M}\sum_{m_2=1}^{M}\sum_{m_3=1}^{M}\widehat{\mathrm{cum}}(x_{m_1},x_{m_2},x_{m_3})v_{m_3}\widehat{W}_{k_1m_1}\widehat{W}_{k_2m_2} \\
&+ 2\sum_{m_1=1}^{M}\sum_{m_2=1}^{M}\delta(m_1,m_2)\widehat{\mathbb{E}}(x_{m_1})v_{m_1}\widehat{W}_{k_1m_1}\widehat{W}_{k_2m_2} \\
&\overset{(a)}{-}\sum_{m_1=1}^{M}\sum_{m_2=1}^{M}\widehat{\mathrm{cov}}(x_{m_1},x_{m_2})v_{m_2}\widehat{W}_{k_1m_1}\widehat{W}_{k_2m_2} \\
&\overset{(b)}{-}\sum_{m_1=1}^{M}\sum_{m_2=1}^{M}\widehat{\mathrm{cov}}(x_{m_1},x_{m_2})v_{m_1}\widehat{W}_{k_1m_1}\widehat{W}_{k_2m_2} \\
&- \sum_{m_1=1}^{M}\sum_{m_3=1}^{M}\widehat{\mathrm{cov}}(x_{m_1},x_{m_3})v_{m_3}\widehat{W}_{k_1m_1}\widehat{W}_{k_2m_1},
\end{aligned}$$

where $\widehat{\mathbf{W}}_k$ denotes the k-th row of $\widehat{\mathbf{W}}$ as a column vector. Note that the expressions (a) and (b) in the 4-th and 5-th line are not equal, due to the presence of k_1 and k_2 indices. By further plugging in the expressions (3.21) for the unbiased finite sample estimates of $\widehat{\mathrm{cov}}$ and $\widehat{\mathrm{cum}}$, we get :

$$\begin{aligned}
\left[\widehat{\mathbf{W}}\widehat{\boldsymbol{\mathcal{T}}}^{DICA}(\mathbf{v})\widehat{\mathbf{W}}^{\top}\right]_{k_1k_2} &= A_1\sum_{n=1}^{N}\langle\widehat{\mathbf{W}}_{k_1},\mathbf{x}_n-\widehat{\mathbb{E}}(\mathbf{x})\rangle\langle\widehat{\mathbf{W}}_{k_2},\mathbf{x}_n-\widehat{\mathbb{E}}(\mathbf{x})\rangle\langle\mathbf{v},\mathbf{x}_n-\widehat{\mathbb{E}}(\mathbf{x})\rangle \\
&+ 2\sum_{m=1}^{M}\widehat{\mathbb{E}}(x_m)v_m\widehat{W}_{k_1m}\widehat{W}_{k_2m} \\
&- A_2\sum_{n=1}^{N}\langle\widehat{\mathbf{W}}_{k_1},\mathbf{x}_n-\widehat{\mathbb{E}}(\mathbf{x})\rangle\langle\mathbf{v}\circ\widehat{\mathbf{W}}_{k_2},\mathbf{x}_n-\widehat{\mathbb{E}}(\mathbf{x})\rangle \\
&- A_2\sum_{n=1}^{N}\langle\mathbf{v}\circ\widehat{\mathbf{W}}_{k_1},\mathbf{x}_n-\widehat{\mathbb{E}}(\mathbf{x})\rangle\langle\widehat{\mathbf{W}}_{k_2},\mathbf{x}_n-\widehat{\mathbb{E}}(\mathbf{x})\rangle \\
&- A_2\sum_{n=1}^{N}\langle\widehat{\mathbf{W}}_{k_1}\circ\widehat{\mathbf{W}}_{k_2},\mathbf{x}_n-\widehat{\mathbb{E}}(\mathbf{x})\rangle\langle\mathbf{v},\mathbf{x}_n-\widehat{\mathbb{E}}(\mathbf{x})\rangle,
\end{aligned}$$

where $A_1 = N\left[(N-1)(N-2)\right]^{-1}$, $A_2 = (N-1)^{-1}$, and $\circ$ denotes the element-wise Hadamard product. Introducing the counts matrix $\mathbf{X} \in \mathbb{R}^{M\times N}$ where each element X_{mn} is the count of the m-th word in the n-th document, this further simplifies

to :

$$
\begin{aligned}
&\widehat{\mathbf{W}}\widehat{\boldsymbol{\mathcal{T}}}^{DICA}(\mathbf{v})\widehat{\mathbf{W}}^{\top} = A_1\,(\widehat{\mathbf{W}}\mathbf{X})\mathrm{Diag}[\mathbf{X}^{\top}\mathbf{v}](\widehat{\mathbf{W}}\mathbf{X})^{\top} \\
&+ A_1\,\langle \mathbf{v}, \widehat{\mathbb{E}}(\mathbf{x})\rangle \left[2N(\widehat{\mathbf{W}}\widehat{\mathbb{E}}(\mathbf{x}))(\widehat{\mathbf{W}}\widehat{\mathbb{E}}(\mathbf{x}))^{\top} - (\widehat{\mathbf{W}}\mathbf{X})(\widehat{\mathbf{W}}\mathbf{X})^{\top}\right] \\
&- A_1 \left[\widehat{\mathbf{W}}\mathbf{X}(\mathbf{X}^{\top}\mathbf{v})(\widehat{\mathbf{W}}\widehat{\mathbb{E}}(\mathbf{x}))^{\top} + \widehat{\mathbf{W}}\widehat{\mathbb{E}}(\mathbf{x})(\widehat{\mathbf{W}}\mathbf{X}(\mathbf{X}^{\top}\mathbf{v}))^{\top}\right] \\
&+ 2\widehat{\mathbf{W}}\mathrm{Diag}[\mathbf{v}\circ\widehat{\mathbb{E}}(\mathbf{x})]\widehat{\mathbf{W}}^{\top} \\
&- A_2 \left[(\widehat{\mathbf{W}}\mathbf{X})(\widehat{\mathbf{W}}\mathrm{Diag}(\mathbf{v})\mathbf{X})^{\top} + (\widehat{\mathbf{W}}\mathrm{Diag}(\mathbf{v})\mathbf{X})(\widehat{\mathbf{W}}\mathbf{X})^{\top} + \widehat{\mathbf{W}}\mathrm{Diag}[\mathbf{X}(\mathbf{X}^{\top}\mathbf{v})]\widehat{\mathbf{W}}^{\top}\right] \\
&+ A_2 \left[(\widehat{\mathbf{W}}\widehat{\mathbb{E}}(\mathbf{x}))(\widehat{\mathbf{W}}\mathrm{Diag}[\mathbf{v}]\widehat{\mathbb{E}}(\mathbf{x}))^{\top} + (\widehat{\mathbf{W}}\mathrm{Diag}[\mathbf{v}]\widehat{\mathbb{E}}(\mathbf{x}))(\widehat{\mathbf{W}}\widehat{\mathbb{E}}(\mathbf{x}))^{\top}\right] \\
&+ A_2\langle \mathbf{v}, \widehat{\mathbb{E}}(\mathbf{x})\rangle\widehat{\mathbf{W}}\mathrm{Diag}[\widehat{\mathbb{E}}(\mathbf{x})]\widehat{\mathbf{W}}^{\top}.
\end{aligned}
$$

Rewriting the last expression in a more compact form, we get :

$$
\begin{aligned}
\widehat{\mathbf{W}}\widehat{\boldsymbol{\mathcal{T}}}^{DICA}(\mathbf{v})\widehat{\mathbf{W}}^{\top} = &\frac{N}{(N-1)(N-2)}\left[\mathbf{T}_1 + \langle \mathbf{v}, \widehat{\mathbb{E}}(\mathbf{x})\rangle\,(\mathbf{T}_2 - \mathbf{T}_3) - (\mathbf{T}_4 + \mathbf{T}_4^{\top})\right] \\
&+ \frac{1}{N-1}\left[\mathbf{T}_5 + \mathbf{T}_5^{\top} - \mathbf{T}_6 - \mathbf{T}_6^{\top} + \widehat{\mathbf{W}}\mathrm{Diag}(\mathbf{t})\widehat{\mathbf{W}}^{\top}\right],
\end{aligned}
\tag{C.5}
$$

where

$$
\begin{aligned}
\mathbf{T}_1 &= (\widehat{\mathbf{W}}\mathbf{X})\mathrm{Diag}[\mathbf{X}^{\top}\mathbf{v}](\widehat{\mathbf{W}}\mathbf{X})^{\top}, \\
\mathbf{T}_2 &= 2N(\widehat{\mathbf{W}}\widehat{\mathbb{E}}(\mathbf{x}))(\widehat{\mathbf{W}}\widehat{\mathbb{E}}(\mathbf{x}))^{\top}, \\
\mathbf{T}_3 &= (\widehat{\mathbf{W}}\mathbf{X})(\widehat{\mathbf{W}}\mathbf{X})^{\top}, \\
\mathbf{T}_4 &= \widehat{\mathbf{W}}\mathbf{X}(\mathbf{X}^{\top}\mathbf{v})(\widehat{\mathbf{W}}\widehat{\mathbb{E}}(\mathbf{x}))^{\top}, \\
\mathbf{T}_5 &= (\widehat{\mathbf{W}}\mathbf{X})(\widehat{\mathbf{W}}\mathrm{Diag}(\mathbf{v})\mathbf{X})^{\top}, \\
\mathbf{T}_6 &= (\widehat{\mathbf{W}}\mathrm{Diag}(\mathbf{v})\widehat{\mathbb{E}}(\mathbf{x}))(\widehat{\mathbf{W}}\widehat{\mathbb{E}}(\mathbf{x}))^{\top}, \\
\mathbf{t} &= 2(N-1)[\mathbf{v}\circ\widehat{\mathbb{E}}(\mathbf{x})] + \langle \mathbf{v}, \widehat{\mathbb{E}}(\mathbf{x})\rangle\widehat{\mathbb{E}}(\mathbf{x}) - \mathbf{X}(\mathbf{X}^{\top}\mathbf{v}).
\end{aligned}
$$

C.2 Multi-View Models

C.2.1 Finite Sample Estimators of the DCCA Cumulants

In this section, we sketch the derivation of unbiased finite sample estimators for the CCA cumulants $\mathbf{S}_{12}$, $\boldsymbol{\mathcal{T}}_{121}$, and $\boldsymbol{\mathcal{T}}_{122}$. Since the derivation is nearly identical to the derivation of the estimators for the DICA cumulants (see Appendix F.2 of Podosinnikova et al. [2015]), all details are omitted.

Given a finite sample $\mathbf{X}^{(1)} = \left\{\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}, \ldots, \mathbf{x}_N^{(1)}\right\}$ and $\mathbf{X}^{(2)} = \left\{\mathbf{x}_1^{(2)}, \mathbf{x}_2^{(2)}, \ldots, \mathbf{x}_N^{(2)}\right\}$, the finite sample estimator of the discrete CCA $\mathbf{S}$-covariance (4.24), i.e., $\mathbf{S}_{12} :=$

$\text{cum}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$, takes the form

$$\widehat{\mathbf{S}}_{12} = \eta_1 \left[\mathbf{X}^{(1)} \mathbf{X}^{(2)\top} - N \widehat{\mathbb{E}}(\mathbf{x}^{(1)}) \widehat{\mathbb{E}}(\mathbf{x}^{(2)})^\top \right], \tag{C.6}$$

where $\widehat{\mathbb{E}}(\mathbf{x}^{(1)}) = N^{-1} \sum_{n=1}^{N} x_n^{(1)}$, $\widehat{\mathbb{E}}(\mathbf{x}^{(2)}) = N^{-1} \sum_{n=1}^{N} x_n^{(2)}$, and $\eta_1 = 1/(N-1)$.

Substitution of the finite sample estimators of the 2nd and 3rd cumulants (see, e.g., Appendix C.4 of Podosinnikova et al. [2015]) into the definition of the DCCA $\boldsymbol{\mathcal{T}}$-cumulants (4.26) leads to the following expressions

$$\begin{aligned}
\widehat{\mathbf{W}}_1 \widehat{\boldsymbol{\mathcal{T}}}_{12j}(\mathbf{v}_j) \widehat{\mathbf{W}}_2^\top &= \eta_2 [(\widehat{\mathbf{W}}_1 \mathbf{X}^{(1)}) \text{Diag}(\mathbf{X}^{(j)\top} \mathbf{v}_j)] \otimes (\widehat{\mathbf{W}}_2 \mathbf{X}^{(2)}) \\
&+ \eta_2 \langle \mathbf{v}_j, \widehat{\mathbb{E}}(\mathbf{x}^{(j)}) \rangle 2N [\widehat{\mathbf{W}}_1 \widehat{\mathbb{E}}(\mathbf{x}^{(1)})] \otimes [\widehat{\mathbf{W}}_2 \widehat{\mathbb{E}}(\mathbf{x}^{(2)})] \\
&- \eta_2 \langle \mathbf{v}_j, \widehat{\mathbb{E}}(\mathbf{x}^{(j)}) \rangle (\widehat{\mathbf{W}}_1 \mathbf{X}^{(1)}) \otimes (\widehat{\mathbf{W}}_2 \mathbf{X}^{(2)}) \\
&- \eta_2 [(\widehat{\mathbf{W}}_1 \mathbf{X}^{(1)}) (\mathbf{X}^{(j)\top} \mathbf{v}_j)] \otimes [\widehat{\mathbf{W}}_2 \widehat{\mathbb{E}}(\mathbf{x}^{(2)})] \\
&- \eta_2 [\widehat{\mathbf{W}}_1 \widehat{\mathbb{E}}(\mathbf{x}^{(1)})] \otimes [(\widehat{\mathbf{W}}_2 \mathbf{X}^{(2)}) (\mathbf{X}^{(j)\top} \mathbf{v}_j)] \\
&- \eta_1 \widehat{\mathbf{W}}_1^{(j)} \mathbf{X}^{(1)}) \otimes (\widehat{\mathbf{W}}_2^{(j)} \mathbf{X}^{(2)}) \\
&+ \eta_1 N [\widehat{\mathbf{W}}_1^{(j)} \widehat{\mathbb{E}}(\mathbf{x}^{(1)})] \otimes [\widehat{\mathbf{W}}_2^{(j)} \widehat{\mathbb{E}}(\mathbf{x}^{(2)})],
\end{aligned}$$

where $\eta_2 = N/((N-1)(N-2))$ and $\widehat{\mathbf{W}}_1^{(1)} = \widehat{\mathbf{W}}_1 \text{Diag}(\mathbf{v}_1)$, $\widehat{\mathbf{W}}_2^{(1)} = \widehat{\mathbf{W}}_2$, $\widehat{\mathbf{W}}_1^{(2)} = \widehat{\mathbf{W}}_1$, and $\widehat{\mathbf{W}}_2^{(2)} = \widehat{\mathbf{W}}_2 \text{Diag}(\mathbf{v}_2)$.

In the expressions above, $\widehat{\mathbf{W}}_1$ and $\widehat{\mathbf{W}}_2$ denote whitening matrices of $\widehat{\mathbf{S}}_{12}$, i.e. such that $\widehat{\mathbf{W}}_1 \widehat{\mathbf{S}}_{12} \widehat{\mathbf{W}}_2^\top = \mathbf{I}$.

Bibliographie

B. Afsari. Simple LU and QR based non-orthogonal matrix joint diagonalization. In *Proc. ICA*, 2006.

B. Afsari. What can make joint diagonalization difficult ? In *Proc. ICASSP*, 2007.

B. Afsari. Sensitivity analysis for the problem of matrix joint diagonalization. *SIAM J. Matrix Anal. Appl.*, 30(2) :1148–1171, 2008.

S. Amari, A. Cichocki, and H.H. Yang. A new learning algorithm for blind signal separation. In *Adv. NIPS*, 1996.

A. Anandkumar, D.P. Foster, D. Hsu, S.M. Kakade, and Y.-K. Liu. A spectral algorithm for latent Dirichlet allocation. In *Adv. NIPS*, 2012a.

A. Anandkumar, D. Hsu, and S.M. Kakade. A method of moments for mixture models and hidden Markov models. In *Proc. COLT*, 2012b.

A. Anandkumar, D.P. Foster, D. Hsu, S.M. Kakade, and Y.-K. Liu. A spectral algorithm for latent Dirichlet allocation. Technical report, arXiv :1204.6703v4, 2013a.

A. Anandkumar, D. Hsu, A. Javanmard, and S.M. Kakade. Learning topic models and latent Bayesian networks under expansion constraints. Technical report, arXiv :1209.5350v3, 2013b.

A. Anandkumar, R. Ge, D. Hsu, S.M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *J. Mach. Learn. Res.*, 15 :2773–2832, 2014.

A. Anandkumar, D.P. Foster, D. Hsu, S.M. Kakade, and Y.-K. Liu. A spectral algorithm for latent Dirichlet allocation. *Algorithmica*, 72(1) :193–214, 2015a.

A. Anandkumar, R. Ge, and M. Janzamin. Analyzing tensor power method dynamics in overcomplete regime. Technical report, arXiv :1411.1488v2, 2015b.

A. Anandkumar, D. Hsu, M. Janzamin, and S.M. Kakade. When are overcomplete topic models identifiable ? Uniqueness of tensor Tucker decompositions with structured sparsity. *J. Mach. Learn. Res.*, 16 :2643–2694, 2015c.

F. Arabshahi and A. Anandkumar. Beyond LDA : A unified framework for learning latent normalized infinitely divisible topic models through spectral methods. Technical report, arXiv :1605.09080v4, 2016.

C. Archambeau and F. Bach. Sparse probabilistic projections. In *Adv. NIPS*, 2008.

S. Arora and R. Kannan. Learning mixtures of arbitrary Gaussians. In *Proc. STOC*, 2001.

S. Arora, R. Ge, and A. Moitra. Learning topic models – Going beyond SVD. In *Proc. FOCS*, 2012.

S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu. A practical algorithm for topic modeling with provable guarantees. In *Proc. ICML*, 2013.

S. Arora, R. Ge, T. Ma, and A. Moitra. Provable algorithms for inference in topic models. In *Proc. ICML*, 2015.

F. Bach and M.I. Jordan. Kernel independent component analysis. *J. Mach. Learn. Res.*, 3 :1–48, 2002.

F. Bach and M.I. Jordan. A Probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley, 2005.

R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

D.J. Bartholomew. *Latent Variable Models and Factor Analysis*. Wiley, 1987.

D.J. Bartholomew, M. Knott, and I. Moustaki. *Latent Variable Models and Factor Analysis : A Unified Approach*. Wiley, 3rd edition, 2011.

A. Basilevsky. *Statistical Factor Analysis and Related Methods : Theory and Applications*. Wiley, 1994.

G. Bergqvist. Exact probabilites for typical ranks of $2 \times 2 \times 2$ and $3 \times 3 \times 2$ tensors. *Linear Algebra Appl.*, 438(2) :663–667, 2013.

P.J. Bickel, C.A.J. Klaassen, Y. Ritov, and J.A. Wellner. *Efficient and Adaptive Estimation for Semiparametric Models*. Springer, 1998.

D. Bini, M. Capovani, G. Lotti, and F. Romani. $O\big(n^{2.7799}\big)$ approximate matrix multiplication. *Inform. Process. Lett.*, 8(5) :234–235, 1979.

D. Bini, G. Lotti, and F. Romani. Approximate solutions for the bilinear form computational problem. *SIAM J. Comput.*, 9(4) :692–697, 1980.

S. Bird, E. Loper, and E. Klei. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.

C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

D.M. Blei. Probabilistic topic models. *Comm. ACM*, 55(4) :77–84, 2012.

D.M. Blei and M.I. Jordan. Modeling annotated data. In *Proc. SIGIR*, 2003.

D.M. Blei and J.D. Lafferty. A correlated topic model of Science. *Ann. Appl. Stat.*, 1(1) :17–35, 2007.

D.M. Blei and J.D. Lafferty. Topic models. In A.N. Srivastava and M. Sahami, editors, *Text Mining : Classification, Clustering, and Applications*, chapter 4, pages 71–94. CRC Press, 2009.

D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3 :993–1022, 2003.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities : A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15 :1455–1459, 2014. URL http://www.manopt.org.

R. Bro and S. Jong. A fast non-negativity constrained least squares algorithm. *J. Chemometrics*, 11(5) :393–401, 1997.

R. Bro and N. Sidiropoulos. least squares algorithms under unimodality and non-negativity constraints. *J. Chemometrics*, 12(4) :223–247, 1998.

M.W. Browne. The maximum-likelihood solution in inter-battery factor analysis. *Br. J. Math. Stat. Psychol.*, 32(1) :75–86, 1979.

A. Bunse-Gerstner, R. Byers, and V. Mehrmann. Numerical methods for simultaneous diagonalization. *SIAM J. Matrix Anal. Appl.*, 14(4) :927–949, 1993.

W. Buntine and A. Jakulin. Discrete principal component analysis. Technical report, Helsinki Institute for Information Technology, 2005.

W. Buntine and A. Jakulin. Discrete component analysis. In *Proc. SLSFS*, 2006.

W.L. Buntine. Operations for learning with graphical models. *J. Artif. Intell. Res.*, 2 :159–225, 1994.

W.L. Buntine. Variational extensions to EM and multinomial PCA. In *Proc. ECML*, 2002.

W.L. Buntine and A. Jakulin. Applying discrete PCA in data analysis. In *Proc. UAI*, 2004.

J. Canny. GaP : a factor model for discrete data. In *Proc. SIGIR*, 2004.

J.-F. Cardoso. Source separation using higher order moments. In *Proc. ICASSP*, 1989.

J.-F. Cardoso. Eigen-structure of the fourth-order cumulant tensor with application to the blind source separation problem. In *Proc. ICASSP*, 1990.

J.-F. Cardoso. Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors. In *Proc. ICASSP*, 1991.

J.-F. Cardoso. Perturbation of joint diagonalizers. Technical report, Télécom Paris, 1994a.

J.-F. Cardoso. On the performance of orthogonal source separation algorithms. In *Proc. EUSIPCO*, 1994b.

J.-F. Cardoso. Multidimensional independent component analysis. In *Proc. ICASSP*, 1998.

J.-F. Cardoso. High-order contrasts for independent component analysis. *Neural Comput.*, 11 :157–192, 1999.

J.-F. Cardoso and A. Souloumiac. Blind beamforming for non Gaussian signals. In *IEE Proc.-F*, 1993.

J.-F. Cardoso and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM J. Mat. Anal. Appl.*, 17(1) :161–164, 1996.

J.D. Carroll and J.J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3) :283–319, 1970.

J.D. Carroll, G. De Soete, and S. Pruzansky. Fitting of the latent class model via iteratively reweighted least squares Candecomp with nonnegativity constraints. In R. Coppi, S. Bolasco, F. Critchley, and Y. Escoufier, editors, *Multiway Data Analysis*, pages 463–472. Elsevier Science, 1989.

A.T. Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Comput. Intell. Neurosci.*, 2009. Article ID 785152.

G. Chabriel, M. Kleinsteuber, E. Moreau, H. Shen, P. Tichavsky, and A. Yeredor. Joint matrices decompositions and blind source separation : A survey of methods, identification, and applications. *IEEE Signal Process. Mag.*, 31(3) :34–43, 2014.

S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1) :33–61, 1999.

P. Chevalier. Méthodes aveugles de filtrage d'antennes. *Revue d'Electronique et d'Electricité 3*, pages 48–58, 1995.

P.A. Chew, B.W. Bader, T.G. Kolda, and A. Abdelali. Cross-language information retrieval using Parafac2. In *Proc. KDD*, 2007.

A. Cichocki, R. Zdunek, A.H. Phan, and S. Amari. *Nonnegative Matrix and Tensor Factorizations : Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. Wiley, 2009.

S. Cohen and M. Collins. A provably correct learning algorithm for latent-variable PCFGs. In *ACL*, 2014.

N. Colombo and N. Vlassis. Tensor decomposition via joint matrix Schur decomposition. In *Proc. ICML*, 2016a.

N. Colombo and N. Vlassis. Approximate joint matrix triangularization. Technical report, arXiv :1607.00514v1, 2016b.

N. Colombo and N. Vlassis. A posteriori error bounds for joint matrix decomposition problems. In *Adv. NIPS*, 2016c.

P. Comon. Independent component analysis, A new concept ? *Signal Process.*, 36(3) : 287–314, 1994.

P. Comon. Blind channel identification and extraction of more sources than sensors. In *Proc. SPIE*, 1998.

P. Comon. Tensor decompositions : State of the art and applications. In J.G. McWhirter and I.K. Proudler, editors, *Mathematics in Signal Processing*, pages 1–24. Oxford University Press, 2002.

P. Comon. Blind identification and source separation in 2×3 under-determined mixtures. *IEEE Trans. Signal Process.*, 52(1) :11–22, 2004.

P. Comon. Tensors versus matrices usefulness and unexpected properties. In *Proc. IEEE SSP*, 2009.

P. Comon. Tensors : A brief introduction. *IEEE Signal Process. Mag.*, 31 :44–53, 2014.

P. Comon and C. Jutten. *Handbook of Blind Sourse Separation : Independent Component Analysis and Applications*. Academic Press, 2010.

P. Comon and J.M.F. Ten Berge. Generic and typical ranks of three-way arrays. In *Proc. ICASSP*, 2008.

P. Comon, G. Golub, L.-H. Lim, and B. Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM J. Matrix Anal. Appl.*, 30(3) :1254–1279, 2008.

P. Comon, X. Luciani, and A.L.F. De Almeida. Tensor decompositions, alternating least squares and other tales. *J. Chemometrics*, 23 :393–405, 2009a.

P. Comon, J.M.F. Ten Berge, L. De Lathauwer, and J. Castaing. Generic and typical ranks of multi-way arrays. *Linear Algebra Appl.*, 430(11) :2997–3007, 2009b.

S. Dasgupta. Learning mixtures of Gaussians. In *Proc. FOCS*, 1999.

A. d'Aspremont, F. Bach, and L. El Ghaoui. Optimal solutions for sparse principal component analysis. *J. Mach. Learn. Res.*, 9 :1269–1294, 2008.

A. d'Aspremont, F. Bach, and L. El Ghaoui. Approximation bounds for sparse principal component analysis. *Math. Prog.*, 148(1) :89–110, 2014.

L. De Lathauwer. A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization. *SIAM J. Matrix Anal. Appl.*, 28(3) : 642–666, 2006.

L. De Lathauwer. Algebraic methods after prewhitening. In P. Comon and C. Jutten, editors, *Handbook of Blind Sourse Separation : Independent Component Analysis and Applications*, chapter 5, pages 155–177. Academic Press, 2010.

L. De Lathauwer and J. Castaing. Blind identification of underdetermined mixtures by simultaneous matrix diagonalization. *IEEE Trans. Signal Process.*, 56(3) :1096–1105, 2008.

L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-(R_1, R_2, ..., R_N) approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.*, 21(4) :1324–1342, 2000.

L. De Lathauwer, B. De Moor, and J. Vandewalle. Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition. *SIAM J. Matrix Anal. Appl.*, 26(2) :295–327, 2004.

L. De Lathauwer, B. De Moor, and J. Vanderwalle. A prewhitening-induced bound on the identification error in independent component analysis. *IEEE Trans. Circuits Syst. I, Reg. Papers*, 52(3), 2005.

L. De Lathauwer, J. Castaing, and J.-F. Cardoso. Fourth-order cumulant-based blind identification of underdetermined mixtures. *IEEE Trans. Signal Process.*, 55 :2965–2973, 2007.

V. De Silva and L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM J. Matrix. Anal. Appl.*, 30(3) :1084–1127, 2008.

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. Assoc. Inf. Sci. Technol.*, 41(6) :391–407, 1990.

S. Dégerine and E. Kane. A comparative study of approximate joint diagonalization algorithms for blind source separation in presence of additive noise. *IEEE Trans. Signal Process.*, 55(6) :3022–3031, 2007.

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Series B*, 39(1) :1–38, 1977.

O. Dikmen and C. Févotte. Maximum marginal likelihood estimation for nonnegative dictionary learning in the gamma-Poisson model. *IEEE Trans. Signal Process.*, 60 (10) :5163–5175, 2012.

C. Ding, T. Li, and W. Peng. Nonnegative matrix factorization and probabilistic latent semantic indexing : Equivalence, chi-square statistics, and a hybrid method. In *Proc. AAAI*, 2006.

C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Comput. Stat. Data Anal.*, 52(8) :3913–3927, 2008.

I. Domanov and L. De Lathauwer. On the uniqueness of the canonical polyadic decomposition of third-order tensors—Part I : Basic results and uniqueness of one factor matrix. *SIAM J. Matrix Anal. Appl.*, 34(3) :855–875, 2013a.

I. Domanov and L. De Lathauwer. On the uniqueness of the canonical polyadic decomposition of third-order tensors—Part II : Uniqueness of the overall decomposition. *SIAM J. Matrix Anal. Appl.*, 34(3) :876–903, 2013b.

I. Domanov and L. De Lathauwer. Canonical polyadic decomposition of third-order tensors : Reduction to generalized eigenvalue decomposition. *SIAM J. Matrix Anal. Appl.*, 35(2) :636–660, 2014.

I. Domanov and L. De Lathauwer. Canonical polyadic decomposition of third-order tensors : Relaxed uniqueness conditions and algebraic algorithm. Technical report, arXiv :1501 :07251v2, 2016.

R. Durrett. *Probability : Theory and Examples*. Cambridge University Press, 4th edition, 2013.

P.J. Eberlein. A Jacobi-like method for the automatic computation of eigenvalues and eigenvectors of an arbitrary matrix. *J. Soc. Indust. Appl. Math.*, 10(1) :74–88, 1962.

G. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3) :211–218, 1936.

S. Favaro and I. Hadjicharalambous, G. Prünster. On a class of distributions on the simplex. *J. Stat. Plan. Inference*, 141(9) :2987–3004, 2011.

B.A. Frigyik, A. Kapila, and M.R. Gupta. Introduction to the Dirichlet distribution and related processes. Technical Report UWEETR-2010-0006, University of Washington, 2010.

T. Fu and X. Gao. Simultaneous diagonalization with similarity transformation for non-defective matrices. In *Proc. ICASSP*, 2006.

E. Gaussier and C. Goutte. Relation between pLSA and NMF and implications. In *Proc. SIGIR*, 2005.

R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points—Online stochastic gradient for tensor decomposition. Technical report, arXiv :1503.02101v1, 2015.

W.R. Gilks, S. Richardson, and D. Spieglhalter. *Markov Chain Monte Carlo in Practice*. CRC Press, 1995.

M. Girolami and A. Kabán. On an equivalence between PLSI and LDA. In *Proc. SIGIR*, 2003.

A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean embedding of co-occurrence data. *J. Mach. Learn. Res.*, 8 :2265–2295, 2007.

G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, 3rd edition, 1996.

G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, 4th edition, 2013.

A. Gordo. Supervised mid-level features for word image representation. In *Proc. CVPR*, 2015.

T. Griffiths. Gibbs sampling in the generative model of latent Dirichlet allocation. Technical report, Stanford University, 2002.

A. Haghighi, P. Liang, T.B. Kirkpatrick, and D. Klein. Learning bilingual lexicons from monolingual corpora. In *Proc. ACL*, 2008.

N. Halko, P.G. Martinsson, and J.A. Tropp. Finding structure with randomness : Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2) :217–288, 2011.

R.A. Harshman. Foundations of the Parafac procedure : Models and conditions for an explanatory multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16 :1–84, 1970.

R.A. Harshman and M.E. Lundy. Parafac : Parallel factor analysis. *Comput. Stat. Data Anal.*, 18(1) :39–72, 1994.

J. Håstad. Tensor rank is NP-complete. *J. Algorithms*, 11(4) :644–654, 1990.

T. Hazan, S. Polak, and A. Shashua. Sparse image coding using a 3D non-negative tensor factorization. In *Proc. ICCV*, 2005.

C.C. Heyde. On a property of the lognormal distribution. *J. R. Stat. Soc. Series B*, 25 :392–393, 1963.

C.J. Hillar and L.-H. Lim. Most tensor problems are NP-hard. *J. ACM*, 60(6), 2013.

F.L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *J. Math. Phys.*, 6 :164–189, 1927a.

F.L. Hitchcock. Multilple invariants and generalized rank of a p-way matrix or tensor. *J. Math. Phys.*, 7 :39–79, 1927b.

T. Hofmann. Probabilistic latent semantic indexing. In *Proc. SIGIR*, 1999a.

T. Hofmann. Probabilistic latent semantic analysis. In *Proc. UAI*, 1999b.

T. Hofmann, J. Puzicha, and M.I. Jordan. Learning from dyadic data. In *Adv. NIPS*, 1999.

R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 2013.

H. Hotelling. Relations between two sets of variates. *Biometrica*, 28(3/4) :321–377, 1936.

D. Hsu and S.M. Kakade. Learning mixtures of spherical Gaussians : Moment methods and spectral decompositions. In *Proc. ITCS*, 2013.

A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. Neural Netw.*, 10(3) :626–634, 1999.

A. Hyvärinen and P. Hoyer. Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Comput.*, 12(7) :1705–1720, 2000.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, 2001.

R. Iferroudjene, K. Abed Meraim, and A. Belouchrani. A new Jacobi-like method for joint diagonalization of arbitrary non-defective matrices. *Appl. Math. Comput.*, 211 :363–373, 2009.

T.s. Jaakkola. Tutorial on variational approximation methods. In M. Opper and D. Saad, editors, *Advanced Mean Field Methods : Theory and Practice*, pages 129–159. MIT Press, 2001.

J. Jacod and P. Protter. *Probability Essentials*. Springer, 2004.

I. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002.

M.I. Jordan. *Learning in Graphical Models*. MIT Press, 1999.

M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *J. Mach. Learn. Res.*, 11 :517–553, 2010.

C. Jutten. *Calcul neuromimétique et traitement du signal : Analyse en composantes indépendantes*. PhD thesis, INP-USM Grenoble, 1987.

C. Jutten and J. Hérault. Blind separation of sources, part I : An adaptive algorithm based on neuromimetric architecture. *Signal Process.*, 24(1) :1–10, 1991.

H.F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrica*, 23 :187–200, 1958.

N. Keriven, A. Bourrier, R. Gribonval, and P. Pérez. Sketching for large-scale learning of mixture models. *Technical Report*, HAL-01329195, 2016a.

N. Keriven, A. Bourrier, R. Gribonval, and P. Pérez. Sketching for large-scale learning of mixture models. In *Proc. ICASSP*, 2016b.

J.R. Kettenring. Canonical analysis of several sets of variables. *Biometrica*, 58(3) : 433–451, 1971.

J. Kim, Y. He, and H. Park. Algorithms for nonnegative matrix and tensor factorizations : A unified view based on block coordinate descent framework. *J. Global Optim.*, 58(2) :285–319, 2014.

A. Klami, S. Virtanen, and S. Kaski. Bayesian exponential family projections for coupled data sources. In *Proc. UAI*, 2010.

A. Klami, S. Virtanen, and S. Kaski. Bayesian canonical correlation analysis. *J. Mach. Learn. Res.*, 14 :965–1003, 2013.

E. Kofidis and P.A. Regalia. On the best rank-1 approximation of higher-order supersymmetric tensors. *SIAM J. Matrix Anal. Appl.*, 23(3) :863–884, 2002.

T.G. Kolda. Orthogonal tensor decompositions. *SIAM J. Matrix Anal. Appl.*, 23(1) : 243–255, 2001.

T.G. Kolda. A counterexample to the possibility of an extension of the Eckart–Young low-rank approximation theorem for the orthogonal rank tensor decomposition. *SIAM J. Matrix Anal. Appl.*, 24(3) :762–767, 2003.

T.G. Kolda and B.W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3) :455–500, 2009.

W.P. Krijnen and J.M.F. Ten Berge. Contrastvrije oplossingen van het Candecomp/Parafac-model. *Kwantitatieve Methoden*, 12(37) :87–96, 1991.

P.M. Kroonenberg. *Three-mode principal component analysis : Theory and applications*. PhD thesis, University of Leiden, 1983.

J.B. Kruskal. Three-way arrays : Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra Appl.*, 18 (2) :95–138, 1977.

J.B. Kruskal. Rank, decomposition, and uniqueness for 3-way and n-way arrays. In R. Coppi, S. Bolasco, F. Critchley, and Y. Escoufier, editors, *Multiway Data Analysis*, pages 7–18. Elsevier Science, 1989.

J.B. Kruskal, R.A. Harshman, and M.E. Lundy. How 3-MFA data can cause degenerate Parafac solutions, among other relationships. In R. Coppi, S. Bolasco, F. Critchley, and Y. Escoufier, editors, *Multiway Data Analysis*, pages 115–122. Elsevier Science, 1989.

H.W. Kuhn. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.*, 2(1-2) :83–97, 1955.

V. Kuleshov, A.T. Chaganty, and P. Liang. Tensor factorization via matrix factorization. In *Proc. AISTATS*, 2015a.

V. Kuleshov, A.T. Chaganty, and P. Liang. Simultaneous diagonalization : the asymmetric, low-rank, and noisy settings. Technical report, arXiv :1501.06318v2, 2015b.

J.M. Landsberg. *Tensors : Geometry and Applications.* American Mathematical Society, 2012.

D.D. Lee and H.S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401(6755) :788–791, 1999.

D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. In *Adv. NIPS*, 2001.

T.-W. Lee, M.S. Lewicki, M. Girolami, and T.J. Sejnowski. Blind source separation of more sources than mixtures using overcomplete representations. *IEEE Signal Process. Lett.*, 6(4) :87–90, 1999.

M.S. Lewicki and T.J. Sejnowski. Learning overcomplete representations. *Neural Comput.*, 12(2) :337–365, 2000.

W. Li and A. McCallum. Pachinko allocation : DAG-structured mixture models of topic correlations. In *Proc. ICML*, 2006.

Y.-O. Li, T. Adalı, W. Wang, and V.D. Calhoun. Joint blind source separation by multiset canonical correlation analysis. *IEEE Trans. Signal Proces.*, 57(10), 2009.

T. Lickteig. Typical tensorial rank. *Linear Algebra Appl.*, 69 :95–120, 1985.

L.-H. Lim. Singular values and eigenvalues of tensors : A variational approach. In *Proc. CAMSAP*, 2005.

L.-H. Lim and P. Comon. Nonnegative approximations of nonnegative tensors. *J. Chemometrics*, 23(7) :432–441, 2009.

G.D. Lin and J. Stoyanov. The logarithmic skew-normal distributions are moment-indeterminate. *J. Appl. Probab.*, 46 :909–916, 2009.

X. Luciani and L. Albera. Joint eigenvalue decomposition using polar matrix factorization. In *Proc. LVA ICA*, 2010.

L. Mackey. Deflation methods for sparse PCA. In *Adv. NIPS*, 2009.

F. Mangili and A. Benavoli. New prior near-ignorance models on the simplex. *Int. J. Approx. Reason.*, 56(B) :278–306, 2015.

J. Marcinkiewicz. Sur une propeiété de la loi de Gauß. *Mathematische Zeitschrift*, 44 :612–618, 1939.

McCullagh. *Tensor Methods in Statistics*. CRC Press, 1987.

G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 2nd edition, 2007.

M. Mella. Singularities of linear systems and the Waring problem. *Trans. Am. Math. Soc.*, 358(12) :5523–5538, 2006.

A. Mesloub, K. Abed-Meraim, and A. Belouchrani. A new algorithm for complex non-orthogonal joint diagonalization based on shear and Givens rotations. *IEEE Trans. Signal Process.*, 62(8), 2012.

K.P. Murphy. *Machine Learning : A Probabilistic Perspective*. MIT Press, 2012.

C. Nikias and J. Mendel. Signal processing with higher-order spectra. *IEEE Signal Process.*, 10(3) :10–37, 1993.

C. Nikias and A. Petropulu. *Higher-Order Spectra Analysis. A Nonlinear Signal Processing Framework*. Prentice Hall, 1993.

B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381 :607–609, 1996.

B.A. Olshausen and D.J. Field. Sparse coding with an overcomplete basis set : A strategy employed by V1 ? *Vision Res.*, 37(23) :3311–3325, 1997.

P. Paatero. A weighted non-negative least squares algorithm for three-way Parafac factor analysis. *Chemometr. Intell. Lab.*, 38(2) :223–242, 1997.

P. Paatero. Construction and analysis of degenerate Parafac models. *J. Chemometrics*, 14(3) :285–299, 2000.

P. Paatero and U. Tapper. Positive matrix factorization : A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2) : 111–126, 1994.

J. Paisley, D.M. Blei, and M.I. Jordan. Bayesian nonnegative matrix factorization with stochastic variational inference. In E.M. Airoldi, D.M. Blei, Erosheva E.A., and S.E. Fienberg, editors, *Handbook of Mixed Membership Models and Their Applications*, chapter 11, pages 203–222. CRC Press, 2014.

C.B. Papadias. Globally convergent blind source separation based on a multiuser kurtosis maximization criterion. *IEEE Trans. Signal Process.*, 48(12) :3508–3519, 2000.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philos. Mag.*, 2(6) :559–572, 1901.

A. Podosinnikova, S. Setzer, and M. Hein. Robust PCA : Optimization of the robust reconstruction error over the Stiefel manifold. In *Proc. GCPR*, 2014.

A. Podosinnikova, F. Bach, and S. Lacoste-Julien. Rethinking LDA : Moment matching for discrete ICA. In *Adv. NIPS*, 2015.

A. Podosinnikova, F. Bach, and S. Lacoste-Julien. Beyond CCA : Moment matching for multi-view models. In *Proc. ICML*, 2016.

J.K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2) :945–959, 2000.

L. Qi. Eigenvalues of a real supersymmetric tensor. *J. Symbolic Comput.*, 40 :1302–1324, 2005.

Y. Qi, P. Comon, and L.-H. Lim. Uniqueness of non-negative tensor approximations. *IEEE Trans. Inf. Theory*, 62(4) :2170–2183, 2016.

E.M. Robeva. *Decomposing matrices, tensors, and images*. PhD thesis, University of California, Berkeley, 2016.

S.M. Ross. *Introduction to Probability Models*. Elsevier, 10th edition, 2010.

S. Roweis. EM algorithms for PCA and SPCA. In *Adv. NIPS*, 1998.

J.-P. Royer, N. Thirion-Moreau, and P. Comon. Computing the polyadic decomposition of nonnegative third order tensors. *Signal Process.*, 91(9) :2159–2171, 2011.

A. Ruhe. On the quadratic convergene of a generalization of the Jacobi method to arbitrary matrices. *BIT Numer. Math.*, 8(3) :210–231, 1968.

S. Senecal and P.-O. Amblard. Bayesian separation of discrete sources via Gibbs sampling. In *Proc. ICA*, 2000.

S. Senecal and P.-O. Amblard. MCMC methods for discrete source separation. In *Proc. AIP*, 2001.

A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proc. ICML*, 2005.

N.D. Sidiropoulos and R. Bro. On the uniqueness of multilinear decomposition of N-way arrays. *J. Chemometrics*, 14(3) :229–239, 2000.

J. Sivic and A. Zisserman. Video Google : A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.

J. Sivic and A. Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(4) :591–606, 2009.

A. Slapak and A. Yeredor. Charrelation matrix based ICA. In *Proc. LVA ICA*, 2012a.

A. Slapak and A. Yeredor. Charrelation and charm : Generic statistics incorporating higher-order information. *IEEE Trans. Signal Process.*, 60(10) :5089–5106, 2012b.

R. Socher and L. Fei-Fei. Connecting modalities : Semi-supervised segmentation and annotation of images using unaligned text corpora. In *Proc. CVPR*, 2010.

L. Song, A. Anandkumar, B. Dai, and B. Xie. Nonparametric estimation of multi-view latent variable models. In *Proc. ICML*, 2014.

D. Sontag and D.M. Roy. Complexity of inference in latent Dirichlet allocation. In *Adv. NIPS*, 2011.

A. Souloumiac. *Utilisation des statistiques d'ordre supérieur pour le filtrage et la séparation de sources en traitement d'antenne.* PhD thesis, Ecole Nationale Supérieure des Télécommunications, 1993.

A. Souloumiac. Nonorthogonal joint diagonalization by combining Givens and hyperbolic rotations. *IEEE Trans. Signal Process.*, 57(6) :2222–2231, 2009a.

A. Souloumiac. Joint diagonalization : Is non-orthogonal always preferable to orthogonal ? In *Proc. IEEE CAMSAP*, 2009b.

A. Souloumiac and J.-F. Cardoso. Performances en séparation de sources. In *Proc. GRETSI*, 1993.

A. Stegeman and N.D. Sidiropoulos. On Kruskal's uniqueness condition for the Candecomp/Parafac decomposition. *Linear Algebra Appl.*, 420(2-3) :540–552, 2007.

G.W. Stewart and J.G. Sun. *Matrix Perturbation Theory.* Academic Press, 1990.

M. Steyvers and T. Griffiths. Probabilistic topic models. In T.K. Landauer, D.S. McNamara, S. Dennis, and W. Kintsch, editors, *Handbook of Latent Semantic Analysis*, chapter 21, pages 427–448. Laurence Erlbaum, 2007.

J. Stoyanov. Determinacy of distributions by their moments. In *Proc. ICMSM*, 2006.

A. Stuart and K. Ord. *Kendall's Advanced Theory of Statistics, Volume 1 : Distribution Theory*. Oxford University Press, 6th edition, 1994.

Z. Szaboó, B. Póczos, and Lőrincz. Undercomplete blind subspace deconvolution. *J. Mach. Learn. Res.*, 8 :1063–1095, 2007.

Z. Szaboó, B. Póczos, and Lőrincz. Separation theorem for independent subspace analysis and its concequences. *Pattern Recogn.*, 45 :1782–1791, 2012.

Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical Dirichlet process. *J. Am. Stat. Assoc.*, 101(476) :1566–1581, 2006.

J.M.F. Ten Berge. Kruskal's polynomial for $2 \times 2 \times 2$ arrays and a generalization to $2 \times n \times n$ arrays. *Psychometrika*, 56(4) :631–636, 1991.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. R. Stat. Soc. Series B*, 58(1) :267–288, 1996.

M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *J. R. Stat. Soc. Series B*, 61(3) :611–622, 1999.

K. Todros and A.O. Hero. Measure transformed independent component analysis. Technical report, arXiv :1501.06318v2, 2013.

L.R. Tucker. Some mathematical notes for three-mode factor analysis. *Psychometrika*, 31(3) :279–311, 1966.

A. Vinokourov and M. Girolami. A probabilistic framework for the hierarchic organisation and classification of document collections. *J. Intell. Inf. Syst.*, 18(2/3) : 153–172, 2002.

A. Vinokourov, J.R. Shawe-Taylor, and N. Cristianini. Inferring a semantic representation of text via cross-language correlation analysis. In *NIPS*, 2002.

S. Virtanen. Bayesian exponential family projections. Master's thesis, Aalto University, 2010.

R. Vollgraf and K. Obermayer. Quadratic optimization for simultaneous matrix diagonalization. *IEEE Trans. Signal Process.*, 54(9) :3270–3278, 2006.

X. T. Vu, S. Maire, C. Chaux, and N. Thirion-Moreau. A new stochastic optimization algorithm to decompose large nonnegative tensors. *IEEE Trans. Signal Process.*, 22(10), 2015.

H.M. Wallach, D. Mimno, and A. McCallum. Rethinking LDA : Why priors matter. In *Adv. NIPS*, 2009a.

H.M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proc. ICML*, 2009b.

C. Wang, X. Liu, Y. Song, and J. Han. Scalable moment-based inference for latent Dirichlet allocation. In *Proc. KDD*, 2014.

X. Wang and E. Grimson. Spatial latent Dirichlet allocation. In *Adv. NIPS*, 2008.

Y. Wang, H.-Y. Tung, A. Anandkumar, and A. Smola. Fast and guaranteed tensor decomposition via sketching. In *Adv. NIPS*, 2015.

P.-A. Wedin. Perturbation bounds in connection with singular value decomposition. *BIT Numer. Math.*, 12(1) :99–111, 1972.

M. Welling and M. Weber. Positive tensor factorization. *Pattern Recogn. Lett.*, 22 (12) :1255–1261, 2001.

A. Yeredor. Blind source separation via the second characteristic function. *Signal Process.*, 80(5) :897–902, 2000.

A. Yeredor. Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Trans. Signal Process.*, 50(7) :1545–1553, 2002.

A. Yeredor, A. Ziehe, and K.-R. Müller. Approximate joint diagonalization using a natural gradient approach. In *Proc. ICA*, 2004.

G. Zhou, A. Cichocki, Q. Zhao, and S. Xie. Nonnegative matrix and tensor factorizations : An algorithmic perspective. *IEEE Signal Process. Mag.*, 31(3) :54–65, 2014.

A. Ziehe, P. Laskov, G. Nolte, and K.-R. Müller. A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *J. Mach. Learn. Res.*, 5 :777–800, 2004.

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *J. Comp. Graph. Stat.*, 15(2) :265–286, 2006.

Résumé

Les modèles linéaires latentes sont des modèles statistique puissants pour extraire la structure latente utile à partir de données non structurées par ailleurs. Ces modèles sont utiles dans de nombreuses applications telles que le traitement automatique du langage naturel et la vision artificielle. Pourtant, l'estimation et l'inférence sont souvent impossibles en temps polynomial pour de nombreux modèles linéaires latents et on doit utiliser des méthodes approximatives pour lesquelles il est difficile de récupérer les paramètres.

Plusieurs approches, introduites récemment, utilisent la méthode des moments. Elles permettent de retrouver les paramètres dans le cadre idéalisé d'un échantillon de données infini tiré selon certains modèles, mais ils viennent souvent avec des garanties théoriques dans les cas où ce n'est pas exactement satisfait. Ceci n'est pas le cas pour les méthodes couramment utilisés, fondées sur l'inférence variationnelle et l'échantillonnage ce qui rend les méthodes à base de moment particulièrement intéressantes.

Dans cette thèse, nous nous concentrons sur les méthodes d'estimation fondées sur l'appariement de moment pour différents modèles linéaires latents. L'utilisation d'un lien étroit avec l'analyse en composantes indépendantes, qui est un outil bien étudié par la communauté du traitement du signal, nous présentons plusieurs modèles semiparamétriques pour la modélisation thématique et dans un contexte multi-vues. Nous présentons des méthodes à base de moment ainsi que des algorithmes pour l'estimation dans ces modèles, et nous prouvons pour ces méthodes des résultats de complexité améliorée par rapport aux méthodes existantes. Nous donnons également des garanties d'identifiabilité, contrairement à d'autres modèles actuels. C'est une propriété importante pour assurer leur interprétabilité.

Pour tous les modèles mentionnés, nous effectuons une comparaison expérimentale extensive des algorithmes associés, à la fois sur des données synthétiques et des données réelles. Elle démontre leurs bonnes performances en pratique.

Mots Clés

modèles thématique, modèles à variable latents, méthode des moments

Abstract

Latent linear models are powerful probabilistic tools for extracting useful latent structure from otherwise unstructured data and have proved useful in numerous applications such as natural language processing and computer vision. However, the estimation and inference are often intractable for many latent linear models and one has to make use of approximate methods often with no recovery guarantees.

An alternative approach, which has been popular lately, are methods based on the method of moments. These methods often have guarantees of exact recovery in the idealized setting of an infinite data sample and well specified models, but they also often come with theoretical guarantees in cases where this is not exactly satisfied. This is opposed to more standard and widely used methods based on variational inference and sampling and, therefore, makes moment matching-based methods especially interesting.

In this thesis, we focus on moment matching-based estimation methods for different latent linear models. Using a close connection with independent component analysis, which is a well studied tool from the signal processing literature, we introduce several semiparametric models in the topic modeling context and for multi-view models and develop moment matching-based methods for the estimation in these models. These methods come with improved sample complexity results compared to the previously proposed methods. The models are supplemented with the identifiability guarantees, which is a necessary property to ensure their interpretability. This is opposed to some other widely used models, which are unidentifiable.

For all mentioned models, we perform extensive experimental comparison of the proposed algorithms on both synthetic and real datasets and demonstrate their promising practical performance.

Keywords

topic models, latent variable models, method of moments

Numéro national de thèse :
xxxxx