GENERALIZED TENSOR FACTORIZATION

by

Yusuf Kenan Yılmaz

B.S., Computer Engineering, Boğaziçi University, 1992

M.S., Biomedical Engineering, Boğaziçi University, 1998

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering

Boğaziçi University

2012

ACKNOWLEDGEMENTS

I am very grateful to my advisor Ali Taylan Cemgil. It was, indeed, a risk to accept a 40-years-old person to supervise for a PhD on machine learning. He took the risk, introduced many valuable ideas such as Bregman divergence besides many others, and oriented me to tensor factorization.

Further my advisor, I thank to my family at home for their understanding throughout my PhD years and especially for the period of NIPS deadlines. I thank to my eleven-years-old daughter Mercan İrem for proposing and playing Beethoven's Ode to Joy for piano recording. I thank to my four-years-old son Sarp Deniz for awaking his father during nights to work on his PhD. I am also very grateful to their mother, my wife, Sibel, for her patience and for sharing her husband with the research. She hopes it finishes one day, but will it?

During my PhD study, my nephew Levent Doğuş Sagun has been around and about to complete his BS degree on Mathematics. I thank to him for sharing his thought about what we work on and answering my preliminary questions on matrix calculus and linear algebra.

In addition, I would like to thank to Cédric Fevotte for useful comments and several corrections on my thesis, to Evrim Acar for the fruitful discussion on the coupled tensor factorization as well as the corrections and to Umut Şimşekli for carring out the audio experiment.

Finally I should thank to Wiki people, Wikipedia, which is my immediate look up engine and start up point for almost all machine learning concepts. By the time of writing this thesis they really deserve any help to survive. Another group of people I should thank to those develop and share TeXlipse which I have used extensively during my PhD years.

ABSTRACT

GENERALIZED TENSOR FACTORIZATION

This thesis proposes a unified probabilistic framework for modelling multiway data. Our approach establishes a novel link between probabilistic graphical models and tensor factorization, that allows us to design arbitrary factorization models utilizing major class of the cost functions while retaining simplicity. Using an expectation-maximization (EM) optimization for maximizing the likelihood (ML) and maximizing the posterior (MAP) of the exponential dispersions models (EDM), we obtain generalized iterative update equations for beta divergence with Euclidean (EU), Kullback-Leibler (KL), and Itakura-Saito (IS) costs as special cases. We then cast the update equations into multiplicative update rules (MUR) and alternating least square (ALS for Euclidean cost) for arbitrary structures besides the well-known models such as CP (PARAFAC) and TUCKER3. We, then, address the model selection issue for any arbitrary non-negative tensor factorization model with KL error by lower bounding the marginal likelihood via a factorized variational Bayes approximation. The bound equations are generic in nature such that they are capable of computing the bound for any arbitrary tensor factorization model with and without missing values. In addition, further the EM, by bounding the step size of the Fisher Scoring iteration of the generalized linear models (GLM), we obtain general factor update equations for real data and multiplicative updates for non-negative data. We, then, extend the framework to address the coupled models where multiple observed tensors are factorized simultaneously. We illustrate the results on synthetic data as well as on a musical audio restoration problem.

ÖZET

GENELLEŞTİRİLMİŞ TENSOR AYRIŞIMI

Bu tez çok boyutlu veri yapıları için genel istatistiksel bir modelleme sistemi önermektedir. Kullandığımız yöntem istatistiksel grafik modelleri ile tensor ayrışımı arasında bir bağ kurarak çok boyutlu ayrışım modellerinin çeşitli maaliyet fonksiyonları için tasarlanmasını ve çözümlenmesini sağlar. *Beklenti-En Büyütme* (EM) tekniği ile *Exponensiyal Sapma Modellerinin* olabilirlik değerlerini en iyileyerek beta-divergence için yinelemeli güncelleme denklemlerini elde ediyoruz. Özel durum olarak Öklid, Kullback-Leibler ve Itakura-Saito maliyet fonksiyonlarını kullanıyoruz. Bu denklemleri, daha sonra, *Çarpanlı Güncelleme Kuralı* (MUR) ve Öklid maliyet fonksiyonu için *Dönüşümlü En Küçük Kareler* (ALS) denklemlerine dönüştürüyoruz. Ardından KL maliyet fonksiyonu kullanan pozitif parametreli alternatif modeller arasında seçim yapabilen bir yöntem geliştirdik. Bu yöntem marjinal olabilirlik değerini (doğrudan hesaplanamadığından) alt sınırdan yuvarlayan variational Bayes tekniğine dayanmaktadır. Ayrıca, EM yanında, *Geneleştirilmiş Doğrusal Modeller* (GLM) teorisinin Fisher Skoru olarak bilinen adım uzunluğunu sınırlandırarak pozitif ve reel sayılar için ayrı genel güncelleme denklemleri geliştirdik. Bu sistemi daha sonra birden fazla gözlem tensorlerin aynı anda çarpanlara ayrılma işleminde kullandık. Geliştirdiğimiz sistemi sentetik verilerle ve ayrıca müzik restorasyon problemini çözmek için kullandık.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES xi
LIST OF TABLES xiv
LIST OF SYMBOLS xvi
LIST OF ACRONYMS/ABBREVIATIONS xviii
1. INTRODUCTION 1
1.1. Extraction of Meaningful Information via Factorization 1
1.2. Multiway Data Modeling via Tensors 3
1.2.1. Relational Database Table Analogy of Tensors 3
1.2.2. Tensor Factorization 6
1.2.3. Learning the Factors 8
1.2.4. Bregman Divergence as Generalization of Cost Functions 10
1.2.5. Link between the Divergences and the Distributions: A Generative Model Perspective 11
1.2.6. Choice of the Generative Model 14
1.2.7. Prior Knowledge 15
1.2.8. Bayesian Model Selection 16
1.3. Structure of this Thesis 19
1.3.1. Motivation and Inspiration 19
1.3.2. Coupled Tensor Factorization 19
1.3.3. Methodology 20
1.3.4. Contributions 21
1.3.5. Thesis Outline 24
2. MATHEMATICAL BACKGROUND 26
2.1. Introduction 26
2.2. Exponential Family 27
2.2.1. Entropy of Exponential Family Distributions 34

2.2.2. Relating Dual of Cumulant Function and Entropy 35
2.2.3. Bregman Divergence 38
2.2.4. Bijection between Exponential Family and Bregman Divergence 40
2.2.5. Conjugate Priors for Exponential Family 42
2.3. Expectation Maximization (EM) 45
2.4. Bayesian Model Selection 49
2.4.1. Asymptotic Analysis 50
2.4.2. Bayesian Information Criterion (BIC) 53
2.5. Bayesian Information Criterion (BIC) for Gamma Distribution 55
2.5.1. Stirling's Formula for Gamma Function 55
2.5.2. Asymptotic Entropy of Gamma Distribution 56
2.6. Tensor Factorization Models 58
2.7. Graphical Models 63
2.8. Summary 65
3. EXPONENTIAL DISPERSION MODELS IN TENSOR FACTORIZATION 68
3.1. Introduction 68
3.2. Background: Exponential Dispersion Models 69
3.3. From PVF to Beta and Alpha Divergence 70
3.3.1. Derivation of Conjugate (Dual) of Cumulant Function 71
3.3.2. Beta divergence 73
3.3.3. Alpha Divergence 75
3.4. Likelihood, Deviance and β-divergence 76
3.4.1. Derivation of Cumulant Function 76
3.4.2. Log Likelihood of Exponential Dispersion Models 78
3.4.3. Entropy of Exponential Dispersion Models 79
3.4.4. The Deviance and the β-divergence 80
3.5. ML and MAP Optimization for EDMs 82
3.6. ML and MAP Optimization of the Data Augmentation Model 84
3.6.1. EM Algorithm for the Exponential Dispersion Models 84
3.6.2. Updates via EM, MUR and Alternating Projections for ML . . 87
3.6.3. Conjugate priors and MAP Estimation 89

3.6.4. Updates via EM and MUR for MAP 89
3.7. Solution for Non-linearity 90
3.7.1. Matching Link and Loss Functions 91
3.7.2. Optimizing for the Factors 94
3.8. Summary 96
4. PROBABILISTIC LATENT TENSOR FACTORIZATION 98
4.1. Introduction 98
4.2. Latent Tensor Factorization Model 99
4.2.1. Probability Model 103
4.2.2. Fixed Point Update Equation for KL Cost 104
4.2.3. Priors and Constraints 106
4.2.4. Relation to Graphical Models 107
4.3. Generalization of TF for β-Divergence 109
4.3.1. EM Updates for ML Estimate 111
4.3.2. Multiplicative Updates for ML Estimate 111
4.3.3. Direct Solution via Alternating Least Squares 116
4.3.4. Update Rules for MAP Estimation 116
4.3.5. Missing Data and Tensor Forms 117
4.4. Representation and Implementation 119
4.4.1. Matricization 119
4.4.2. Junction Tree for the Factors of TUCKER3 125
4.5. Discussion 127
4.6. Summary 127
5. MODEL SELECTION FOR NON-NEGATIVE TENSOR FACTORIZATION WITH KL 130
5.1. Introduction 130
5.2. Model Selection for $PLTF_{KL}$ Models 130
5.2.1. Bayesian Model Selection 131
5.2.2. Model Selection with Variational Methods 131
5.3. Variational Methods for $PLTF_{KL}$ Models 132
5.3.1. Tensor Forms via Δ Function 138

5.3.2. Handling Missing Data 139
5.4. Likelihood Bound via Variational Bayes 140
5.5. Asymptotic Analysis of Model Selection 145
5.5.1. Generalization of Asymptotic Variational Lower Bound 146
5.5.2. Variational Bound and Bayesian Information Criterion (BIC) . 147
5.6. Experiments 148
5.6.1. Experiment 1 149
5.6.2. Experiment 2 151
5.7. Summary 153
6. GENERALIZED TENSOR FACTORIZATION 157
6.1. Introduction 157
6.2. Generalized Linear Models for Matrix/Tensor Factorization 157
6.2.1. Two Equivalent Representations via Vectorization 160
6.3. Generalized Tensor Factorization 161
6.3.1. Iterative Solution for GTF 166
6.3.2. Update Rules for Non-Negative GTF 168
6.3.3. General Update Rule for GTF 171
6.3.4. Handling of the Missing Data 175
6.4. Summary 175
7. GENERALIZED COUPLED TENSOR FACTORIZATION 177
7.1. Introduction 177
7.2. Coupled Tensor Factorization 178
7.2.1. General Update for GCTF 181
7.3. Mixed Costs Functions 182
7.4. Experiments 183
7.4.1. Audio Experiments 185
7.5. Summary 188
7.6. Implementation 189
8. CONCLUSION 192
8.1. Future Work 193
APPENDIX A: PLTF FOR GAUSSIAN GENERATIVE MODELS 196

A.1. Gaussian Modelling of the EU Cost 196
A.2. Gaussian Modelling of the IS Cost 198
APPENDIX B: EXPONENTIAL FAMILY DISTRIBUTIONS 200
B.1. Exponential Family Distributions 200
B.1.1. Gamma Distributions 200
B.1.2. Poisson Distribution 202
B.1.3. Relation of the Poisson and Multinomial distributions 203
B.1.4. KL divergence of Distributions 206
APPENDIX C: MISCELLANEOUS 208
C.1. Miscellaneous 208
C.1.1. Matrix Operations 208
C.1.2. Gamma function and Stirling Approximation 208
REFERENCES 210

LIST OF FIGURES

Figure 1.1. Extraction of meaningful information via factorization. 2

Figure 1.2. Physical interpretation of factors obtained from a spectrogram of a piano recording. 4

Figure 1.3. Tensors as relational database tables. 5

Figure 1.4. Visualization of CP factorization. 8

Figure 1.5. Link between various divergence functions and probabilistic generative models. 13

Figure 1.6. Demonstration of model selection problem for identifying the notes for a piano recording. 18

Figure 1.7. Illustrative example for a coupled tensor factorization problem. . . 20

Figure 1.8. Application for coupled tensor factorization. 21

Figure 2.1. Bregman divergence illustration for Euclidean distance and KL divergence. 39

Figure 2.2. Peak of the posterior distribution around MAP estimate. 43

Figure 2.3. EM iterations: E-step and M-step. 49

Figure 2.4. Asymptotic normality of the posterior distribution. 51

Figure 2.5. Visualization of CP factorization. 60

Figure 2.6. Classical sprinkler example for graphical models. 65

Figure 2.7. Conversion of a DAG to a junction tree. 66

Figure 3.1. Classification of EDM distributions. 71

Figure 4.1. A graphical view of probabilistic latent tensor factorization. 101

Figure 4.2. Undirected graphical models for TUCKER3. 108

Figure 4.3. Monotone increase of the likelihood for TUCKER3 factorization. . 113

Figure 4.4. Graphical visualization of PARATUCK2 nested factorization. . . . 122

Figure 4.5. Algorithm: Implementation for TUCKER3 MUR for β-divergence. 125

Figure 4.6. Junction tree for TUCKER3 factorization. 126

Figure 5.1. Algorithm: Model selection for KL cost for PLTF. 150

Figure 5.2. Model order determination for TUCKER3. 151

Figure 5.3. Model order selection comparison for CP generated data. 152

Figure 5.4. Reconstructing the images with missing data by the CP model. . . 153

Figure 5.5. Comparing bound vs model order for the image data with missing values 154

Figure 7.1. CP/MF/MF coupled tensor factorization. 178

Figure 7.2. Solution of CP/MF/MF coupled tensor factorization. 185

Figure 7.3. Restoration of the missing parts in audio spectrograms. 186

Figure 7.4. Solving audio restoration problem via GCTF. 188

Figure 7.5. Algorithm: Coupled tensor factorization for non-negative data. . . 190

Figure 7.6. Algorithm: Coupled tensor factorization for real data. 191

LIST OF TABLES

Table 1.1. History of the factorization of the multiway dataset. 7

Table 2.1. Various characteristics popular exponential family distributions. . . 31

Table 2.2. Canonical characteristics of popular exponential family distributions. 41

Table 2.3. Element-wise and matrix representations of tensor models. 63

Table 3.1. Power variance functions. 70

Table 3.2. Variance functions and canonical link functions. 92

Table 4.1. Graphical representation of popular factorization models. 102

Table 4.2. Δ functions for popular tensor factorization models. 110

Table 4.3. EM update equations and posterior expectations. 115

Table 4.4. Factors updates in tensor forms via $\Delta_\alpha()$ function. 118

Table 4.5. Index notation used for matricization. 120

Table 4.6. MUR and ALS updates for tensor factorization models. 124

Table 6.1. Orders of various objects in the GTF update equation. 167

Table B.1. Exponential family distributions. 205

Table B.2. Entropy of distributions. 206

LIST OF SYMBOLS

A_α	Hyperparameters tensor for factor α
$A \otimes B$	Kronecker product
$A \odot B$	Khati-Rao product
$A \circ B$	Hadamard (element-wise) product
B_α	Hyperparameters tensor for factor α
$\mathcal{B}(.)$	Lower bound of the log-likelihood
$d_\alpha(x, \mu)$	Alpha divergence of x from μ
$d_\beta(x, \mu)$	Beta divergence of x from μ
$d_\phi(x, \mu)$	Divergence of x from μ with respect to convex function $\phi(.)$
$E[x]$	Expected value of x
$H[x]$	Entropy of x
i, j, k, p, q, r	Indices of tensors and nodes of graphical models
I	Identity matrix
$\mathcal{L}(\theta)$	Log-likelihood of θ
$t(x)$	Sufficient statistics of the variable x
v	Index configuration for all indices
v_0	Index configuration for observables
$v_{0,\nu}$	Index configuration for observable ν
v_α	Index configuration for factor α
$v(x)$	Variance function of x
$\|v\|$	Cardinality of the index configuration v
$\mathbf{vec}(X)$	Vectorization of the matrix/tensor X
$\mathrm{Var}(x)$	Variance of x as $\mathrm{Var}(x) = \varphi^{-1}v(x)$
$\mathcal{V}$	Set of indices
$\mathcal{V}_0$	Set of indices for observables X
$\mathcal{V}_\alpha$	Set of indices for latent tensor Z_α
x	Random variable
$\hat{x}$	Expected value of the variable x

X	Observation tensor
$\hat{X}$	Model estimate for the observation tensor
$\vec{X}$	Vectorization of the matrix/tensor X
$X^{i,j,k}$	Specific element of a tensor, a scalar
X_ν	Observation tensor indexed by ν
$\hat{X}_\nu$	Model estimate for the observation tensor ν
$X(i,j,k)$	Specific element of a tensor, a scalar
$X_{(n)}$	n-mode unfolding for tensor X
$\langle x \rangle$	Expected value of the variable x
$\left[X^{i,j,k}\right]$	A tensor formed by iterating the indices of the scalar $X^{i,j,k}$
Z	Tensor Factors to be estimated (latent tensors)
$Z_{1:\lvert\alpha\rvert}$	Latent tensors indexed from 1 to $\lvert\alpha\rvert$
Z_α	Latent tensor indexed by α
α	Factor index such as Z_α
$\Delta_\alpha()$	Delta function for latent tensor α
μ	Expectation parameter, expectation of the sufficient statistics
θ	Canonical (natural) parameter
Θ	Parameter set composed of A_α, B_α
$\phi(\mu)$	Dual of the cumulant (generating) function
φ	Inverse dispersion parameter
$\psi(\theta)$	Cumulant (generating) function
∇_α	Derivative of a function of $\hat{X}$ with respect to Z_α
$\mathbf{1}$	All ones matrix/tensor

LIST OF ACRONYMS/ABBREVIATIONS

CP	Canonical Decomposition / Parallel Factor Analysis
DAG	Directed Acyclic Graph
EDM	Exponential Dispersion Models
EM	Expectation Maximization
EU	Euclidean (distance)
GCTF	Generalized Coupled Tensor Factorization
GLM	Generalized Linear Models
GM	Graphical Models
GTF	Generalized Tensor Factorization
ICA	Independent Components Analysis
IS	Itakura-Saito (divergence)
KL	Kullback-Leibler (divergence)
LSI	Latent Semantic Indexing
MAP	Maximum A-Posteriori
MF	Matrix Factorization
ML	Maximum Likelihood
MUR	Multiplicative Update Rules
NMF	Non-negative Matrix Factorization
PARAFAC	Parallel Factor Analysis
PCA	Principal Component Analysis
PMF	Probabilistic Matrix Factorization
PLTF	Probabilistic Latent Tensor Factorization
PVF	Power Variance Function
TF	Tensor Factorization
VF	Variance Function

1. INTRODUCTION

The main challenge facing computer science in the 21st century is the large data problem. In many applied disciplines, on one hand, data accumulates faster than we can process it. On the other hand, advances in computing power, data acquisition, storage technologies made it possible to collect and process huge amounts of data in those disciplines. Applications in these diverse fields such as finance, astronomy, bio-informatics, acoustics require effective and efficient computational tools for processing huge amounts of data for extracting useful information.

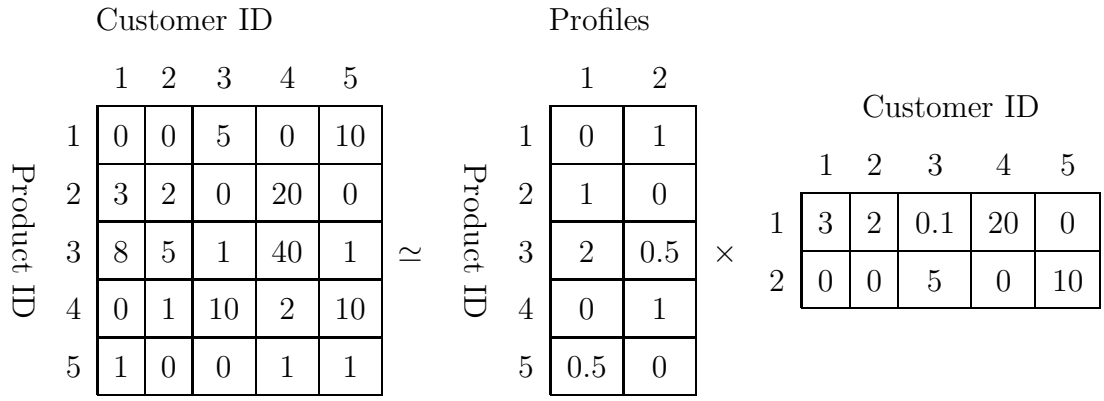
Popular data processing and dimensionality reduction methods of machine learning and statistics that scale well with large datasets are clustering [1], source separation, principal component analysis (PCA), independent components analysis (ICA) [2, 3], non-negative matrix factorization (NMF) [4, 5], latent semantic indexing (LSI) [6], collaborative filtering [7] and topic models [8] to name a few. In fact, many of these algorithms are usually understood and expressed as matrix factorization (MF) problems. Thinking of a matrix as the basic data structure facilitates parallelization and provides access to many of well understood algorithms with precise error analysis, performance guarantees and efficient standard implementations (e.g., SVD) [9].

1.1. Extraction of Meaningful Information via Factorization

As already mentioned that a fruitful modelling approach for extracting meaningful information from highly structured multivariate datasets is based on matrix factorizations (MFs). The matrix factorization reveals latent (hidden) structure in data that consists of two entities, i.e. factor matrices. Notationally, given a particular matrix factorization model, the objective is to estimate a set of latent factor matrices A and B

$$\text{minimize } D(X||\hat{X}) \text{ s.t. } \hat{X}^{i,j} = \sum_r A^{i,r} B^{j,r} \tag{1.1}$$

where i, j are observed indices, and r is latent index. Consider the following MF example in Table 1.1 originated from [10]. Each cell of the observation matrix on the left (Product ID $\times$ Customer ID) shows the quantities of a certain product bought by a specific customer. The objective is to estimate which products go along well, and to extract customer profiles. The profiles here have semantics such as customers buying sugar, flour and yeast could be associated with the act of cooking, and some others, for example, buying beer, snacks and balloons could be having a party. This problem may be regarded as a classification problem, where each profile is a class, noting that in this case one customer may belong to more than one profile. Extracting such user profiles is of paramount importance for many applications.



Product ID \ Customer ID	1	2	3	4	5
1	0	0	5	0	10
2	3	2	0	20	0
3	8	5	1	40	1
4	0	1	10	2	10
5	1	0	0	1	1

$\simeq$

Product ID \ Profiles	1	2
1	0	1
2	1	0
3	2	0.5
4	0	1
5	0.5	0

$\times$

Profiles \ Customer ID	1	2	3	4	5
1	3	2	0.1	20	0
2	0	0	5	0	10

Figure 1.1. Illustration of extracting meaningful information via factorization. Here customers are grouped into two profiles according to their purchase history.

Another interesting example for the information gained from the factorization is from [11], where they showed physical meaning of the factors of non-negative matrix factorization (NMF) [5] of musical spectra as

$$\hat{X} = WH \qquad \text{or} \qquad \hat{X}^{i,j} = \sum_r W^{i,r} H^{j,r} \qquad \text{s.t. } min\ D(X||\hat{X}) \tag{1.2}$$

with non-negativity constraint as $W, H \geq 0$. Here $D(X||\hat{X})$ is appropriate cost function, r is the latent dimension while W and H are latent factors or parameters of the model to be discovered. [11] showed that the factor W encodes the frequency content of the data while the factor H encodes the temporal information. Here we just repeat

this demonstration with a piano recording of the main theme of Ode to Joy (Symphony No.9 Op.125) of Beethoven given in Figure 1.2. The recording consists of 8 notes where four of them are distinct as D, E, F and G. The factor H in Figure 1.2a gives the position of the notes in time while the factor W in Figure 1.2b encodes the frequency information.

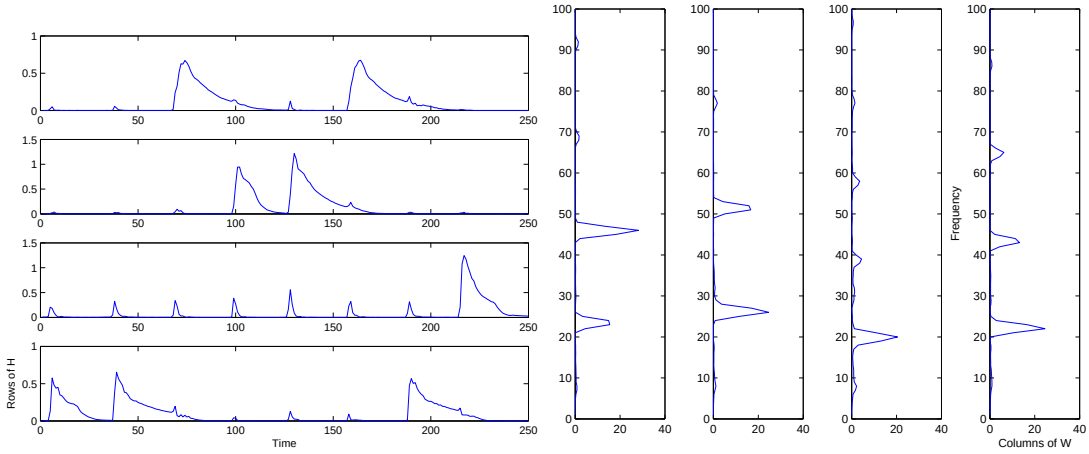
1.2. Multiway Data Modeling via Tensors

Increase in the amount of multidimensional digital data and processing capacity bring its own mathematical notational abstracts, i.e. tensors. Indeed, tensors appear as a natural generalization of matrices, when observed data and/or a latent representation have more than two semantically meaningful dimensions. Even further, we regard vectors, matrices and higher dimensional objects simply a single entity as *tensors* and threat all in the same mathematical notation.

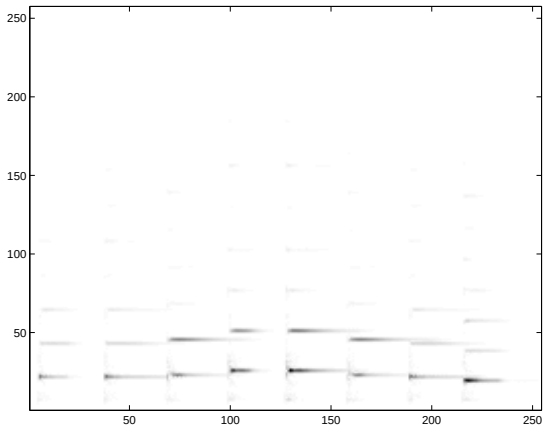
1.2.1. Relational Database Table Analogy of Tensors

In computer science, a tensor is analogous to a *normalized* relational database table where tensor indices correspond to key columns of primary key set. The analogy is based on the similarity that the indices in a tensor address each cell uniquely. Likewise the primary key set of a relational database table uniquely addresses rows of the table.

To make the link more precise and to illustrate that a tensor is indeed very natural representation, consider the following example. A researcher records temperature of certain locations and comes up with a relational database table similar to Table 1.3 where the location column is primary key column. Recall that by definition whenever two rows match on their primary key column they must match on all the other columns. The observer wants to add more information about his recording process, such as date information. Then the primary key can no longer be a single column, instead, the location and date columns form *primary key set* together. Even the recording can be further refined by adding altitude information. Then, the primary key set consists of three columns of location, date and altitude. In this way, the table turns to be a three



(a) H matrix (b) W matrix



(c) Spectrogram of the piano recording

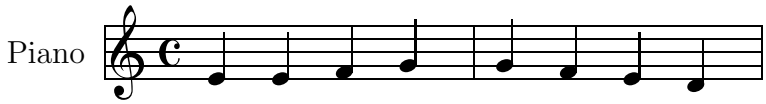
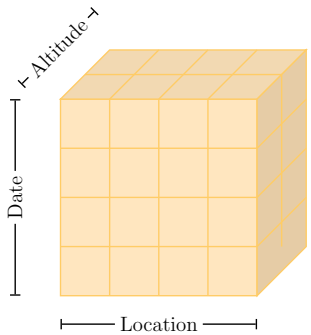


Figure 1.2. Physical interpretation of factors obtained from a spectrogram of the piano recording of Ode to Joy Symphony No.9 Op.125 of Beethoven. The recoding consists of 8 notes with 4 distinct as D, E, F and G. The factorization is as $X \simeq WH$ where (a) is the factor H that encodes the temporal information while (b) is the factor W encodes the frequency information.

Location	**Temp.**
Ankara	30
İzmir	36
İstanbul	28
Adana	10

Location	**Date**	**Temp.**
Ankara	2009	30
İzmir	2009	36
İstanbul	2010	28
Ankara	2010	10



Location	**Date**	**Altitude**	**Temp.**
Ankara	2009	500	30
İzmir	2009	0	36
İstanbul	2010	0	28
Ankara	2010	500	10
İzmir	2009	500	22

Figure 1.3. A tensor can be viewed as analogy of a normalized relational database table whose columns in primary key set are indices of the multidimensional objects.

dimensional object, i.e. to a cube with three axes as location, date and altitude, whose cells are temperature values. More formally, this cube is called as three-way tensor or the third-order tensor and it has three indices (dimensions). Note that, with this view, we regard the remaining columns not in primary key set as a single entity or a cell of the tensor which bound to real number in our further treatment.

Various real world multiway datasets can be viewed with the same formalism by choosing the appropriate factors from the related domains such as

EEG Data : Channel × Frequency × Time × Subject
Music Data : Channel × Frequency × Time × Song
Text Data : Documents × Terms × Language
fMRI Data : Voxels × Run × Time × Trial
Netflix Data : Movies × Users × Time
Enron Data : Users × Messages × Time

Clearly we could collapse multiway datasets to matrices but important structural information might get lost. Instead, a useful method for factorization of these multisets is to respect their multi-linear structure. There are many natural ways to factorize a multiway dataset and there exists a plethora of related models with distinct names, such as canonical decomposition (CP), PARAFAC, TUCKER3 to name a few as discussed in detail in excellent tutorial reviews [12–14].

1.2.2. Tensor Factorization

As summarized in Table 1.1 history of factorization of multiway datasets goes back to Hitchcock in 1927 [15]. Later it is popularized in the field of psychometrics in 1966 by Tucker [16] and in 1970 by Carroll, Chang and Harshman [17,18]. Besides psychometrics, over time many applications emerge in various fields such as chemometrics for analysis of fluorescence emission spectra, signal processing for audio applications, and biomedical for analysis of EEG data.

Table 1.1. History of the factorization of the multiway dataset. Models are named later than the year they introduced. 3MFA stands for *three-mode factor analysis* [13].

Year	**Authors**	**Model**	**Original Name**	**Field**
1927	Hitchcock	CP		Math. Physics
1966	Tucker	TUCKER3	3MFA	Psychometrics
1970	Harshman	CP	PARAFAC	Psychometrics
1970	Carroll & Chang	CP	CANDECOMP	Psychometrics
1996	Harshman & Lundy	CP+TUCKER2	PARATUCK2	Psychometrics

The proposed models are closely related, indeed. Going back to Hitchcock [15] in 1927, he proposed expressing a tensor as sum of finite number of rank-one tensors (simply outer product of vectors) as

$$\hat{X} = \sum_r v_{r,1} \times v_{r,2} \times \ldots \times v_{r,n} \tag{1.3}$$

which as an example for $n = 3$, i.e. as 3-way tensor, it can be expressed as

$$\hat{X}^{i,j,k} = \sum_r A^{i,r} B^{j,r} C^{k,r} \tag{1.4}$$

This special decomposition is discovered and named by many researchers independently such as CANDECOMP (canonical decomposition) [17] and PARAFAC (parallel factors) [18] where Kiers simply named them as CP [12]. In 1963, Tucker introduced a factorization which resembled high order PCA or SVD for the tensors [16]. It summarizes given tensor X into *core tensor* G considered to be a compressed version of X. Then, for each mode (simply the dimensions) there is a *factor matrix* desired to be orthogonal interacting with the rest of the factors as follows

$$\hat{X}^{i,j,k} = \sum_{pqr} G^{p,q,r} A^{i,p} B^{j,q} C^{k,r} \tag{1.5}$$

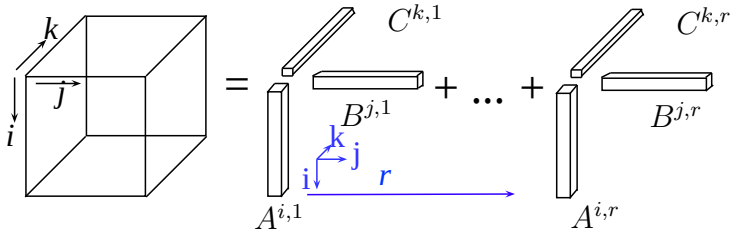


Figure 1.4. CP factorization of 3-way tensor as sum of $|r|$ rank-one tensors.

The factorization models emerged over the years have close relationship with each other. For example, the CP model is indeed a TUCKER3 model with identity (super-diagonal) core tensor. TUCKER factorization has two more variants as TUCKER1 and TUCKER2. PARATUCK2 is a combination of CP and TUCKER2. The relationships among models inspires us to develop a unified view and common formalization for general tensor factorization with appropriate notation. There are many benefits of such general treatment such as single development of parameter update equations, invention of new factorization models and even coupling of them that allows simultaneous factorization.

1.2.3. Learning the Factors

One of the principal methods used for learning the factors is the error minimization between the observation X and the model output $\hat{X}$ computed by using the estimated factors. The error, then is distributed back proportionally to the factors and they are adjusted accordingly in an iterative update schema. To quantify the quality of the approximation we use various cost functions denoted by $D(X||\hat{X})$. The iterative algorithm, then, optimizes the factors in the direction of the minimum error

$$\hat{X}^* = \arg\min_{\hat{X}} D(X||\hat{X}) \tag{1.6}$$

Squared Euclidean cost is the most common choice of available cost functions

$$D(X||\hat{X}) = \|X - \hat{X}\|^2 = \sum_{i,j} \left(X^{i,j} - \hat{X}^{i,j}\right)^2 \tag{1.7}$$

while one other is the Kullback-Leibler divergence defined as

$$D(X||\hat{X}) = \sum_{i,j} X^{i,j} \log \frac{X^{i,j}}{\hat{X}^{i,j}} - X^{i,j} + \hat{X}^{i,j} \tag{1.8}$$

In addition, KL becomes relative entropy when X and $\hat{X}$ are normalized probability distributions.

On the other hand, for some applications other cost functions might be more appropriate such that the divergence must not change if the signals are scaled by the same amount. Consider, on one hand, comparing signals $P(\omega)$ and $\hat{P}(\omega)$, and on the other hand, comparing their scaled versions $\lambda P(\omega)$ and $\lambda \hat{P}(\omega)$ where ω is frequency and $P(\omega)$ power at frequency ω. The scale-invariance property provides a kind of self-similarity such that low scaled quantity provides as much important and discrimination power as the high scaled quantity. Notationally, we can consider the scaled form of a function $f(x)$ under rescaling of the variable x as $f(\lambda x)$ for some scale factor λ. Then the scale invariance property implies that for any λ

$$f(\lambda x) = \lambda^{\alpha} f(x) \tag{1.9}$$

for some choice of exponent α. For an examples, $f(x) = x^n$ is a scale-invariant function since $f(\lambda x) = (\lambda x)^n = \lambda^n f(x)$ [19, 20]. One such cost function is the *Itakura-Saito divergence* (Burg cross entropy) proposed by F. Itakura and S. Saito in the 1970s [21] that measures the dissimilarity (or equivalently similarity) between a spectrum $P(\omega)$ and an approximation $\hat{P}(\omega)$ of that spectrum of the reconstructed signal. The Itakura-

Saito divergence [1] is defined as

$$D(X||\hat{X}) = \sum_{i,j} \frac{X^{i,j}}{\hat{X}^{i,j}} - \log \frac{X^{i,j}}{\hat{X}^{i,j}} - 1 \tag{1.11}$$

where it is clear that here the exponent α in Equation 1.9 is set to $\alpha = 0$

$$D(X||\hat{X}) = D(\lambda X||\lambda \hat{X}) \tag{1.12}$$

1.2.4. Bregman Divergence as Generalization of Cost Functions

Obtaining inference algorithms for the factors for various cost functions requires a separate optimization effort and is a time consuming task. For NMF, for example, the authors obtained two different versions of update equations for Euclidean and KL cost functions by a separate development [5]. On the other hand, the *Bregman divergence* enables us to express large class of divergence (cost) functions in the same expression [22]. Assuming ϕ be a convex function, the Bregman divergence $D_\phi(X||\hat{X})$ for matrix arguments is defined as

$$D_\phi(X||\hat{X}) = \sum_{i,j} \phi(X^{i,j}) - \phi(\hat{X}^{i,j}) - \frac{\partial \phi(X)}{\partial \hat{X}^{i,j}}(X^{i,j} - \hat{X}^{i,j}) \tag{1.13}$$

The Bregman divergence is a non-negative quantity as $D_\phi(X||\hat{X}) \geq 0$. It is zero if and only if $X = \hat{X}$. Major class of the cost functions can be generated by the Bregman divergence by applying appropriate functions $\phi(.)$. For example, squared Euclidean distance is obtained by the function $\phi(x) = \frac{1}{2}x^2$ while the KL divergence and the IS divergence are generated by the functions $\phi(x) = x \log x$ and $\phi(x) = -\log x$ respectively [22].

[1] It is originally defined as [21]

$$D_{IS}(P(\omega)||\hat{P}(\omega)) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left(\frac{P(\omega)}{\hat{P}(\omega)} - \log \frac{P(\omega)}{\hat{P}(\omega)} - 1 \right) d\omega \tag{1.10}$$

1.2.5. Link between the Divergences and the Distributions: A Generative Model Perspective

Here, the natural question is how to optimize Bregman divergence for the factors. On the other hand, statistical generative models have already well established optimization techniques such Expectation-Maximization (EM) and Fisher Scoring to maximize the likelihood. By exploiting one-to-one correspondence between the Bregman divergence and exponential family distributions [22] we can consider the optimization of the factors by maximizing the generative model's likelihood instead of minimizing the cost functions. We recall that exponential family of distributions are defined as

$$p(x|\theta) = h(x) \exp\big(t(x)^T\theta - \psi(\theta)\big) \tag{1.14}$$

where $h(.)$ is the base measure independent of the canonical parameter θ and $\psi(.)$ is the cumulant function while $t(x)$ is the sufficient statistics. Banerjee *et al.* proved the bijection between regular exponential family distributions and the Bregman divergence as [22]

$$\log p(x|\theta) = \log h_\phi(x) - d_\phi\big(t(x), \hat{x}\big) \tag{1.15}$$

By taking the derivative we note that maximizing the log likelihood is equivalent to minimizing the divergence function

$$\frac{\partial \log p(x|\theta)}{\partial \theta} = -\frac{\partial d_\phi\big(t(x), \hat{x}\big)}{\partial \theta} \tag{1.16}$$

Here, an important question is that to derive general update equations for the factors, which functions $\phi()$ we should use. Rather than using Bregman divergence and exponential family duality we can work with the beta divergence and *Exponential Dispersion Models* (EDM) [23] where we prove their dualities as well. Exponential

dispersion models are a subset of the exponential family of distributions defined as

$$p(x|\theta,\varphi) = h(x,\varphi)\exp\left\{\varphi\left(\theta x - \psi(\theta)\right)\right\} \tag{1.17}$$

where θ is the *canonical (natural) parameter*, φ is the inverse *dispersion parameter* and ψ is the cumulant generating function. One useful property of dispersion models is that they tie the variance of a distribution to its mean

$$Var(x) = \varphi^{-1} v(\hat{x}) \tag{1.18}$$

where here $\hat{x}$ is the mean of the variable x and $v(\hat{x})$ is the *variance function* [23–25]. A useful observation is that the variance function is in the form of power function and therefore it is called as *power variance functions* (PVF) [23, 25] given as

$$v(\hat{x}) = \hat{x}^p \tag{1.19}$$

By using this property we can identify the dual cumulant function of dispersion models as

$$\phi(\hat{x}) = \left((1-p)(2-p)\right)^{-1}\hat{x}^{2-p} + m\hat{x} + d \tag{1.20}$$

The function $\phi(.)$ is closely related to the entropy and the Bregman divergence corresponding the function $\phi(.)$

$$d_\phi(x,\hat{x}) = \phi(x) - \phi(\hat{x}) - (x-\hat{x})\frac{\partial\phi(\hat{x})}{\partial\hat{x}}$$

ends up with the beta divergence [26–28] that expresses major class of cost functions in a single expression

$$d_\beta(x,\hat{x}) = \left((1-p)(2-p)\right)^{-1}\left\{x^{2-p} - \hat{x}^{2-p} - (x-\hat{x})(2-p)\hat{x}^{1-p}\right\} \tag{1.21}$$

where for $p = 1, 2$ we have the special cases for KL and IS costs as the limits. This derivation shows that for cost function generalization perspective, minimizing the beta divergence is equivalent to maximizing the likelihood of the exponential dispersion model family. Figure 1.5 summarizes the link between divergence functions and probability models.

The probabilistic interpretation of the factorization models has many useful benefits such as incorporating the prior knowledge and Bayesian treatment for the model selection. In fact, matrix factorization models have well understood probabilistic/statistical interpretations as probabilistic generative models. Indeed, many standard algorithms mentioned above can be derived as maximum likelihood or maximum a-posteriori parameter estimation procedures. This approach is central in probabilistic matrix factorization (PMF) [29] or non-negative matrix factorization [30, 31]. It is also possible to do a full Bayesian treatment for model selection [30]. Indeed, this aspect provides a certain practical advantage over application specific hierarchical probabilistic models, and in this context, matrix factorization techniques have emerged as a useful modelling paradigm [5, 29, 32]. On the other hand, statistical interpretations on similar lines for tensor factorization focus on a given factorization structure only, such as CP or TUCKER3. For example, sparse non-negative TUCKER is discussed in [33], while probabilistic non-negative PARAFAC is discussed in [34–36].

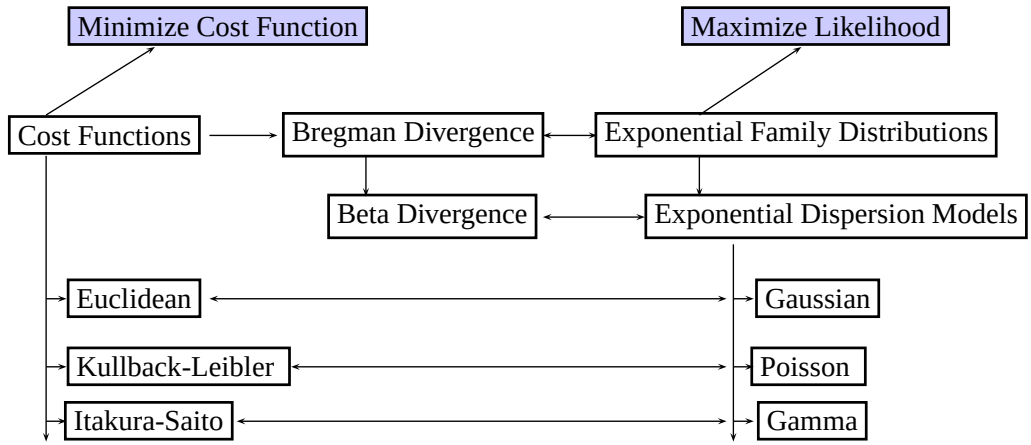


Figure 1.5. Link between various divergence functions and probabilistic generative models.

1.2.6. Choice of the Generative Model

Gaussian distribution is natural choice for the modeling of a real-world phenomenon. When developing a mathematical model we usually make certain assumptions about the underlying generative model. One of them is that the natural population frequencies are assumed to exhibit a normal, i.e. a Gaussian distribution. This assumption is well understood and is explained by the consequence of the *central limit theorem* which simply states that sum of i.i.d random variables tends to be the Gaussian variables as the number increases regardless type and shape of the original distribution. This is also referred as *asymptotic normality* [37]. It is one of the main theorems of the probability together with *strong law of large numbers* which simply states that mean of the sum converges to the mean of the original distribution.

However, for many phenomena the Gaussian assumption about the underlying model could be too simplified, and hence, some other distributions can be better suitable for the empirical data. In other words, the *maximum entropy distribution* [38] may not always be the Gaussian. For example, for many cases we may be interested in the count or the frequency of the events, such as DNA mutations, data packets passing trough a computer network router, or photons arriving at a detector are all representing countable events and can be modeled by the Poisson. To sum up, Poisson modeling is suitable to model counting variables including frequencies in contingency tables.

Another interesting example for using non-Gaussian models is from the seminal work [39] where the authors generalized the PCA to the exponential family. The authors compared regular PCA based on the Gaussian to exponential PCA based on exponential distribution. In the case where there was no outliers two versions of PCA produced similar results. However, in another case, where there were a few outliers in the data, they observed the larger effect on the regular PCA than the exponential PCA that concluded the robustness of the exponential PCA to the outliers.

In this thesis we have two generalization perspectives for tensor factorization;

generalization of the factorization arbitrary structures and generalization to a large class of cost functions. In Chapter 4 where we introduce *probabilistic latent tensor factorization* we mainly focus on Euclidean, KL and IS cost functions in the form of beta divergence that are related to the Gaussian, the Poisson and the gamma distributions. In Chapter 6 and 7 where we introduce *generalized tensor factorization* and its coupled extension we derive update equations general for *exponential dispersion models* which is a subset of the exponential family.

1.2.7. Prior Knowledge

One benefit of developing probabilistic generative models for the factorization is use of the prior knowledge. Every member of exponential family distributions has a conjugate prior which is also a member of the family. The use of the conjugate priors is that when a conjugate prior is multiplied by the corresponding likelihood function the resulting posterior distribution has the same functional form as the prior as illustrated by the Poisson likelihood with the gamma prior example

$$\text{gamma posterior} \propto \text{ Poisson likelihood} \times \text{ gamma prior} \tag{1.22}$$

Having the same functional form for the prior and the posterior makes the Bayesian analysis easier. Recall that Bayesian analysis is needed for computing a posterior distribution over parameters and computing a point estimate of the parameters that maximizes their posterior (MAP optimization).

One benefit of using priors is to impose certain constraints into the factorization problem. For example, for sparseness we can take a gamma prior for the Poisson generative model as $\mathcal{G}(x; a, a/m)$ with mean m, and variance m^2/a. For small α, most of the elements of the factors are expected to be around zero, while only a few are non-zero [30].

1.2.8. Bayesian Model Selection

For matrix factorization, we face a model selection problem that deals with choosing the model order, i.e. the cardinality of the latent index. For tensor factorization the problem is even more complex since besides the determination of the cardinality of the latent indices, selecting the right generative model among many alternatives can be a difficult task. In other words the actual structure of the factorization may be unknown. For example, given an observation $X^{i,j,k}$ with three indices one can propose a CP generative model as $\hat{X}^{i,j,k} = \sum_r A^{i,r}B^{j,r}C^{k,r}$, or a TUCKER3 model $\hat{X}^{i,j,k} = \sum_{p,q,r} A^{i,p}B^{j,q}C^{k,r}G^{p,q,r}$.

We address the determination of the correct number of the latent factors and their structural relations as well as cardinality of latent indices by the Bayesian model selection. For a Bayesian point of view, we associate a factorization model with a random variable m interacting with the observed data x simply as $p(m|x) \propto p(x|m)p(m)$. Then, we choose the model having the highest posterior probability such that $m^* = \arg\max_m p(m|x)$. Assuming the model priors $p(m)$ are equal the quantity $p(x|m)$ becomes important since comparing $p(m|x)$ is the same as comparing $p(x|m)$. The quantity $p(x|m)$ is called *marginal likelihood* [1] and it is the average over the parameter space as

$$p(x|m) = \int d\theta \; p(x|\theta, m)p(\theta|m) \tag{1.23}$$

Then comparing two models m_1 and m_2 for the observation x we use the ratio of the marginal likelihoods

$$\frac{p(x|m_1)}{p(x|m_2)} = \frac{\int d\theta_1 \; p(x|\theta_1, m_1)p(\theta_1|m_1)}{\int d\theta_2 \; p(x|\theta_2, m_2)p(\theta_2|m_2)} \tag{1.24}$$

where this ratio is known as *Bayes Factors* [1] and it is considered to be a Bayesian alternative for the frequentist hypothesis testing with likelihood ratio. However, computation of the integral for the marginal likelihood is itself a difficult task that requires

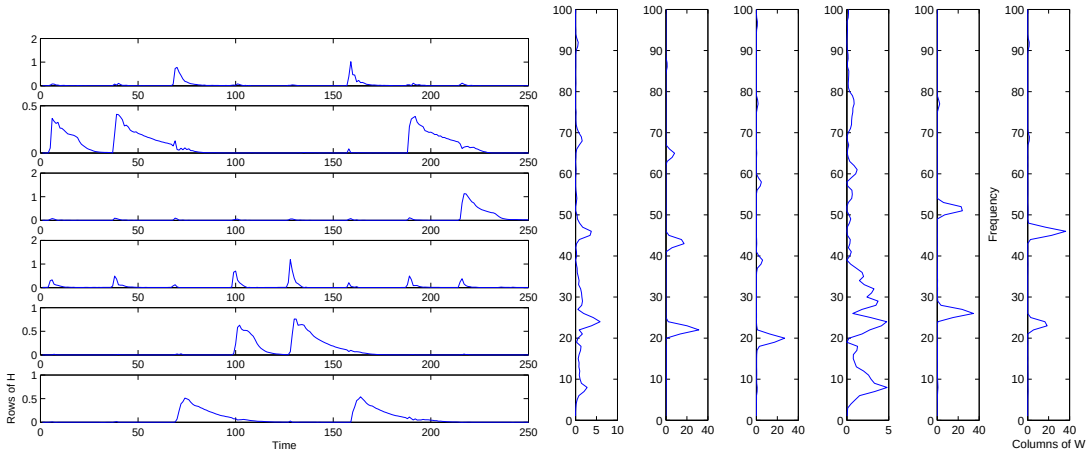
averaging on parameter space. There are several approximation methods such as sampling or deterministic approximations such as Gaussian approximation. One other approximation method is to bound the log marginal likelihood by using variational inference [1,40] where an *approximating distribution* q is introduced into the log marginal likelihood equation as in

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

To illustrate the importance of the model selection, reconsider the factorization of musical spectra example where we set the size of the latent index r to be equal to the number of distinct notes a priori. In most cases it is difficult to determine the correct latent index size beforehand, and this number is usually set by trial and error. For illustrative purposes in the musical spectra example we re-set the size of latent index as six while the true value is four. Figure 1.6 shows the plot of the factor H that encodes the temporal information along with the factor W that encodes the frequency information. Four out of six plots clearly locates the positions of four notes in time. The other two plots seem to be spiky versions.

One other important concern regarding the specifying the cardinality of latent indices is the uniqueness property. CP factorization yields unique factors up to scaling and permutation subject to Kruskal's condition. Kruskal first defined the concept of k-rank, denoted by k_A for a given matrix A, defined that it is the maximum number of randomly chosen columns that are linearly independent. In other word there is at least one set of $k_A + 1$ columns whose regular rank is less than $k_A + 1$. Then the uniqueness condition for CP is [41, 42]

$$k_A + k_B + k_C \geq 2|r| + 2 \tag{1.26}$$



(a) H matrix (b) W matrix

Figure 1.6. The figure illustrates the model selection problem for identifying the notes for a piano recording. The plots show the rows of the H factor that encodes the temporal information and the columns of the factor W that encodes the frequency information. It is interesting to compare the factor H with six latent components here to the factor H with four latent components (correct size) in Figure 1.2.

Hence when we choose the cardinality of latent index $|r|$ unnecessarily large as

$$|i| + |j| + |k| \leq 2|r| + 2 \tag{1.27}$$

we may not determine factors uniquely, meaning that two or more different sets of the factors may generate the same model.

In Chapter 4 we design a non-negative tensor factorization model selection framework with KL error by lower bounding the marginal likelihood via a factorized variational Bayes approximation. The bound equations are generic in nature such that they are capable of computing the bound for any arbitrary tensor factorization model with and without missing values.

1.3. Structure of this Thesis

1.3.1. Motivation and Inspiration

The motivation behind this thesis is to pave the way to a unifying tensor factorization framework where, besides well-known models, any arbitrary factorization model can be constructed and the associated inference algorithm can be derived automatically for major class of cost functions using matrix computation primitives. The framework should respect to the missing values and should take account the prior knowledge for the factorization. In addition, model selection and coupled simultaneous factorization are desired features of the framework. Finally, it should be practical and easy for implementation. This is very useful in many application domains such as audio processing, network analysis, collaborative filtering or vision, where it becomes necessary to design application specific, tailored factorizations.

Our unified notation is partially inspired by probabilistic graphical models, where we exploit the link between graphical models and tensor factorization. Our computation procedures for a given factorization have a natural message passing interpretation. This provides a structured and efficient approach that enables very easy development of application specific custom models, priors or error measures as well as algorithms for joint factorizations where an arbitrary set of tensors can be factorized simultaneously. Well known models of multiway analysis (CP, TUCKER [13]) appear as special cases and novel models and associated inference algorithms can be automatically developed.

1.3.2. Coupled Tensor Factorization

We formulate a motivating problem illustrated in Figure 1.7 and target to solve it in Chapter 7. A coupled tensor factorization problem [43] involves many observations tensors to be factorized simultaneously. As our generalization perspective (i) the numbers and structural relationship of the latent factor tensors are to be arbitrary and (ii) the fixed point update equations are to be derived for major class of cost functions.

Consider the following illustrative example

$$X_1^{i,j,k} \simeq \sum_r A^{i,r} B^{j,r} C^{k,r} \qquad X_2^{j,p} \simeq \sum_r B^{j,r} D^{p,r} \qquad X_3^{j,q} \simeq \sum_r B^{j,r} E^{q,r} \tag{1.28}$$

where X_1 is 3-mode observation tensor and X_2, X_3 are observation matrices, while $A : E$ are the latent matrices. Here, B is a shared factor, coupling all the models. Such models are of interest when one can obtain different views of the same piece of information (here B) under different experimental conditions. We note that we will not specifically solve this problem, but rather, we will build a practical framework that is easy to implement, and also general in the sense that any cost function can be applied. It is to be of the arbitrary size such that observation tensors can be many and they can interact with the latent tensors in any way.

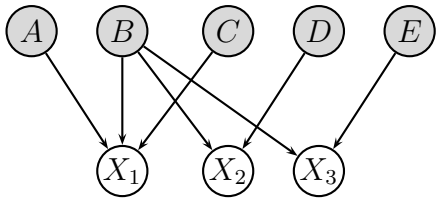


Figure 1.7. Illustrative example for a coupled tensor factorization problem.

The coupled tensor factorization can find various application areas easily. One example is restoration of the missing parts in audio spectrograms given in Figure 1.8. It is a difficult problem since entire time frames, i.e. columns of observation X can be missing. With coupling, however, we can incorporate different kinds of musical knowledge into a single model such as temporal and harmonic information from an approximate musical score, and spectral information from isolated piano sounds [44].

1.3.3. Methodology

In this thesis for the probabilistic tensor factorization we used two main methodologies. The first one uses one big latent augmentation tensor S in the middle of the observation X and the factors Z_α. S is never needed to be computed explicitly, and it disappears smoothly from the fixed point update equations for the latent tensors

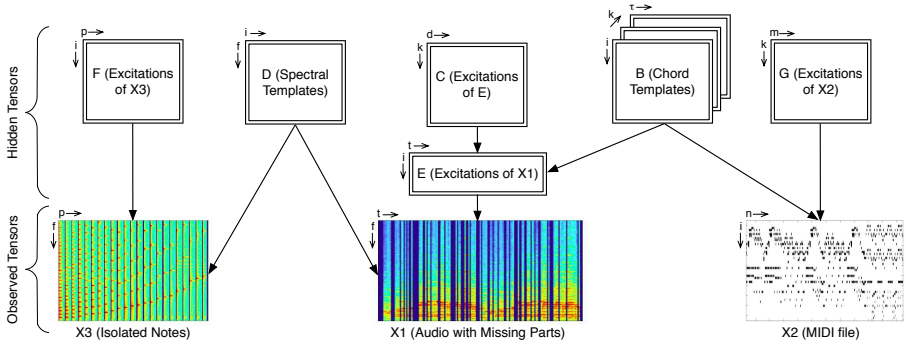


Figure 1.8. Restoration of the missing parts in audio spectrograms as a coupled tensor factorization problem. The musical piece to be reconstructed shares B and D in common.

Z_α. Here we use EM to obtain the fixed point iterative update equations. This work is called *probabilistic latent tensor factorization* (PLTF). Chapter 4 and Chapter 5 (model selection) use this approach.

The other methodology does not assume an augmentation tensor in the middle. Here we extend generalized linear models theory to cover the tensor factorization. We call this work as *generalized tensor factorization* (GTF). We then, extend GTF to include simultaneous factorization of multiple observation tensors and call this work as *generalized coupled tensor factorization* (GCTF). Chapter 6 introduces GTF and Chapter 7 extends it for GCTF for coupled factorization.

1.3.4. Contributions

The main contributions of this thesis are as follows.

- *Unified representation of arbitrary tensor factorization models.* In this thesis, we regard vectors, matrices and other higher dimensional objects as simply tensors and hence we regard matrix factorization as tensor factorization. This unified view is based on the same index-based notation. This notation enables us generalization of the factorization to any arbitrary latent structure besides well-known

models such as CP and TUCKER3. We introduce this notation in Chapter 4 with examples for NMF, CP and TUCKER3 [45].

- *Graphical visualization of the factorization models.* A novel representation of tensor factorization models that closely resembles probabilistic graphical models [46] (undirected graphs and factor graphs) enables easy visualization of higher dimensional multiway datasets. In the graph visualization of the tensor factorization, tensor indices become nodes in the graph, factors become cliques (fully connected subgraph). We give examples for graph representations of MF, CP and TUCKER3 in Chapter 4.
- *Practical message passing framework and compact representation.* The link between tensor factorizations and graphical models is more than just visualization aid. The factor update equations turn to inference in a message passing framework composed of marginalization and contraction operations. Identifying this link enables building efficient algorithms and compact representations based on matrix primitives facilitated by our $\Delta(.)$ function. We give examples in Chapter 4 for MF, CP and TUCKER3.
- *Generalization to major class of the cost functions for fixed point update equation.* Exponential dispersion models are first introduced by Tweedie [47] in 1947 (under a different name) and are generalized by Jorgensen [23] in 1986. These models link the variance of the distribution to the mean by so-called variance functions. As a special case, the variance functions can be written in terms of power variance functions for certain major distributions expressed as $Var(x) = \varphi^{-1}\hat{x}^p$ where here φ^{-1} is the distribution specific dispersion parameter for scale, and $\hat{x}$ is the mean of x. The distributions are then indexed by p and we simply set $p = 0, 1, 2, 3$ for the Gaussian, the Poisson, the gamma, and the inverse Gaussian respectively. By use of power variance functions we obtained generic update equations for the factors indexed by p for the distributions. Distribution specific dispersion parameters usually cancel out except when we use different cost functions for coupled factorization models. We use PVFs throughout the thesis explained first in Chapter 3 and used in from Chapter 4 to Chapter 7.

- *Maximum likelihood (ML) estimation framework.* We derive fixed point update equations for maximum likelihood (ML) estimation via expectation maximization (EM) for a large class of statistical models (the so-called exponential dispersion models EDM's). The same fixed point update equation is used for various factorization structures while in the implementation we only replace matrix primitives. The equations are then casted to multiplicative forms as multiplicative update rules (MUR) and alternating least squares (ALS) for the Euclidean cost. We give examples in Chapter 4 [48].
- *A maximum a-posteriori (MAP) estimation framework.* We extend the framework to include prior knowledge via conjugate priors for exponential dispersion models which is a subset of exponential family distributions. This issue is addressed in Chapter 4.
- *A practical implementation technique via matricization and tensorization.* We also sketched a straightforward matricization procedure to convert element-wise equations into the matrix forms to ease implementation and compact representation. The use of the matricization procedure is simple, easy and powerful that without any use of matrix algebra it is possible to derive the update equations mechanically in the corresponding matrix forms. We give examples for matricization procedure in Chapter 4.
- *A model selection framework for factorization models.* Our framework computes model marginal log-likelihood bounds via variational approximation methods for TF models. While we developed the framework practically for KL cost we also outlined generalization for other cost functions. We also get expressions for asymptotic case, i.e. model selection under large samples. We dedicate Chapter 5 for the non-negative model selection issue.
- *Theoretical link between beta divergence and exponential dispersion models.* As already stated before we generalize the cost function to beta divergence which is generated by the Bregman divergence with a suitable function. Beta divergence introduced in [26,27] that unifies Euclidean, KL and IS costs in a single expression [49]. In this thesis we show the link between beta divergence and exponential dispersion models and derive it from the power variance functions directly in

Chapter 3.

- *Extending generalized linear models theory for the tensor factorization.* Generalized linear models theory was first introduced in 1972 [50] for the linear model $g(\hat{X}) = LZ$ where $\hat{X}$ is the random component (expectation of the observation X), L is given model matrix and Z denotes the desired parameters while $g(.)$ is the link function. We extend this theory to include the factorization in the forms $\hat{X} = Z_1 Z_2 \dots Z_{|\alpha|}$ where $Z_{1:|\alpha|}$ are the tensor factors in an alternating update schema. This issue is addressed in Chapter 5 [51].
- *Coupled tensor factorization for arbitrary cost function and arbitrary structures.* We construct a general coupled tensor factorization framework where multiple observation tensors $X_1, X_2, X_{|\nu|}$ are factorized simultaneously. The framework is general to accomplish the factorization with arbitrary structures and large class of cost functions. This issue is addressed in Chapter 7.
- *Handling of missing data for factorization.* Our framework considers partial observations by encoding the missing values into the likelihood equation as in [29,30] and we obtain update equations with missing values in from Chapter 4 to Chapter 7.

1.3.5. Thesis Outline

We start with mathematical background in Chapter 2 where we briefly summarize the related concepts such as exponential family distributions, entropy, expectation-maximization algorithm, model selection, graphical models and tensors.

Chapter 3 is about exponential dispersion models (EDM) [46]. Here we show the link between beta divergence and power variance functions (PVF) of EDMs, and derive the beta divergence from PVF. In this chapter we also relate beta divergence to the deviance. In addition we derive general entropy equations of EDMs. We also compute expectation of posterior distribution of the E-Step of an EM setup with and without conjugate priors. Most of the derivations of this chapter will be referred in the following chapters.

In Chapter 4 we introduce our tensor related notation where we use throughout this thesis. Here we relate the notation hence tensor factorization to graphical models. Following the notation section we use EM to obtain fixed point update equations for latent tensors for KL and Euclidean cost. Also we use conjugate priors during the optimizations and obtain related update equations. Then we generalized the update equations for the β-divergence cost functions. Here we also introduce a matricization technique that turns the element-wise equations into tensor equivalent forms.

Chapter 5 proposes a model selection framework for arbitrary non-negative tensor factorization structures for KL cost via a variational bound on the marginal likelihood. While we will explicitly focus on KL divergence the methodology in this chapter can be extended for other error measures and divergences as well.

In Chapter 6, we derive algorithms for generalized tensor factorization (GTF) by building upon the well-established theory of Generalized Linear Models. Our algorithms are general in the sense that we can compute arbitrary factorizations for a broad class of exponential family distributions including special cases such as Tweedie's distributions corresponding to β-divergences.

Chapter 7 extends the GTF developed in Chapter 6 to the coupled tensor factorization where multiple observation tensors are factorized simultaneously. We illustrate our coupled factorization approach on synthetic data as well as on a musical audio restoration problem.

Chapter 8 is for general discussions and conclusions of this thesis followed by Appendix.

2. MATHEMATICAL BACKGROUND

2.1. Introduction

As already pointed out, one of the main objectives of this thesis is generalization to a large class of cost functions for the tensor factorization. This objective is achieved by the use of exponential dispersion models (EDMs) which is a subset of the (general) exponential family. While EDMs deserve for its own chapter for detail analysis, here in Section 2.2 we start with introducing exponential family of distributions. In addition, in Chapter 4 we use beta divergence as a generalized cost function. While in Chapter 3 we derive beta divergence from power variance functions of exponential dispersion models by using Bregman divergence, before that here in Section 2.2, we introduce Bregman divergence and its duality to the exponential family distributions with examples.

Section 2.3 introduces Expectation Maximization (EM) optimization and related concepts. We use EM method in Chapter 4 where we introduce the PLTF model and obtain fixed point update equations for Euclidean, KL and IS costs.

Section 2.4 is about model selection, Laplace approximation, asymptotic analysis of model selection, i.e. model selection under large number of samples, and Bayesian Information Criterion (BIC). Section 2.5 relates BIC and the gamma distribution. These concepts are used in Chapter 5 where we propose Variational Bayes (VB) based model selection framework for TF models with KL cost function.

Section 2.6 introduces tensors in general while Section 2.7 is about probabilistic graphical models (GM). We recall that our tensor factorization notation is inspired from GMs and we visualize factorization models as the graphical models.

2.2. Exponential Family

An important class of distributions that share a common property is *exponential family of distributions*. Many well-known distributions such as the Gaussian, exponential, gamma, chi-square, beta, Dirichlet, Bernoulli, binomial, multinomial, Poisson can be expressed as exponential family distributions. Regular exponential family distributions are characterized by the cumulant function ψ

$$p(x|\theta) = h(x) \exp\left(t(x)^T\theta - \psi(\theta)\right) \tag{2.1}$$

with $t(x)$ being vector of *sufficient statistic* as $t(x) = [t_1(x), \ldots, t_m(x)]^T$ and θ being vector of *natural (canonical) parameters* as $\theta = [\theta_1, \ldots, \theta_m]^T$ while $h(x)$ being the *base measure* depending only on x. The cumulant generating function ψ (aka. log partition function) provides the normalization of the probability and given as

$$\psi(\theta) = \log \int dx\ h(x) \exp\left(t(x)^T\theta\right) \tag{2.2}$$

whose derivatives are the cumulants of the sufficient statistics $t(x)$. In particular, the first two derivatives

$$\frac{\partial\psi(\theta)}{\partial\theta} = E_\theta[t(x)] \qquad \frac{\partial^2\psi(\theta)}{\partial\theta\partial\theta^T} = Var_\theta[t(x)] \tag{2.3}$$

Before discussing the construction of exponential families we need to explain sufficient statistics. Consider two slots of persistent storage area. Storage A has n slots while Storage B has much less m slots where $m << n$. We have a random i.i.d sample $x_1, \ldots, x_n$ drawn from some probability distribution governed by parameter θ to be stored in the slots. Storage A can have entire sample since it has required capacity while Storage B has to *summarize* the sample by use of some functions t over the sample. If it has only one slot $m = 1$, then one choice can be the average. Indeed, various t functions can be defined as follows where we use $t(x)$ for $t(x_1, \ldots, x_n)$ for

short

$$t_1(x) = \frac{1}{n}\sum_i^n x_i \qquad\qquad t_2(x) = \frac{1}{n}\sum_i^n x_i^2 \tag{2.4}$$

$$t_3(x) = n \qquad\qquad t_4(x) = 3 \tag{2.5}$$

where the last line is irrelevant but completely valid. Then, the likelihood (a function of θ) is desired. By using Storage A we use the entire set $L(\theta) = \Pi_i^n p(x_i|\theta)$. This is not so easy when we use Storage B, since we have only *summary*, i.e. some functions of sample $t_1(x), \ldots, t_m(x)$ that are called as statistics. Question is that what m should be so that we can almost regenerate similar sample as in Storage A. An important property of exponential family is that m is fixed for distributions and independent of n, i.e. as sample size increases, number of sufficient statistics remains the same. Note that this is analogous to the difference between *empirical data* and *theory* of a phenomenon.

An easy way to obtain sufficient statistics is due to *Neyman-Fisher factorization criterion* that connects sufficiency to a factorization of the joint distribution $p(x|\theta)$ which is factorized as

$$p(x|\theta) = g(t_1, \ldots t_m, \theta)h(x) \tag{2.6}$$

where for short we use x for the iid random sample $x_1, \ldots, x_n$, e.g. $t_j(x) = t_j(x_1, \ldots, x_n)$.

The exponential family of distributions arises naturally as the answer to the question: what is the most *uninformative* distribution among all that are consistent with the data. This implies the optimization problem of the functional that provides maximum entropy subject to the data consistency constraints formulated as below [8, 38]. *Empirical expectations* of the sufficient statistics are defined as

$$\hat{\mu}_j = \frac{1}{n}\sum_i^n t_j(x_i) \qquad\qquad \text{for } j = 1, \ldots, m \tag{2.7}$$

where we say $\hat{\mu}_j$ is an empirical expectation for the sufficient statistics $t_j(x)$. Then given a distribution p the expectations of the sufficient statistics under p

$$E_p[t_j(x)] = \int dx\ t_j(x)p(x) \qquad \text{for } j = 1, \ldots, m \tag{2.8}$$

We say that probability distribution p is consistent with the data iff

$$\hat{\mu}_j = E_p[t_j(x)] \tag{2.9}$$

Remark 2.1. In the following sections we will use the symbol μ for the expectation of the sufficient statistics $E_p[t_j(x)]$. Note the difference between $\hat{\mu}$ and μ where the $\hat{\mu}$ is for the empirical expectation whereas μ is expectation under the probability p.

Hence the maximum entropy distribution optimization problem may be formulated as

$$p^* = \arg\max_p H[p] \qquad \text{subject to} \tag{2.10}$$

$$(i) \quad \int dx\ p(x) = 1 \qquad \text{normalizing the density} \tag{2.11}$$

$$(ii) \quad \hat{\mu}_j = E_p[t_j(x)] \qquad j = 1, \ldots, m \tag{2.12}$$

Using the Lagrange multipliers we have the optimization problem

$$p^* = \arg\max_p J(p) \tag{2.13}$$

with respect to functional [38]

$$\begin{aligned} J(p) = &- \int dx\ p(x) \log p(x) + \lambda_0 \left(1 - \int dx\ p(x)\right) \\ &+ \sum_{j=1}^{m} \lambda_j \left(\int dx\ t_j(x)p(x) - \hat{\mu}_j\right) \end{aligned} \tag{2.14}$$

and after taking derivative with respect to $p(x)$ and equating to zero

$$\frac{\partial J(p)}{\partial p(x)} = -\log p(x) - 1 + \lambda_0 + \sum_{j=1}^{m} \lambda_j t_j(x) = 0 \tag{2.15}$$

we end up with the solution [38]

$$p^* = \exp\left(\lambda_0 - 1 + \sum_{j=1}^{m} \lambda_j t_j(x)\right) \tag{2.16}$$

where we may drop the constant terms to find the general form of the distribution that we seek [8]

$$p^* \propto \exp\left(\sum_{j=1}^{m} \lambda_j t_j(x)\right) \tag{2.17}$$

Interestingly the Lagrange multipliers turn to be canonical parameters and we have

$$p^* \propto \exp\left(\sum_{j=1}^{m} \theta_j t_j(x)\right) = \exp\left(t(x)^T \theta\right) \tag{2.18}$$

We list various characteristics of the exponential family distribution in Table 2.1.

Example 2.1. *In this example we obtain cumulant function $\psi(\theta)$ for unknown mean, known variance of the Gaussian distribution by starting from the density of the Gaussian*

$$p(x; a, b) = \exp\left(-\frac{(x-a)^2}{2b} + \frac{1}{2}\log\frac{1}{2\pi b}\right) \tag{2.19}$$

$$= \exp\left(-\frac{x^2}{2b} - \frac{a^2}{2b} + \frac{ax}{b} - \frac{1}{2}\log 2\pi - \frac{1}{2}\log b\right) \tag{2.20}$$

we identify the natural parameter θ as the parameter of the first sufficient statistics x

$$\theta = \frac{a}{b} \tag{2.21}$$

Then we rewrite Equation 2.20 in terms of the natural parameters θ where $a = b\theta$ due to Equation 2.21

$$p(x;\theta,b) = \exp\left(-\frac{x^2}{2b} - \frac{\theta^2 b^2}{2b} + \frac{xb\theta}{b} - \frac{1}{2}\log 2\pi - \frac{1}{2}\log b\right) \tag{2.22}$$

$$= \exp\left(-\frac{x^2}{2b} - \frac{\theta^2 b}{2} + x\theta - \frac{1}{2}\log 2\pi - \frac{1}{2}\log b\right) \tag{2.23}$$

$$= \exp\left(x\theta - \psi(\theta) + \Big(-\frac{x^2}{2b} - \frac{1}{2}\log 2\pi - \frac{1}{2}\log b\Big)\right) \tag{2.24}$$

where

$$\psi(\theta) = \frac{b}{2}\theta^2 \tag{2.25}$$

Table 2.1. Various characteristics of well-known exponential family distributions.

	Sufficient statistics	**Expectations of sufficient statistics**	**Canonical parameters**
Gaussian	$\begin{bmatrix} t_1(x) \\ t_2(x) \end{bmatrix} = \begin{bmatrix} x \\ x^2 \end{bmatrix}$	$\left\langle \begin{bmatrix} t_1(x) \\ t_2(x) \end{bmatrix} \right\rangle = \begin{bmatrix} a \\ a^2 + b \end{bmatrix}$	$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \frac{a}{b} \\ -\frac{1}{2b} \end{bmatrix}$
Poisson	$t_1(x) = x$	$\langle t_1(x) \rangle = a$	$\theta_1 = \log a$
Gamma	$\begin{bmatrix} t_1(x) \\ t_2(x) \end{bmatrix} = \begin{bmatrix} x \\ \log x \end{bmatrix}$	$\left\langle \begin{bmatrix} t_1(x) \\ t_2(x) \end{bmatrix} \right\rangle = \begin{bmatrix} a/b \\ \Psi(a) - \log b \end{bmatrix}$	$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -b \\ a - 1 \end{bmatrix}$
	Base measures	**Cumulant functions**	
Gaussian	$h(x) = (2\pi)^{-\frac{1}{2}}$	$\psi(\theta_1,\theta_2) = -\frac{\theta_1^2}{4\theta_2} - \frac{1}{2}\log(-2\theta_2)$	
Poisson	$h(x) = (x!)^{-1}$	$\psi(\theta_1) = \exp(\theta_1)$	
Gamma	$h(x) = 1$	$\psi(\theta_1,\theta_2) = -(\theta_2+1)\log(-\theta_1) + \log\Gamma(\theta_2+1)$	

Example 2.2. *In this example we obtain cumulant function $\psi(\theta)$ for known mean, unknown variance of the Gaussian. From Equation 2.20 we identify the natural parameter θ as the multiplier of the second sufficient statistic*

$$\theta = -\frac{1}{2b} \tag{2.26}$$

Then we rewrite Equation 2.20 in terms of the natural parameters θ where $b = -\frac{1}{2\theta}$ due to Equation 2.26

$$p(x; a, \theta) = \exp\left(\theta x^2 + \theta a^2 - 2\theta x a - \frac{1}{2}\log 2\pi - \frac{1}{2}\log\theta\right) \tag{2.27}$$

$$= \exp\left((\theta x^2 - 2\theta x a) + \theta a^2 - \frac{1}{2}\log\theta\right) - \frac{1}{2} \tag{2.28}$$

$$= \exp\left([x \;\; x^2][-2\theta a \;\; \theta]^T + \psi(\theta)\right)(2\pi)^{-\frac{1}{2}} \tag{2.29}$$

where

$$\psi(\theta) = \theta a^2 - \frac{1}{2}\log\theta \tag{2.30}$$

We check that the first derivative is to be expectation of the first sufficient statistic where here we have a single sufficient statistic as x^2

$$\frac{\partial\psi(\theta)}{\partial\theta} = a^2 - \frac{1}{2\theta} = a^2 - \frac{1}{-2\frac{1}{2b}} = a^2 + b \tag{2.31}$$

where we plug in Equation 2.26 $\theta = -\frac{1}{2b}$.

Example 2.3. *In the following we find sufficient statistics, canonical parameters and cumulant function for unknown mean, unknown variance of the Gaussian. First we write the density*

$$p(x; a, b) = \exp\left(-\frac{(x-a)^2}{2b} + \frac{1}{2}\log\frac{1}{2\pi b}\right) \tag{2.32}$$

$$= \exp\left(-\frac{x^2}{2b} - \frac{a^2}{2b} + \frac{xa}{b} - \frac{1}{2}\log 2\pi - \frac{1}{2}\log b\right) \tag{2.33}$$

Sufficient statistics are identified as functions of x that are interacting (multiplying) with the parameters (via Neyman-Fisher factorization criteria). Hence, in the density

we have $t_1(x) = x$ *and* $t_2(x) = x^2$ *and*

$$\begin{bmatrix} t_1(x) \\ t_2(x) \end{bmatrix} = \begin{bmatrix} x \\ x^2 \end{bmatrix} \tag{2.34}$$

Canonical parameters are identified as the multipliers of the sufficient statistics implied by the definition

$$\begin{bmatrix} t_1(x) \\ t_2(x) \end{bmatrix}^T \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x \\ x^2 \end{bmatrix}^T \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x \\ x^2 \end{bmatrix}^T \begin{bmatrix} \frac{a}{b} \\ -\frac{1}{2b} \end{bmatrix} \tag{2.35}$$

Thus

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \frac{a}{b} \\ -\frac{1}{2b} \end{bmatrix} \qquad or \qquad \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -\frac{\theta_1}{2\theta_2} \\ -\frac{1}{2\theta_2} \end{bmatrix} \tag{2.36}$$

Then cumulant function is

$$p(x; a, \theta) = \exp\left(\begin{bmatrix} x \\ x^2 \end{bmatrix}^T \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} - \frac{a^2}{2b} - \frac{1}{2}\log 2\pi - \frac{1}{2}\log b \right) \tag{2.37}$$

$$= \exp\left(\begin{bmatrix} x \\ x^2 \end{bmatrix}^T \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \theta_2 \left(\frac{-\theta_1}{2\theta_2} \right)^2 + \frac{1}{2}\log\left(-2\theta_2\right) \right) (2\pi)^{-\frac{1}{2}} \tag{2.38}$$

where $\psi(\theta_1, \theta_2)$ *is left as the remaining terms depending only on canonical parameters* θ_1 *and* θ_2 *as*

$$\psi(\theta_1, \theta_2) = -\frac{\theta_1^2}{4\theta_2} - \frac{1}{2}\log(-2\theta_2) \tag{2.39}$$

h(x) is identified as the terms that do not contain any canonical parameters

$$h(x) = (2\pi)^{-\frac{1}{2}} \tag{2.40}$$

2.2.1. Entropy of Exponential Family Distributions

Entropy is closely related to the dual of the cumulant function denoted by $\phi()$. Recall the formulation for the exponential family of distributions

$$p(x|\theta) = \exp\left(t(x)^T\theta - \psi(\theta) + \log h(x)\right) \tag{2.41}$$

Then entropy for the exponential family of distributions is given as below adapting the notation in [52] appropriately and writing the term $\log h(x)$ explicitly

$$H(\theta) = -\Big(\mu(\theta)^T\theta - \psi(\theta) + \log h(x)\Big) \tag{2.42}$$

where $\mu(\theta)$ is expectation of the sufficient statistics as by definition

$$\mu_j(\theta) = E_\theta[t_j(x)] = \int dx\ p(x|\theta)t_j(x) \qquad\qquad j = 1, \ldots, m \tag{2.43}$$

m being number of sufficient statistics. Alternatively we relate μ and cumulant function ψ as

$$\mu_j(\theta) = \frac{\partial\psi(\theta)}{\partial\theta_j} \tag{2.44}$$

Example 2.4. *The Gaussian distribution has two sufficient statistics. The expected value of the second sufficent statistic $t_2(x) = x^2$ is computed as*

$$\mu_2(\theta) = \frac{\partial\psi(\theta)}{\partial\theta_2} = \frac{\partial\psi(\theta)}{\partial\theta_2} = \frac{\partial}{\partial\theta_2}\left(-\frac{\theta_1^2}{4\theta_2} - \frac{1}{2}\log(-2\theta_2)\right) \tag{2.45}$$

$$= \left(\frac{\theta_1^2}{4\theta_2^2} - \frac{1}{2}\frac{1}{-2\theta_2}(-2)\right) = \left(\frac{\theta_1^2}{4\theta_2^2} - \frac{1}{2\theta_2}\right) \tag{2.46}$$

After replacing canonical parameters with the location and scale parameters where $\theta_1 = a/b$ and $\theta_2 = -1/(2b)$

$$\mu_2 = \langle t_2(x)\rangle = \left\langle x^2\right\rangle = a^2 + b \tag{2.47}$$

Example 2.5. *In this formulation Equation 2.42 the (Shannon) entropy is easily derived as replacing the sufficient statistics by their expectations in the negative log of the density [52, 53]. That is,*

$$\mu = \langle t(x) \rangle = \left\langle \begin{bmatrix} x \\ x^2 \end{bmatrix} \right\rangle = \begin{bmatrix} a \\ a^2 + b \end{bmatrix} \tag{2.48}$$

Hence $\mu^T \theta(\mu)$ *is*

$$\mu^T \theta(\mu) = \langle t(x) \rangle^T \theta(\mu) = \begin{bmatrix} a \\ a^2 + b \end{bmatrix}^T \begin{bmatrix} \frac{a}{b} \\ -\frac{1}{2b} \end{bmatrix} = \frac{a^2}{b} - \frac{a^2}{2b} - \frac{1}{2} \tag{2.49}$$

Next, the Gaussian cumulant function $\psi()$ *is*

$$\psi(\theta_1, \theta_2) = -\frac{\theta_1^2}{4\theta_2} - \frac{1}{2}\log(-2\theta_2) \tag{2.50}$$

after replacing canonical parameters with corresponding expectation parameters

$$\psi(\theta_1(\mu), \theta_2(\mu)) = -\frac{(a/b)^2}{4(-1/(2b))} - \frac{1}{2}\log(-2(-1/(2b))) = \frac{a^2}{2b} + \frac{1}{2}\log b \tag{2.51}$$

Then using Equation 2.42

$$H[X] = \left(\begin{bmatrix} a \\ a^2 + b \end{bmatrix}^T \begin{bmatrix} \frac{a}{b} \\ -\frac{1}{2b} \end{bmatrix} \right) - \left(\frac{a^2}{2b} + \frac{1}{2}\log b \right) + \frac{1}{2}\log 2\pi \tag{2.52}$$

$$= \frac{1}{2}\left(1 + \log 2\pi + \log b\right) \tag{2.53}$$

2.2.2. Relating Dual of Cumulant Function and Entropy

The relationship between the entropy and dual cumulant function is already pointed out by many authors [8]. By definition the dual conjugate of the cumulant

function is given [22]

$$\phi(\mu) = \sup_{\theta \in \Omega} \{\mu^T \theta - \psi(\theta)\} \tag{2.54}$$

$$= \theta(\mu)^T \mu - \psi(\theta(\mu)) \tag{2.55}$$

after that we re-write the entropy in Equation 2.42

$$H(\mu) = -\Big(\phi(\mu) + \log h(x)\Big) \tag{2.56}$$

Hence the dual cumulant function is an important quantity. In the following two examples we show how to derive the dual of cumulant function $\phi()$ for the Gaussian. The first example computes the ϕ in terms of location and scale parameters while the second example computes $\phi(\mu)$ i.e. in terms of the expectation parameter. By using the equality between expectation and location-scale parameters $\mu_1 = a, \mu_2 = a^2 + b$ both results can be converted to each others form.

Example 2.6. *In this example we compute the dual cumulant function $\phi(a, b)$ for the Gaussian where a, b are the location and scale parameters.*

$$\theta(\mu)^T \mu \equiv \begin{bmatrix} \frac{a}{b} \\ -\frac{1}{2b} \end{bmatrix}^T \begin{bmatrix} a \\ a^2 + b \end{bmatrix} = \frac{a^2}{b} - \frac{a^2}{2b} - \frac{1}{2} \tag{2.57}$$

Then by using the dual cumulant function in Equation 2.55

$$\phi(a, b) = \left(\frac{a^2}{b} - \frac{a^2}{2b} - \frac{1}{2}\right) - \left(\frac{a^2}{2b} + \frac{1}{2}\log(b)\right) \tag{2.58}$$

$$= -\frac{1}{2} - \frac{1}{2}\log b \tag{2.59}$$

Example 2.7. *In this example we compute the dual cumulant functions $\phi(\mu)$ for the*

Gaussian. The Gaussian distribution has two sufficient statistics

$$\mu_1(\theta) = \frac{\partial\psi(\theta)}{\partial\theta_1} = \frac{\partial\psi(\theta)}{\partial\theta_1} = \frac{\partial}{\partial\theta_1}\left(-\frac{\theta_1^2}{4\theta_2} - \frac{1}{2}\log(-2\theta_2)\right) = -\frac{\theta_1}{2\theta_2} \tag{2.60}$$

$$\mu_2(\theta) = \frac{\partial\psi(\theta)}{\partial\theta_2} = \frac{\partial\psi(\theta)}{\partial\theta_2} = \frac{\partial}{\partial\theta_2}\left(-\frac{\theta_1^2}{4\theta_2} - \frac{1}{2}\log(-2\theta_2)\right) \tag{2.61}$$

$$= \left(\frac{\theta_1^2}{4\theta_2^2} - \frac{1}{2}\frac{1}{-2\theta_2}(-2)\right) = \left(\frac{\theta_1^2}{4\theta_2^2} - \frac{1}{2\theta_2}\right) \tag{2.62}$$

Now we invert and find $\theta_1(\mu)$ and $\theta_2(\mu)$ (where $\mu = [\mu_1 \quad \mu_2]^T$). Then, after we re-arrange the terms Equation 2.60

$$\theta_2 = -\frac{\theta_1}{2\mu_1} \tag{2.63}$$

and substitute into Equation 2.62

$$\mu_2 = \frac{\theta_1^2}{4(-\frac{\theta_1}{2\mu_1})^2} - \frac{1}{2(-\frac{\theta_1}{2\mu_1})} = \mu_1^2 + \frac{\mu_1}{\theta_1} \tag{2.64}$$

we solve for θ_1

$$\mu_2 - \mu_1^2 = \frac{\mu_1}{\theta_1} \quad \Rightarrow \quad \theta_1 = \frac{\mu_1}{\mu_2 - \mu_1^2} \tag{2.65}$$

and θ_2

$$\theta_2 = -\frac{\theta_1}{2\mu_1} = -\frac{\frac{\mu_1}{\mu_2-\mu_1^2}}{2\mu_1} = -\frac{1}{2(\mu_2 - \mu_1^2)} \tag{2.66}$$

Finally the dual cumulant function $\phi()$ is identified as

$$\phi(\mu) = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}^T \begin{bmatrix} \frac{\mu_1}{\mu_2 - \mu_1^2} \\ -\frac{1}{2(\mu_2 - \mu_1^2)} \end{bmatrix} - \psi(\mu_1, \mu_2) \tag{2.67}$$

$$= \Big(\mu_1(\frac{\mu_1}{\mu_2 - \mu_1^2}) + \mu_2(-\frac{1}{2(\mu_2 - \mu_1^2)})\Big) - \Big(-\frac{\theta_1^2}{4\theta_2} - \frac{1}{2}\log(-2\theta_2)\Big) \tag{2.68}$$

$$= \Big(\frac{\mu_1^2}{\mu_2 - \mu_1^2} - \frac{\mu_2}{2(\mu_2 - \mu_1^2)}\Big) - \Big(-\frac{\mu_1^2}{-2(\mu_2 - \mu_1^2)} + \frac{1}{2}\log(\mu_2 - \mu_1^2)\Big) \tag{2.69}$$

$$= -\frac{1}{2} - \frac{1}{2}\log(\mu_2 - \mu_1^2) \tag{2.70}$$

2.2.3. Bregman Divergence

Definition 2.1. *Let ϕ be a convex function. The Bregman divergence $d_\phi(x, y)$ is defined as*

$$d_\phi(x, y) = \phi(x) - \phi(y) - \nabla\phi(y)(x - y) \tag{2.71}$$

with $\nabla\phi(y)$ being the gradient vector of ϕ evaluated at y [22].

The Bregman divergence is a non-negative quantity as $d_\phi(x, y) \geq 0$. It is zero iff $x = y$, i.e. $d_\phi(x, x) = 0$. Note that Bregman divergence is not a metric since it does not provide neither symmetry nor triangular inequality. An interesting point is that Bregman divergence $d_\phi(x, y)$ is equal to tail of first-order Taylor expansion of $\phi(x)$ at y

$$\phi(x) = \phi(y) + \nabla\phi(y)(x - y) + d_\phi(x, y) \tag{2.72}$$

which turns to the Bregman divergence as in Equation 2.71.

In the following examples we assume x and y are scalar variables and ϕ' is the derivative.

Example 2.8. *For Euclidean distance* $\phi(x) = \frac{1}{2}x^2$ *with the derivative* $\phi'(y) = y$

$$d_\phi(x, y) = \frac{1}{2}x^2 - \frac{1}{2}y^2 - (x - y)y = \frac{1}{2}(x - y)^2 \tag{2.73}$$

Example 2.9. *For the KL divergence* $\phi(x) = x \log x$ *with the derivative* $\phi'(y) = \log y + 1$

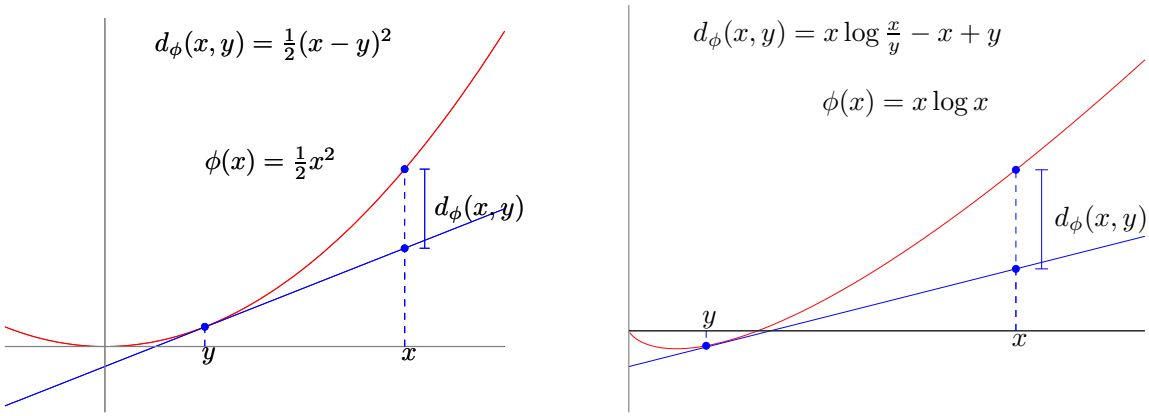
$$d_\phi(x, y) = x \log x - y \log y - (x - y)(\log y + 1) \tag{2.74}$$

$$= x \log \frac{x}{y} - x + y \tag{2.75}$$

Example 2.10. *For the IS divergence* $\phi(x) = -\log x$ *with the derivative* $\phi'(y) = -\frac{1}{y}$

$$d_\phi(x, y) = -\log x + \log y - (x - y)(-\frac{1}{y}) \tag{2.76}$$

$$= \frac{x}{y} - \log \frac{x}{y} - 1 \tag{2.77}$$



(a) (b)

Figure 2.1. Bregman divergence illustration; (a) for Euclidean distance, (b) for KL divergence.

2.2.4. Bijection between Exponential Family and Bregman Divergence

We recall that exponential family of distributions is defined as

$$p(x|\theta) = h(x) \exp\big(t(x)^T\theta - \psi(\theta)\big) \tag{2.78}$$

Banerjee *et al.* proved the bijection between regular exponential family distributions and the Bregman divergence [22] by taking log of both sides and then adding and subtracting $\phi(t(x))$

$$\begin{aligned}
\log p(x|\theta) &= \log h(x) + \big(t(x)^T\theta - \psi(\theta)\big) && (2.79)\\
&= \big\{\log h(x) + \phi\big(t(x)\big)\big\} - \big\{-t(x)^T\theta + \psi(\theta) + \phi\big(t(x)\big)\big\} && (2.80)\\
&= \log b_\phi(x) - d_\phi\big(t(x), \mu\big) && (2.81)
\end{aligned}$$

where an important observation is that $b_\phi(x)$ is independent of μ as

$$b_\phi(x) = \exp(\phi(t(x)))h(x) \tag{2.82}$$

where $h(x)$ is the base measure. Here ψ and ϕ are conjugate dual functions while μ as *expectation parameter* and θ as *canonical parameters* are dual parameters. Recall that conjugate dual function ϕ to ψ is defined as

$$\begin{aligned}
\phi(\mu) &= \sup_{\theta\in\Omega}\{\mu^T\theta - \psi(\theta)\} && (2.83)\\
&= \theta(\mu)^T\mu - \psi(\theta(\mu)) && (2.84)
\end{aligned}$$

Here are some important properties [22].

Table 2.2. Various identities and functions for Gaussian, Poisson and exponential distributions [22] with the parametrization as $\mathcal{N}(x; a, b), \mathcal{PO}(x; a), \mathcal{E}(x; a)$. The Gaussian variance parameter is assumed to be known. Note that here we have a single sufficient statistics as $t(x) = x$ and μ is expected value of the sufficient statistics $t(x)$.

	θ	$\mu = \langle t(x) \rangle$	$\theta(\mu)$	$\mu(\theta)$
Gaussian	$\frac{a}{b}$	a	$\frac{\mu}{b}$	θb
Poisson	$\log a$	a	$\log \mu$	$\exp \theta$
Exponential	$-a$	$\frac{1}{a}$	$-\frac{1}{\mu}$	$-\frac{1}{\theta}$
	$\psi(\theta)$	$\psi(\theta(\mu))$	$\phi(\mu)$	$\phi(\mu) = \theta(\mu)\mu - \psi(\theta(\mu))$
Gaussian	$\frac{b}{2}\theta^2$	$\frac{1}{2}\frac{\mu^2}{b}$	$\frac{1}{2}\frac{1}{b}\mu^2$	$\frac{\mu}{b} \times \mu - \frac{1}{2}\frac{\mu^2}{b} = \frac{1}{2}\frac{\mu^2}{b}$
Poisson	$\exp(\theta)$	μ	$\mu \log \mu - \mu$	$\log \mu \times \mu - \mu = \mu \log \mu - \mu$
Exponential	$-\log(-\theta)$	$\log \mu$	$-\log \mu - 1$	$\frac{-1}{\mu} \times \mu - \log \mu = -1 - \log \mu$

(i) Let ϕ and ψ are convex conjugates

$$d_\phi(x, \mu) = \phi(x) + \psi(\theta(\mu)) - x\theta(\mu) \tag{2.85}$$

(ii) (Dual Divergence) Let μ be convex conjugate of θ, then

$$d_\phi(\mu_1, \mu_2) = \phi(\mu_1) + \psi(\theta_2) - \mu_1\theta_2 = d_\psi(\theta_2, \theta_1) \tag{2.86}$$

(iii) Forward mapping from canonical parameter θ to expectation parameter μ

$$\mu(\theta) = \frac{\partial \psi(\theta)}{\partial \theta} \qquad\qquad \theta(\mu) = \frac{\partial \phi(\mu)}{\partial \mu} \tag{2.87}$$

(iv) Derivatives of the convex conjugates are related as follows where f^{-1} is the inverse function

$$\frac{\partial \phi(\mu)}{\partial \mu} = \left(\frac{\partial \psi(\theta)}{\partial \theta} \right)^{-1} \qquad \text{hence} \qquad (\mu(\theta))^{-1} = \theta(\mu) \tag{2.88}$$

Recall that for $f(x) = y$ and $f^{-1}(y) = x$ we have

$$\frac{\partial f(x)}{\partial x} = \frac{1}{\frac{\partial f^{-1}(y)}{\partial y}} \tag{2.89}$$

2.2.5. Conjugate Priors for Exponential Family

Every member of exponential family distributions has a conjugate prior which is also a member of the family. The benefit of the conjugate priors is that when a conjugate prior is multiplied by the corresponding likelihood function, the resulting posterior distribution has the same functional form as the prior. This makes the Bayesian analysis, i.e. computing a posterior distribution over parameters and MAP estimate, i.e. computing a point estimate of the parameters that maximizes their posterior easier. Let's consider the following example without going into canonical format.

Example 2.11. *The conjugate prior to the Poisson observation is the gamma distribution as*

$$\textit{gamma posterior} \propto \textit{ Poisson likelihood} \times \textit{ gamma prior} \tag{2.90}$$

$$p(z|x, \theta) \propto p(x|z)p(z|\theta) \tag{2.91}$$

where the distributions are

$$p(x|z) = \mathcal{P}(x; z) \qquad \textit{and} \qquad p(z|\theta) = \mathcal{G}(z; a, b) \tag{2.92}$$

Then the log (conjugate) prior and log likelihood are

$$\log p(x|z) = x \log z - z - \log x! \tag{2.93}$$

$$\log p(z|\theta) = (a - 1) \log z - \frac{1}{b} z - \log \Gamma(a) + a \log b \tag{2.94}$$

If we have n independent Poisson observation with the following log likelihood

$$\sum_i^n \log p(x_i|z) = \sum_i^n (x_i \log z - z - \log x_i!) \tag{2.95}$$

$$= \log z \sum_i^n x_i - nz - \sum_i^n \log x_i! \tag{2.96}$$

Then we add them

$$\log p(x|z) + \log p(z|\theta) = \left(\sum_i^n x_i + a - 1\right) \log z - \left(\frac{1}{b} + n\right) z + const \tag{2.97}$$

For n observations the posterior distribution is

$$p(z|x,\theta) = \mathcal{G}(z|\alpha,\beta) \qquad where \qquad \alpha = a + \sum_i^n x_i \qquad \beta = \left(\frac{1}{b} + n\right)^{-1} \tag{2.98}$$

where for $n = 0$ we recover the original gamma prior distribution.

Remark 2.2. We recall that as n the number of observations gets larger the shape of the posterior distribution gets closer to that of the Gaussian distribution. This phenomenon is also known as *asymptotic normality*. Therefore, in such cases, the Gamma posterior can be approximated by the Gaussian. This is one of the consequences of the *central limit theorem*. Figure 2.2 illustrates the asymptotic normality.

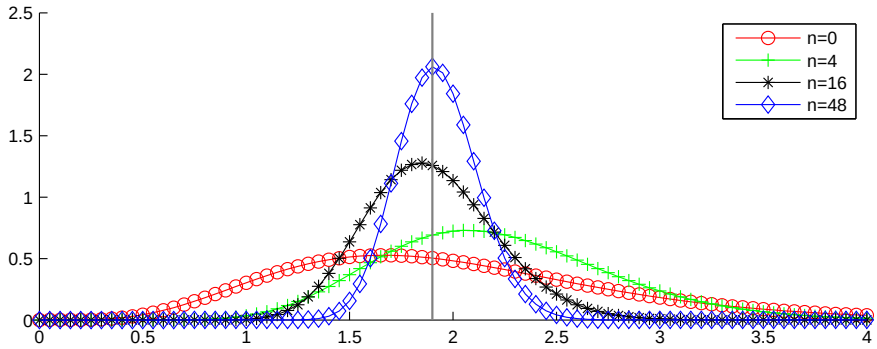


Figure 2.2. Posterior distribution peaks around MAP estimate and turns to be a Gaussian distribution as the number of observations n increases.

An exponential family of distribution and its conjugate prior have the following functional form [37]

$$p(x|\theta) = h(x)\exp\left(t(x)^T\theta - \psi(\theta)\right) \tag{2.99}$$

$$p(\theta|\tau_0,\tau_1) = g(\tau_0,\tau_1)\exp(\tau_1^T\theta - \tau_0\psi(\theta)) \tag{2.100}$$

with τ_0, τ_1 being the hyperparameters where τ_0 is interpreted as the effective number of pseudo-observations in the prior by [1,54]. Note that τ_0 is the coefficient of the term that has the same form as the log partition function $\psi(\theta)$ of the observation model. Here the function $g(\tau_0,\tau_1)$ is the normalization coefficient and disappears during optimization.

Example 2.12. *Poisson distribution is associated with KL cost function and its conjugate prior is the gamma distribution. Then given that*

$$\mathcal{PO}(x;\lambda) = \exp\left(x\log\lambda - \lambda - \log x!\right) \tag{2.101}$$

$$\mathcal{G}(\lambda;a,b) = \exp\left((a-1)\log\lambda - b\lambda + a\log b - \log\Gamma(a)\right) \tag{2.102}$$

where we want to relate (or tie) the hyperparameters a and b to the τ_0 and τ_1. That is, we want to match Equation 2.100 and Equation 2.102. The clue is that the cumulant function ψ in Equation 2.100 is to be equal to that of Equation 2.99, i.e. for this example that of Poisson. The Poisson in canonical format is

$$\mathcal{PO}(x;\theta) = h(x)\exp\left(x\theta - \psi(\theta)\right) \tag{2.103}$$

where recall that the cumulant function for Poisson is $\psi(\theta) = \exp\theta$ and to obtain $\exp\theta$ term in Equation 2.102 we re-parametrize the gamma by substituting $u = \log\theta$ as

$$u = \log\theta \qquad \Rightarrow \qquad \theta = \exp u \tag{2.104}$$

$$\mathcal{G}(u;a,b) = \exp\left((a-1)u - b\exp(u) + a\log b - \log\Gamma(a)\right) \tag{2.105}$$

where we identify and tie the parameters as

$$\tau_0 = b \qquad \tau_1 = a - 1 \tag{2.106}$$

2.3. Expectation Maximization (EM)

The Expectation Maximization (EM) algorithm is an iterative optimization method for finding Maximum Likelihood estimate of the parameters of the underlying distribution. The EM was discovered and employed independently by many researchers until Dempster *et al.* [55] brought their ideas together under the name *EM Algorithms* and proved convergence. It is specifically suitable when some parameters are completely or partially hidden. Its convergence is guaranteed to a local maximum on the likelihood surface. For notational convenience, let the log-likelihood function $\mathcal{L}(\theta)$ is defined as

$$\mathcal{L}(\theta) = \mathcal{L}(x|\theta) = \log p(x|\theta) \tag{2.107}$$

where X is observed data and θ is unknown parameter vector. Our goal is to find θ that maximizes $\mathcal{L}(\theta)$ such that

$$\theta^* = \arg\max_{\theta} p(x|\theta) \tag{2.108}$$

However, suppose $p(x|\theta)$ is hard to compute, while $p(x, z|\theta)$ is easy where Z is the latent variable. Then EM cycles through two steps, in E step we compute the expectations of the hidden structure whereas in M step we maximize $\mathcal{L}(\theta)$ with respect to (w.r.t.) θ given the expected value in E step as

$$\text{(E Step)} \qquad z^{(n)} = \langle Z \rangle_{p(z|x,\theta^{(n)})} \tag{2.109}$$

$$\text{(M Step)} \qquad \theta^{(n+1)} = \arg\max_{\theta} \left\langle \log p(x, z^{(n)}|\theta) \right\rangle \tag{2.110}$$

where convergence is guaranteed such that

$$\mathcal{L}(\theta^{(1)}) \leq \mathcal{L}(\theta^{(2)}) \leq \ldots \mathcal{L}(\theta^{(n)}) \tag{2.111}$$

An interesting analysis [56] shows that the EM algorithm is a hill-climbing in the log likelihood $\log p(x|\theta)$ by use of a *functional* [2] $\mathcal{B}(q,\theta)$ where q is an *averaging distribution*. The idea is based on the fact that rather than maximizing directly the complete log likelihood due to the randomness imposed by the hidden Z we work on its expected value with an averaging distribution. Indeed let the functional $\mathcal{B}(q,\theta)$ be a lower bound on the true incomplete (marginal) log likelihood

$$\mathcal{L}(\theta) = \log p(x|\theta) \geq \mathcal{B}(q,\theta) = \int dz\; q(z) \log \frac{p(x,z|\theta)}{q(z)} \tag{2.112}$$

noting that $\int dz\; q(z) = 1$ we have

$$\mathcal{L}(\theta) = \log p(x|\theta) = \int dz\; q(z) \log p(x|\theta) = \int dz\; q(z) \log \frac{p(x,z|\theta)}{p(z|x,\theta)} \tag{2.113}$$

$$= \int dz\; q(z) \log \frac{p(x,z|\theta)}{q(z)} \frac{q(z)}{p(z|x,\theta)} \tag{2.114}$$

$$= \int dz\; q(z) \log \frac{p(x,z|\theta)}{q(z)} - \int dz\; q(z) \log \frac{p(z|x,\theta)}{q(z)} \tag{2.115}$$

$$= \int dz\; q(z) \log \frac{p(x,z|\theta)}{q(z)} + KL[q(z)||p(z|x,\theta)] \tag{2.116}$$

Hence we have the equality [1, 56]

$$\mathcal{L}(\theta) = \mathcal{B}(q,\theta) + KL[q(z)||p(z|x,\theta)] \tag{2.117}$$

with the following remarks;

- Since KL is a positive quantity we have the lower bound $\mathcal{B}$ as $\mathcal{L}(\theta) \geq \mathcal{B}(q,\theta)$.

[2] The functional is related to variational calculus [46] and convex analysis. As an example, entropy function $H[p]$ is a functional. We refer to [1] for the functional.

- KL is zero iff $q(z) = p(z|x, \theta)$, i.e. q is equal to the true posterior $p(z|x, \theta)$. Then $\mathcal{L}(\theta) = \mathcal{B}(q, \theta)$.
- In statistical physics, the (negative) bound $\mathcal{B}$ is known as *variational free energy* where physical states are represented by the values of latent Z while energy of a state is $-\log p(x, z|\theta)$ [56]. Note that the bound can be decomposed as

$$\mathcal{B}(q, \theta) = \langle \log p(x, z|\theta) \rangle_{q(z)} + H[q] \tag{2.118}$$

- By factorizing the expectation of the joint distribution $\langle \log p(x, z|\theta) \rangle$ in Equation 2.118 the bound $\mathcal{B}(q, \theta)$ is written as

$$\mathcal{B}(q, \theta) = \langle \log p(x|z, \theta) \rangle_{q(z)} + \langle \log p(z|\theta) \rangle_{q(z)} + H[q(z)] \tag{2.119}$$

$$= \langle \log p(x|z, \theta) \rangle_{q(z)} - KL[q(z)||p(z|\theta)] \tag{2.120}$$

  By using this fact the log likelihood can be written in terms of two KL divergence as

$$\mathcal{L}(\theta) = \langle \log p(x|z, \theta) \rangle_{q(z)} - KL[q(z)||p(z|\theta)] + KL[q(z)||p(z|x, \theta)] \tag{2.121}$$

  For model selection for tensor factorization we will use this expression by assuming the posterior $q(z) = p(z|x, \theta)$ and this results to

$$\mathcal{L}(\theta) = \langle \log p(x|z, \theta) \rangle_{p(z|x,\theta)} - KL[p(z|x, \theta)||p(z|\theta)] \tag{2.122}$$

- For the true posterior, i.e. $q(z) = p(z|x, \theta)$ we have the equality since KL term in Equation 2.117 becomes zero

$$\mathcal{L}(\theta) = \langle \log p(x|z, \theta) \rangle_{p(z|x,\theta)} + \langle \log p(z|\theta) \rangle_{p(z|x,\theta)} + H[p(z|x, \theta)] \tag{2.123}$$

  and in terms of KL it is as follows where we will use this result in Chapter 5

related to model selection

$$\mathcal{L}(\theta) = \langle \log p(x|z,\theta) \rangle_{p(z|x,\theta)} - KL[p(z|x,\theta)||p(z|\theta)] \tag{2.124}$$

Now, we may re-write EM iterations as

$$\text{(E Step)} \qquad q^{(n+1)} = \arg\max_{q} \mathcal{B}(q, \theta^{(n)}) \tag{2.125}$$

$$\text{(M Step)} \qquad \theta^{(n+1)} = \arg\max_{\theta} \mathcal{B}(q^{(n+1)}, \theta) \tag{2.126}$$

where in this view, in E step θ is fixed and we search for the $q(z)$ that maximizes the bound $\mathcal{B}$. It can be shown that the choice of the posterior of Z as $q^{(n+1)}(z) = p(z|x,\theta^{(n)})$ gets the maximum as

$$q(z)^* = \arg\max_{q} \mathcal{B}(q,\theta) + \lambda(1 - \int dz\ q(z)) \tag{2.127}$$

$$= \arg\max_{q} \log p(x|\theta) + \int dz\ q(z) \log p(z|x,\theta) \tag{2.128}$$

$$- \int dz\ q(z) \log q(z) + \lambda(1 - \int dz\ q(z)) \tag{2.129}$$

where we added a Lagrange multiplier due to the constraint $\int dz\ q(z) = 1$ [57]. Then taking derivative w.r.t. $q(z)$ and solving for q after equating to zero

$$\frac{\partial \mathcal{B}(q,\theta)}{\partial q(z)} = 0 + \log p(z|x,\theta) - (1 \times \log q(z) + q(z) \times 1/q(z)) + \lambda = 0 \tag{2.130}$$

$$= \log p(z|x,\theta) - \log q(z) + \lambda - 1 = 0 \tag{2.131}$$

then ignoring the constant terms $\lambda - 1$ we obtain [56, 57]

$$q(z)^* = p(z|x,\theta) \tag{2.132}$$

Whenever q is found, we fix it and we search for θ in M step so that it maximizes $\mathcal{B}(q^{(n+1)},\theta)$ and hence maximizes the log likelihood.

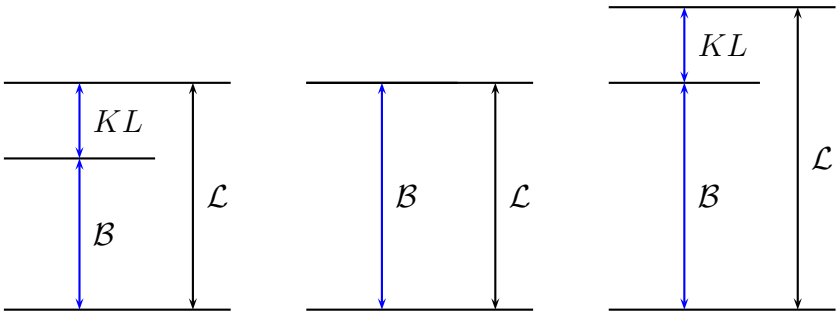


Figure 2.3. EM iterates between E step and M step.

2.4. Bayesian Model Selection

For a Bayesian point of view, a model is associated with a random variable m and it interacts with the observed data X simply as $p(m|x) \propto p(x|m)p(m)$. Then, for a model selection task we choose the model associated with m^* having the highest posterior probability such that $m^* = \arg\max_m p(m|x)$. Meanwhile, assuming the model priors $p(m)$ are equal the quantity $p(x|m)$ becomes important since comparing $p(m|x)$ is the same as comparing $p(x|m)$. The quantity $p(x|m)$ is called *marginal likelihood* [1] and it is the average over the space of the parameters, in our case, Z as

$$p(x|m) = \int dz\, p(x|z,m)p(z|m) \tag{2.133}$$

On the other hand, computation of this integral is itself a difficult task that requires averaging on several models and parameters. There are several approximation methods such as sampling or deterministic approximations such as Gaussian approximation. One other approximation method is to bound the log marginal likelihood by using variational inference [1,40] where an *approximating distribution* q is introduced into the log marginal likelihood equation as in Equation 2.116 where the bound attains its maximum and becomes equal to the log marginal likelihood whenever $q(z)$ is set as $p(z|x,m)$, that is the exact posterior distribution. However, the posterior is usually intractable, and we introduce approximating distributions that are easier to compute. One other approximation technique is the *Laplace approximation* of the posterior distribution by

a Gaussian centered at the MAP estimate.

2.4.1. Asymptotic Analysis

Recall that we have a log likelihood equation in terms of the expectations w.r.t. true posterior

$$\mathcal{L}(\theta) = \langle \log p(x|z,\theta) \rangle_{p(z|x,\theta)} + \langle \log p(z|\theta) \rangle_{p(z|x,\theta)} + H[p(z|x,\theta)] \tag{2.134}$$

On the other hand, if we have large number of observation $X = X_1, X_2, \ldots X_n$, the posterior $p(z|x,\theta)$ is peaked at around the mode z_{map}. In other words, as n gets larger, the posterior distribution $p(z|x,\theta)$ first turns to be the Gaussian (also called asymptotic normality) and then turns to be delta dirac such that we may approximately write $p(Z = z_{map}|x,\theta) \simeq 1$ and $p(Z \neq z_{map}|x,\theta) \simeq 0$. By using this approximation as $n \to \infty$ we may drop the integral terms from the expectations

$$\langle \log p(x|z,\theta) \rangle_{p(z|x,\theta)} = \int dz \ \log p(x|z,\theta) p(z|x,\theta) \simeq \log p(x|z_{map},\theta) \tag{2.135}$$

$$\langle \log p(z|\theta) \rangle_{p(z|x,\theta)} = \int dz \ \log p(z|\theta) p(z|x,\theta) \simeq \log p(z_{map}|\theta) \tag{2.136}$$

since only for $Z = z_{map}$ we have $p(Z = z_{map}|x,\theta) = 1$ and zero for $Z \neq z_{map}$. For the entropy term in Equation 2.134, it is the entropy of peaked posterior distribution which turns to Gaussian as number of observations gets larger. Hence it is the entropy of the multivariate Gaussian distribution for $Z = Z_1, Z_2, \ldots, Z_m$ as follows where K is the determinant of the m dimensional covariance matrix

$$H[Z] = H[p(z|x,\theta)] = \frac{1}{2} \log((2\pi e)^m |K|) \tag{2.137}$$

As a result we have the following lemma:

Lemma 2.1. *Log marginal likelihood given in Equation 2.134 is approximated asymp-*

totically as

$$\mathcal{L}(\theta) \simeq \log p(x|z_{map}, \theta) + \log p(z_{map}|\theta) + \frac{1}{2}\log((2\pi e)^m |K|) \tag{2.138}$$

with m being the number of the dimension and $|K|$ being the determinant of the m-dimensional covariance matrix.

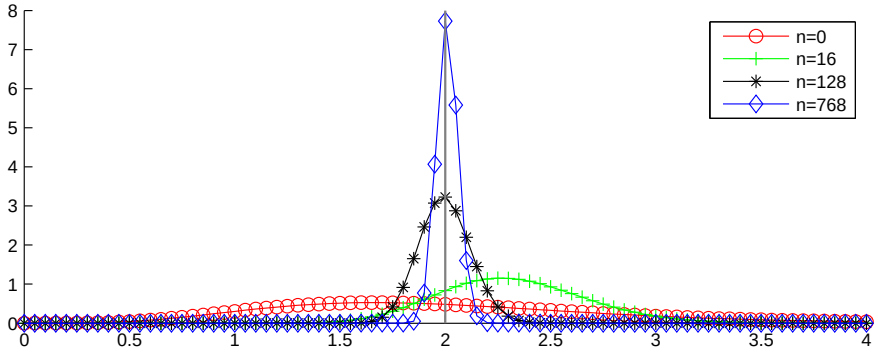


Figure 2.4. Asymptotically as $n \to \infty$ posterior distribution peaks around MAP estimate and turns to be first a Gaussian then delta Dirac function.

Remark 2.3. Equation 2.138 is equivalent to the *Laplace approximation* [1] defined below

$$\mathcal{L}(\theta) \simeq \log p(x|z_{map}, \theta) + \log p(z_{map}|\theta) + \frac{m}{2}\log(2\pi) - \frac{1}{2}\log|A| \tag{2.139}$$

where A is $m \times m$ Hessian matrix of the second derivative of negative log posterior and given as [1]

$$A = -\partial^2 \log p(x|z_{map}, \theta) \log p(z_{map}|\theta) = -\partial^2 \log p(z_{map}|x, \theta)) \tag{2.140}$$

Laplace approximation is obtained by expanding the log posterior (which is approximately Gaussian under large sample assumption) by the second order Taylor

expansion around the mode z_{map} [1, 58]

$$\log p(z|x,\theta) \simeq \log p(z_{map}|x,\theta) + \frac{\partial \log p(z|x,\theta)}{\partial z}\Big|_{z=z_{map}}(z - z_{map}) + \tag{2.141}$$

$$\frac{1}{2}(z - z_{map})\frac{\partial^2 \log p(z|x,\theta)}{\partial z^2}\Big|_{z=z_{map}}(z - z_{map})^T \tag{2.142}$$

$$= \log p(z_{map}|x,\theta) - \frac{1}{2}(z - z_{map})|A|(z - z_{map})^T \tag{2.143}$$

where $A = \partial^2 \log p(z|x,\theta)|_{z=z_{map}}$. Note that the first derivative $\partial \log p(z|x,\theta)$ becomes zero since z_{map} is the local maximum and also the second derivative is negative. Here this equation approximates the posterior distribution by *Laplace approximation*. Also, rising both sides to the power of exp shows that posterior distribution approximate to the Gaussian as follows

$$p(z|x,\theta) \simeq \log p(z_{map}|x,\theta) \exp\Big(- \frac{1}{2}(z - z_{map})|A|(z - z_{map})^T\Big) \tag{2.144}$$

The quantity that we want to approximate is log marginal likelihood $\mathcal{L}(\theta)$. To obtain this noting that the posterior distribution is proportional to the product of the likelihood and the prior (Bayes rule)

$$p(z|x,\theta) \propto p(x|z,\theta)p(z|\theta) \tag{2.145}$$

$$p(z_{map}|x,\theta) \propto p(x|z_{map},\theta)p(z_{map}|\theta) \tag{2.146}$$

and then equating two equations of the posterior distributions, i.e. Equation 2.143 and Equation 2.145

$$p(x|z,\theta)p(z|\theta) \simeq \log p(z_{map}|x,\theta) \exp\Big(- \frac{1}{2}(z - z_{map})|A|(z - z_{map})^T\Big) \tag{2.147}$$

$$\simeq p(x|z_{map},\theta)p(z_{map}|\theta) \exp\Big(- \frac{1}{2}(z - z_{map})|A|(z - z_{map})^T\Big) \tag{2.148}$$

and finally integration over z by using the Gaussian integral [59] we end up with the log marginal likelihood approximation

$$\sum_z p(x|z,\theta)p(z|\theta) \simeq \sum_z p(x|z_{map},\theta)p(z_{map}|\theta)\exp\Big(-\frac{1}{2}(z-z_{map})|A|(z-z_{map})^T\Big)$$

$$p(x|\theta) \simeq p(x|z_{map},\theta)p(z_{map}|\theta)\sum_z \exp\Big(-\frac{1}{2}(z-z_{map})|A|(z-z_{map})^T\Big)$$

$$\mathcal{L}(\theta) = \log p(x|\theta) \simeq \log p(x|z_{map},\theta) + \log p(z_{map}|\theta)\log\Big((2\pi)^{m/2}|A|^{-1/2}\Big)$$

Finally an important observation is that the Hessian matrix of the second derivative of negative log posterior denoted by A is equivalent to the covariance matrix denoted by K where this is closely related with the fact that the second derivative of the cumulant generating function is the variance. We refer to the section on Exponential Family of distributions (hence to the canonical representation) for further details.

2.4.2. Bayesian Information Criterion (BIC)

Here we want to drop terms that don't scale with number of samples n in log-likelihood approximation in Equation 2.138.

Lemma 2.2. *Let n be the number of the samples as $X = X_1, \ldots, X_n$ and the latent parameters be m-dimensional as $Z = Z_1, \ldots, Z_m$. As n gets larger the maximum entropy of the posterior $Z \sim p(z|x,\theta)$ can be approximated as*

$$H[Z] = \frac{1}{2}\log((2\pi e)^m|K|) \simeq -\frac{1}{2}m\log n \tag{2.149}$$

Proof. We already know that the posterior $p(z|x,\theta)$ turns to be Gaussian as n grows while the dimension m is kept as constant. The entropy terms becomes

$$\lim_{n\to\infty} H[Z] = \frac{1}{2}\log(|K|) \tag{2.150}$$

where K is the diagonal matrix as

$$K_{m\times m} = \begin{bmatrix} \frac{\Sigma_1}{n} & 0 & \dots & 0 \\ 0 & \frac{\Sigma_2}{n} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \frac{\Sigma_m}{n} \end{bmatrix}$$

with Σ_i being the prior and $\frac{\Sigma_i}{n}$ being the posterior variance of Z_i. Note that Z_is are assumed to be independent but not identically distributed and therefore they may have distinct prior variances, while the independence assumption implies non-diagonal elements of K being zero, i.e. $K_{ij} = 0$ for $i \neq j$. Then, the determinant of the diagonal matrix K is

$$|K| = \prod_i^m \frac{\Sigma_i}{n} \tag{2.151}$$

Plug it in as follows where we drop the constant term which is $const = \frac{1}{2}\sum_i^m \log \Sigma_i$ that does not scale with n

$$\lim_{n\to\infty} H[Z] = \frac{1}{2}\log(|K|) = \frac{1}{2}\log(\prod_i^m \frac{\Sigma_i}{n}) = \frac{1}{2}\log(\prod_i^m \frac{1}{n}) + const \tag{2.152}$$

$$= -\frac{1}{2} m \log n \tag{2.153}$$

□

Lemma 2.3. *BIC is recovered as we drop terms that do not scale with the number of data n in Equation 2.138*

$$\mathcal{L}(\theta) = \mathcal{L}(x|\theta) \simeq \log p(x|z_{map}, \theta) - \frac{1}{2} m \log n \tag{2.154}$$

noting that $\log p(z_{map}|\theta)$ in Equation 2.138 does not scale with n and hence is dropped out.

2.5. Bayesian Information Criterion (BIC) for Gamma Distribution

The main motivation of this section is to check the applicability of BIC for the gamma distributed variables. BIC assumes asymptotic normality under large sample, but the question is that how well this approximation applies to the gamma distributed data. We will answer this question here and will use the result of this section in Chapter 7 for the model selection.

2.5.1. Stirling's Formula for Gamma Function

The gamma function is approximated by the Stirling's formula as

$$\Gamma(\alpha) \simeq (\frac{2\pi}{\alpha})^{1/2} e^{-\alpha} \alpha^{\alpha} \tag{2.155}$$

For sufficiently large α, after dropping the constant term the log of the gamma function becomes

$$\log \Gamma(\alpha) \simeq -\frac{1}{2} \log \alpha - \alpha + \alpha \log \alpha \tag{2.156}$$

The derivative of $\log \Gamma(\alpha)$ is of a special interest for the gamma distribution and denoted by Ψ function

$$\Psi(\alpha) = \frac{\partial \log \Gamma(\alpha)}{\partial \alpha} \simeq -\frac{1}{2\alpha} - 1 + \log \alpha + \alpha \frac{1}{\alpha} = -\frac{1}{2\alpha} + \log \alpha \tag{2.157}$$

such that as α gets larger $\Psi(\alpha)$ becomes asymptotically

$$\lim_{\alpha \to \infty} \Psi(\alpha) = \log \alpha \tag{2.158}$$

2.5.2. Asymptotic Entropy of Gamma Distribution

For the given convention of the gamma distribution,

$$Z_i \sim \mathcal{G}(Z_i;\alpha,\beta) = \exp((\alpha-1)\log z_i - \frac{1}{\beta}z_i - \log\Gamma(\alpha) + \alpha\log\beta) \tag{2.159}$$

the entropy is

$$H[Z_i] = \log\Gamma(\alpha) - (\alpha-1)\Psi(\alpha) + \log\beta + \alpha \tag{2.160}$$

Note that the sign of $\log\beta$ term depends on the convention of gamma distribution. However, it will not affect the following analysis. It is known that for sufficiently large shape parameter α, gamma distribution can be approximated as Gaussian distribution. Here we will not use the Gaussian approximation and continue with the gamma prior, but instead we will use String's approximation and compare the result with that of the Gaussian approximation of the posterior. In our case, as we derived in Equation 2.98, let a and b be the shape and scale parameters of the gamma prior, i.e. $Z \sim p(z_i|\theta) = \mathcal{G}(Z_i|a,b)$ then we have a posterior distribution $p(z_i|x,\theta)$ in gamma form with n observations

$$p(z_i|x,\theta) = \mathcal{G}(Z_i|\alpha,\beta) \qquad \text{where } \alpha = a + \sum_j^n X_j \quad \beta = (\frac{1}{b}+n)^{-1} \tag{2.161}$$

hence here we may assume a large shape parameter α augmented with the observations X_j. Then, we may use Stirling's approximation in place of gamma function and logarithm in place of digamma function in the entropy as

$$H[Z_i] \simeq \left(-\frac{1}{2}\log\alpha - \alpha + \alpha\log\alpha\right) - (\alpha-1)\log(\alpha) + \log\beta + \alpha \tag{2.162}$$

$$= \frac{1}{2}\log\alpha + \log\beta = \frac{1}{2}\log\alpha\beta^2 \tag{2.163}$$

Interestingly the term inside the log is the variance of gamma distribution and asymptotic (large n) entropy term here has the same shape as that of the Gaussian in

Equation 2.149. If we use the other convention of gamma distribution, i.e. $p(z_i|.) = \exp((\alpha-1)z_i - \beta z_i + \ldots)$, we again end up with the variance of the distribution as $H[Z_i] = \frac{1}{2}\log\frac{\alpha}{\beta^2}$. Furthermore we derive the BIC complexity term (except the term for the number of free parameters) when we plug in the posterior variance $\frac{\Sigma_i}{n}$ and drop the constants as

$$H[Z_i] \simeq \frac{1}{2}\log\alpha\beta^2 = \frac{1}{2}\log\frac{\Sigma_i}{n} \tag{2.164}$$

$$\simeq -\frac{1}{2}\log n \tag{2.165}$$

recalling that $Z = Z_1, \ldots, Z_m$. Noting also that each Z_i is independent but not identically distributed the total approximated entropy becomes

$$H[Z] = H[Z_1, \ldots, Z_m] \simeq m\left(-\frac{1}{2}\log n\right) = -\frac{1}{2}m\log n \tag{2.166}$$

Remark 2.4. We also note that plugging α and β in the posterior variance equation

$$Var = \alpha\beta^2 = (a + \sum_j^n X_j)((\frac{1}{b} + n)^{-1})^2 \tag{2.167}$$

and let X_a be the maximum of X_j's, then as n gets larger the variance approximates to

$$Var = nX_a n^{-2} = \frac{X_a}{n} \tag{2.168}$$

Plugging it in Equation 2.163 and dropping the constant term $\log X_a$ ends up with the same equation Equation 2.166.

The result of this section is that in a model selection framework, BIC information criterion can be used for the Possion likelihood-gamma prior data. As penalty term

we may approximately use the posterior variance as

$$Penalty = \sum_{i}^{m} \frac{1}{2} \log \alpha_i \beta_i^2 \tag{2.169}$$

$$= \frac{1}{2} m \log n \tag{2.170}$$

where at the last equation we further approximate for large n.

2.6. Tensor Factorization Models

A *tensor* is simply an array with many dimensions. An observer recording temperature of a point with certain altitude, location and date of a year, would need a three dimensional object, which is a cube with three axes as altitude, location and date, whose cells are temperature values. More formally, this cube is called as three-way tensor or the third-order tensor and it has three indices. A vector is, then, first-order tensor while a matrix is a second-order tensor. The order of a tensor is the number of the dimensions also known as *modes* or *ways* [12, 13]. An N-way tensor is denoted as $X \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$. Back to the temperature cube, it is denoted as $X \in \mathbb{R}^{|a| \times |l| \times |t|}$ where the symbol $|.|$ denotes the cardinality of the indices a, l, t for altitude, location and date. One specific element, i.e. a temperature value, is then denoted by either $X(a, l, t)$ or equivalently by $X^{a,l,t}$.

A *slice* of a tensor is a matrix obtained by fixing all but any two modes. For the temperature cube $X^{*,*,1}$, for example, could denote the temperature matrix over all altitude-location combinations for January, $X^{*,*,2}$ is for February and so on. A three-way array, i.e. a cube can have frontal, lateral and horizontal slices. A *fiber* of a tensor is, similar to slice, is a vector obtained by fixing all but any single mode. For the temperature cube $X^{a,l,*}$, for example, could denote the temperature vector recorded on a certain location index l at a certain altitude a. A cube can have row, column and tube fibers. *mode-n* fibers are vectors obtained by fixing all the modes except mode-n. A tensor can be transformed into and thus represented as a matrix by replacing and reordering its fibers as matrix. This process is called as *matricization*

or *unfolding* whereas we prefer the term unfolding and reserve the term *matricization*. *mode-n unfolding* of a tensor results a matrix and is denoted by $X_{(n)}$. Matrix $X_{(n)}$ is simply formed by replacing the mode-n fibers of tensor to be column vectors [12, 13].

Example 2.13. *Let two frontal slices of* $X \in \mathbb{R}^{2\times2\times2}$ *be as*

$$X^{*,*,1} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \quad X^{*,*,2} = \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix}$$

*Then, mode-*1*, mode-*2 *and mode-*3 *unfolding are*

$$X_{(1)} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{bmatrix} \quad X_{(2)} = \begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{bmatrix} \quad X_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

An N-way *rank one* tensor $X \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ is created by outer product of N vectors as

$$X = v_1 \times v_2 \ldots \times v_N \tag{2.171}$$

where the symbol $\times$ is for vector outer product. In 1927 Hitchcock [15] proposed expressing a tensor as sum of finite number of rank-one tensors as

$$X = \sum_r v_{r,1} \times v_{r,2} \ldots \times v_{r,N} \tag{2.172}$$

which as an example for $N = 3$, i.e. as 3-way tensor, it can be expressed as

$$X^{i,j,k} = \sum_r A^{i,r} B^{j,r} C^{k,r} \tag{2.173}$$

This special decomposition is discovered and named by many researchers independently such as CANDECOMP (canonical decomposition) [17] and PARAFAC (parallel factors) [18] where Kiers simply called it as CP [12]. The interesting and useful property of CP is that it ends up with unique factors up to scaling and permutation.

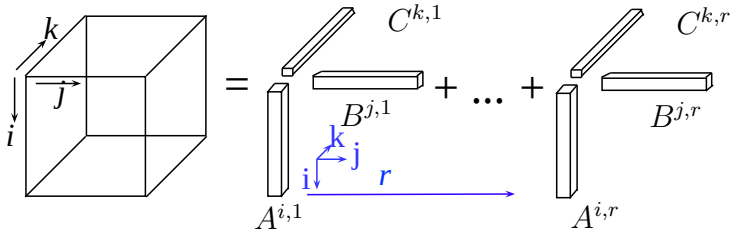


Figure 2.5. CP factorization of 3-way tensor as sum of r rank-one tensors.

CP decomposition can be expressed as Khati-Rao product of *factor matrices* and unfolding operation where Khati-Rao product is defined as follows

Definition 2.2. *Let b_j be a column vector. Khati-Rao product is defined as*

$$A \odot B = \begin{bmatrix} a_{11}\boldsymbol{b_1} & a_{12}\boldsymbol{b_2} & \dots & a_{1n}\boldsymbol{b_n} \\ a_{21}\boldsymbol{b_1} & a_{22}\boldsymbol{b_2} & \dots & a_{2n}\boldsymbol{b_n} \\ \dots & \dots & \dots & \\ a_{m1}\boldsymbol{b_1} & a_{m2}\boldsymbol{b_2} & \dots & a_{mn}\boldsymbol{b_n} \end{bmatrix} \quad \textit{where} \quad \boldsymbol{b_j} = \begin{bmatrix} b_{1j} \\ b_{2j} \\ \dots \\ b_{kj} \end{bmatrix}$$

A related matrix product that we will use for higher factorization models is Kronecker product defined as follows

Definition 2.3. *Kronecker product*

$$A \otimes B = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \dots & a_{1n}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \dots & a_{2n}\boldsymbol{B} \\ \dots & \dots & \dots & \\ a_{m1}\boldsymbol{B} & a_{m2}\boldsymbol{B} & \dots & a_{mn}\boldsymbol{B} \end{bmatrix}$$

Example 2.14. *Let Y and Z be two matrices as*

$$Y = \begin{bmatrix} 9 & 11 \\ 10 & 12 \end{bmatrix} \quad Z = \begin{bmatrix} 13 & 15 \\ 14 & 16 \end{bmatrix}$$

Then, Khati-Rao product and Kronecker product are as follows

$$Y \odot Z = \begin{bmatrix} 9 \times \begin{bmatrix} 13 \\ 14 \end{bmatrix} & 11 \times \begin{bmatrix} 15 \\ 16 \end{bmatrix} \\ 10 \times \begin{bmatrix} 13 \\ 14 \end{bmatrix} & 12 \times \begin{bmatrix} 15 \\ 16 \end{bmatrix} \end{bmatrix} \qquad Y \otimes Z = \begin{bmatrix} 9 \times \begin{bmatrix} 13 & 15 \\ 14 & 16 \end{bmatrix} & 11 \times \begin{bmatrix} 13 & 15 \\ 14 & 16 \end{bmatrix} \\ 10 \times \begin{bmatrix} 13 & 15 \\ 14 & 16 \end{bmatrix} & 12 \times \begin{bmatrix} 13 & 15 \\ 14 & 16 \end{bmatrix} \end{bmatrix}$$

Then CP factorization is expressed as mode-1 unfolding

$$X_{(1)} = A(C \odot B)^T \tag{2.174}$$

In 1963, Tucker introduced a factorization which resembled high order PCI or SVD for the tensors [16]. It summarizes given tensor X into *core tensor* G which has the same number of modes as X. G is considered to be a compressed version of X. Then, for each mode there is a *factor matrix* which may be orthogonal. Element-wise notation for TUCKER3 is given as follows

$$X^{i,j,k} = \sum_{pqr} G^{p,q,r} A^{i,p} B^{j,q} C^{k,r} \tag{2.175}$$

Here is in matrix form [13]

$$X_{(1)} = AG_{(1)}(C \otimes B)^T \qquad \text{mode-1 unfolding} \tag{2.176}$$

$$X_{(2)} = BG_{(2)}(C \otimes A)^T \qquad \text{mode-2 unfolding} \tag{2.177}$$

$$X_{(3)} = CG_{(3)}(B \otimes A)^T \qquad \text{mode-3 unfolding} \tag{2.178}$$

TUCKER factorization has two more variants as TUCKER1 and TUCKER2

$$TUCKER2 \qquad X^{i,j,k} = \sum_{pqr} G^{p,q,r} A^{i,p} B^{j,q} I^{k,k} \qquad k = r \tag{2.179}$$

$$TUCKER1 \qquad X^{i,j,k} = \sum_{pqr} G^{p,q,r} A^{i,p} I^{j,j} I^{k,k} \qquad j = q, k = r \tag{2.180}$$

where for TUCKER2 or TUCKER1 we do not decrease number of modes of G, neither we lessen the number of mode matrices. What we do is to plug in identity matrix I for the disappearing mode.

There are a number of methods to update (learn) the latent structure. One of them which we briefly explain is *alternating least square* (ALS) which simply updates latent matrices one by one by fixing all but one. ALS minimizes the error as squared Euclidean distance

$$\min_{\hat{X}} \|X - \hat{X}\|_F \qquad \text{where } \hat{X}_{(1)} = A G_{(1)} (C \otimes B)^T \tag{2.181}$$

where $\|A\|_F$ is the Frobenius norm of the matrix A where for $m \times n$ matrix it is defined as

$$\|A\|_F = \sqrt{\sum_{i}^{m} \sum_{j}^{n} |a_{ij}|^2} \tag{2.182}$$

Here the model estimate $\hat{X}_{(1)}$ is computed after each factor is computed as below (we only give updates for the factor matrix A and core tensor G whereas updates for B and C are similar)

$$A = X_{(1)} \left(G_{(1)} (C \otimes B)^T \right)^\dagger \tag{2.183}$$

$$G_{(1)} = A^\dagger X_{(1)} \left((C \otimes B)^T \right)^\dagger \tag{2.184}$$

where $X^\dagger = X^T (X X^T)^{-1}$ is for the pseudoinverse (Moore-Penrose inverse).

Table 2.3. Element-wise and matrix representations of well-known tensor models. The symbols $\otimes$, $\odot$ and $\circ$ are for Knonecker, Khatri-Rao and Hadamard products in the order and the division is of the element-wise type.

Model	**Element-wise**	**Matrix**
MF	$\hat{X}^{i,j} = \sum_r A^{i,r} B^{j,r}$	$\hat{X} = AB^T$
CP	$\hat{X}^{i,j,k} = \sum_r A^{i,r} B^{j,r} C^{k,r}$	$\hat{X}_{(1)} = A(C \odot B)^T$
TUCKER3	$\hat{X}^{i,j,k} = \sum_{pqr} G^{p,q,r} A^{i,p} B^{j,q} C^{k,r}$	$\hat{X}_{(1)} = AG_{(1)}(C \otimes B)^T$
Model	**ALS for certain components**	
MF	$A = \hat{X}B^{\dagger}$	
CP	$A = \hat{X}_{(1)}((C \odot B)^T)^{\dagger}$	
TUCKER3	$A = \hat{X}_{(1)} \left(G_{(1)}(C \otimes B)^T\right)^{\dagger}$	$G_{(1)} = A^{\dagger}\hat{X}_{(1)} \left((C \otimes B)^T\right)^{\dagger}$

2.7. Graphical Models

A *graphical model* (GM) is simply a graph associated with a joint probability distribution. What makes a graph to be a model is that we associate a random variable with each vertex. Depending on the problem domain, the random variable may be discrete or continuous. Notationally, let V be a set of vertices (also called nodes) and E be a set of edges between any pair of vertices. A graph G is formed by the sets V and E as $G = (V, E)$. We may have unordered set of edges that forms an undirected graph or ordered set edge set that forms a directed graph. The interpretation of the existence and non-existence of the edges between vertices (nodes) is directly related with probabilistic dependence and independence relations between the random variables.

State space of a model with discrete variables is the all possible configurations (states) of the joint distribution, i.e. a state space is all possible states that a GM can represent. For example, a GM with n vertices with binary (two valued) variables has 2^n possible states.

Associated with a GM there are certain tasks such as structure learning, param-

eter learning, inference and/or belief propagation. Structure Learning is constructing the graph from samples and parameter learning is learning of the parameters of the distributions associated with each vertex. An *evidence* is simply a partial observation of the variables. Given the evidence computation of the probability distribution of the unobserved variables is defined as *inference*. This computation can be carried out either exactly called as exact inference or approximately called as *approximate inference* . On the other hand, approximate inference is a sampling technique (usually Gibb's sampler) where given the evidence we make samples of the unobserved random variables under interest and take averages.

To deal with the tasks a researcher may come up with different representations of the graphical models such as *undirected graphical model*, and yet one other *directed acyclic graphs* (DAG) where no cycles are allowed. In a DAG the directed edges have a notion of parent-child relationships among the vertices and is shown such that $Pa(X)$ is set of all the vertices that are parents of the vertex X, i.e. there are directed links from $Pa(X)$ to X. The relationship between the parents and the child vertex is *Markovian* expressed in the form of *conditional independence* such that the variable Y has no influence on X given (or knowing) the parents of X, i.e. $Pa(X)$, (assuming no other indirect or direct edges between X and Y) as follows

$$p(X|Pa(X), Y) = p(X|Pa(X)) \tag{2.185}$$

This Markovian relation is also shown as $X \perp\!\!\!\perp Y|Pa(X)$. Then the joint distribution is factorized as

$$p(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} p(X_i|Pa(X_i)) \tag{2.186}$$

Example 2.15. *Consider the sprinkler example whose DAG given in Figure 2.6 where*

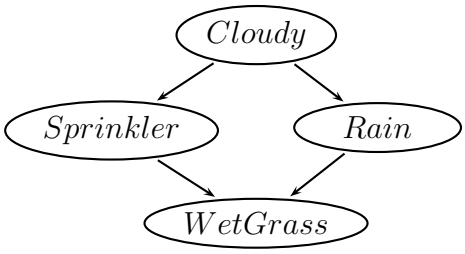


Figure 2.6. Sprinkler graphical model. The model is completed by the probability tables for $p(C), p(S|C), p(R|C), p(W|S,R)$.

the vertex set and edge set are as

$$V = \{C, S, R, W\} \tag{2.187}$$

$$E = \{(C,S), (C,R), (S,W), (R,W)\} \tag{2.188}$$

The DAG encodes the joint probability distribution that is factorized as

$$p(C,S,R,W) = p(C)\ p(S|C)\ p(R|C,S)\ p(W|C,S,R) \tag{2.189}$$

According to the graph by using the conditional independence, the joint distribution can be simplified

$$p(C,S,R,W) = p(C)\ p(S|C)\ p(R|C)\ p(W|S,R) \tag{2.190}$$

Exact inference on a DAG is shown to be NP-hard [60]. To achieve exact inference in polynomial time a DAG can be converted to a tree called *junction tree* (JT). An example taken from [61] for the construction of JT from a DAG is given in Figure 2.7.

2.8. Summary

In this chapter, various background materials essential for this thesis have been reviewed. Here is a summary of the main equations. The entropy of exponential family

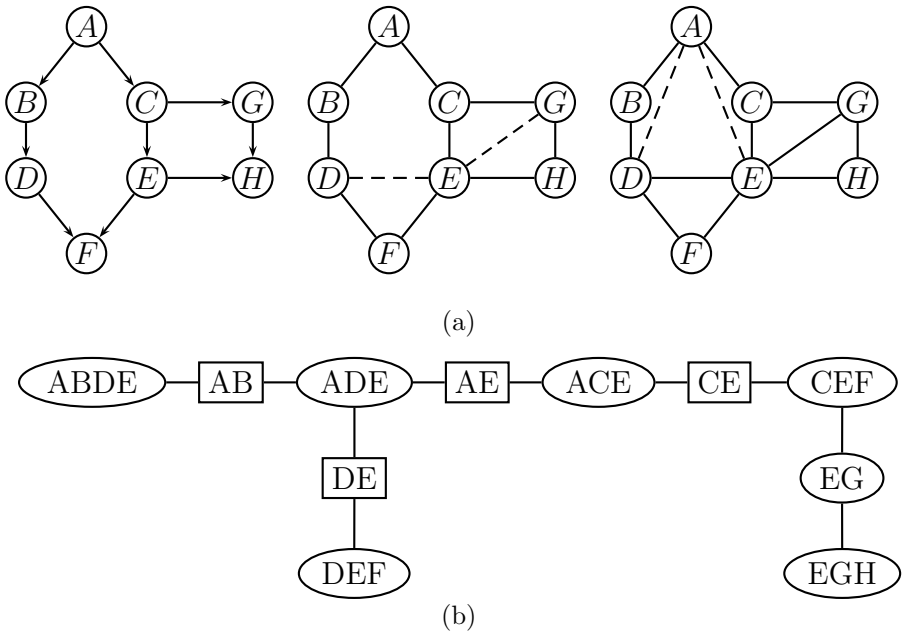


Figure 2.7. (a) illustrates conversion of a DAG to a moral and then to a triangulated graph, (b) represents the junction tree obtained from the triangulated graph. The example is from [61].

distributions is listed as

$$H(\mu) = -\Big(\phi(\mu) + \log h(x)\Big) \tag{2.191}$$

$$H(\theta) = -\Big(\mu(\theta)^T\theta - \psi(\theta) + \log h(x)\Big) \tag{2.192}$$

Next, regular exponential family distributions and the Bregman divergence are shown to be tied to each other [22]

$$\log p(x|\theta) = \log b_\phi(x) - d_\phi\big(t(x), \mu\big) \tag{2.193}$$

An exponential family of distribution and its conjugate prior have the following functional form [37] sharing a common cumulant function $\psi(\theta)$

$$p(x|\theta) = h(x)\exp\big(t(x)^T\theta - \psi(\theta)\big) \tag{2.194}$$

$$p(\theta|\tau_0, \tau_1) = g(\tau_0, \tau_1)\exp(\tau_1^T\theta - \tau_0\psi(\theta)) \tag{2.195}$$

The log-likelihood bound is given as

$$\mathcal{L}(\theta) = \mathcal{B}(q, \theta) + KL[q(z)||p(z|x, \theta)] \tag{2.196}$$

which can also be written in terms of two KL divergence as

$$\mathcal{L}(\theta) = \langle \log p(x|z, \theta) \rangle_{q(z)} - KL[q(z)||p(z|\theta)] + KL[q(z)||p(z|x, \theta)] \tag{2.197}$$

Finally we showed that the asymptotic entropy for m-dim variable Z is expressed as

$$\lim_{n \to \infty} H[Z] = -\frac{1}{2} m \log n \tag{2.198}$$

which is equivalent to the penalty term of the criterion BIC. Here n is number of observations. Then we used this expression for the entropy of gamma distribution under large sample

$$\mathcal{L}(\theta) = \mathcal{L}(x|\theta) \simeq \log p(x|z_{map}, \theta) - \frac{1}{2} m \log n \tag{2.199}$$

Finally we showed that the entropy of posterior identified as the gamma distribution under large sample case is approximated as

$$H[Z_i] \simeq \frac{1}{2} \log \alpha \beta^2 \tag{2.200}$$

where α, β are the shape and scale parameters.

3. EXPONENTIAL DISPERSION MODELS IN TENSOR FACTORIZATION

3.1. Introduction

Generalization of cost functions for the tensor factorization is one of our main objective in this thesis and it is based on the *Exponential Dispersion Models* (EDM). EDMs are subset of the exponential family of distributions while their history goes back to Tweedie's unnoticed work in 1947 [24, 47] such that in 1972 Nelder and Wedderburn published GLMs [50] without any references to Tweedie's work where their error distribution formulation was similar to Tweedie's formulation. In 1982 Morris used the term *Natural Exponential Models* (NEF) [62], and 1987 Jorgensen [23] generalized the ideas and named as EDMs. The main difference between EDM and exponential family is that EDMs are linear exponential family of distributions where the sufficient statistics function is simply identity as $t(x) = x$. In addition, a new parameter called as dispersion parameter is introduced which is analogous to the scale parameter. The mean parameter and the dispersion parameter of EDMs can be viewed as location and scale parameters, and therefore EDMs are similar to location-scale models.

This chapter is organized as follows. Section 3.2 briefly reviews the EDMs. In Section 3.3 we show how to derive alpha and beta divergences from *power variance functions* (PVF) of EDMs, hence we link alpha and beta divergences and the EDM theory. Section 3.4 shows the equivalence of the beta divergence and deviance theory and discusses the use of goodness of fit for the beta divergence. In this section we also derive the general entropy equation for EDMs. Section 3.5 derives equations for ML and MAP optimizations for EDMs; we will use these results in Section 3.6 for ML/MAP optimization for the augmentation models such as $x = \sum_i s_i$. In Chapter 4 we will apply the results of Section 3.6 to the tensors for generalization. Finally Section 3.7 is about non-linear models where we obtain fixed point update rules for the latent factors under the assumption of matching link and loss functions.

3.2. Background: Exponential Dispersion Models

Exponential Dispersion Models (EDM) are subset of the exponential family of distributions defined as

$$p(x|\theta,\varphi) = h(x,\varphi)exp\left\{\varphi\left(\theta x - \psi(\theta)\right)\right\} \tag{3.1}$$

where θ is the *canonical (natural) parameter*, φ^{-1} is the *dispersion parameter* and ψ is the cumulant generating function. Here, $h(x,\varphi)$ is the *base measure* and is independent of the canonical parameter. The mean parameter (also called *expectation parameter*) is denoted by μ and is tied to the canonical parameter θ with the differential equations

$$\mu = \mu(\theta) = \frac{\partial\psi(\theta)}{\partial\theta} \tag{3.2}$$

$$\theta = \theta(\mu) = \frac{\partial\phi(\mu)}{\partial\mu} \tag{3.3}$$

where $\phi(\mu)$ is the conjugate dual of $\psi(\theta)$ just as the canonical parameter θ is conjugate dual of expectation parameter μ. The relationship between θ and μ is more direct and given as [23]

$$\frac{\partial\theta}{\partial\mu} = v(\mu)^{-1} \tag{3.4}$$

where $v(\mu)$ being the *variance function* [23–25]. Variance function, on the other hand, is equal to the variance of the distribution scaled by the dispersion parameter φ^{-1} as

$$Var(x) = \varphi^{-1}v(\mu) \tag{3.5}$$

A useful assumption is that the variance function is in the form of power function and therefore it is called as *power variance function* (PVF) [23, 25] given as

$$v(\mu) = \mu^p \tag{3.6}$$

where the variance is, then,

$$Var(x) = \varphi^{-1}\mu^p \tag{3.7}$$

Thus, the dispersion models relate the variance of a distribution to some power p of its mean μ. Table 3.1 shows PVFs for various well known distributions.

Remark 3.1. Note that the parametrization in PVF is arbitrary, and one can set the exponent as

$$v(\mu) = \mu^{2-p} \tag{3.8}$$

where in this case the Gaussian, the Poisson, the gamma and the inverse Gaussian distributions correspond to $p = 2, 1, 0, -1$ in the order.

Table 3.1. Variance functions and related characteristics of some EDM distributions. Note that parametrization of the gamma distribution is such that its variance is μ^2/k.

	p	**Dispersion Param.**	**Variance Function**	**Variance**
Gaussian	0	σ^2	1	σ^2
Poisson	1	1	μ	μ
Gamma	2	$1/k$	μ^2	μ^2/k
Inv. Gaussian	3	σ^2	μ^3	$\sigma^2\mu^3$

3.3. From PVF to Beta and Alpha Divergence

In this section we show how to derive alpha and beta divergence from *power variance functions* (PVF) of *exponential dispersion models* (EDM) and/or of *natural exponential families* (NEF). Starting from the PVF first we obtain the dual (conjugate) of cumulant function by solving the differential equations; next, we use Bregman divergence to obtain beta divergence. Then the same dual of cumulant function is used to obtain alpha divergence when applied to f-divergence as pointed out by [63]. Changing

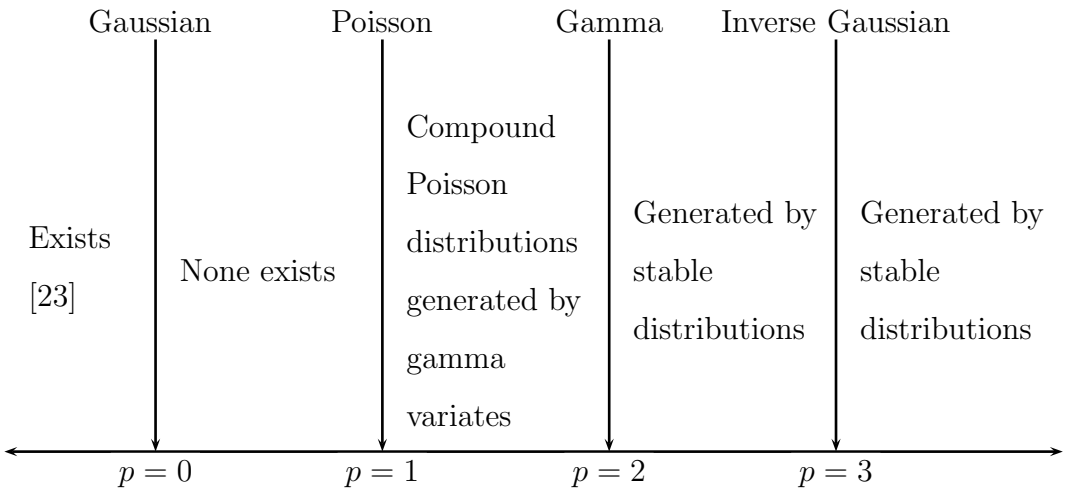


Figure 3.1. Classification of EDM distributions. Adapted from [25].

the parametrization of PVF, we come up with different parametrization of the beta and alpha divergences.

3.3.1. Derivation of Conjugate (Dual) of Cumulant Function

We first obtain the canonical parameter by solving the differential equation $\frac{\partial\theta}{\partial\mu} = \mu^{-p}$ where here m is a constant

$$\int \partial\theta = \int \mu^{-p}\partial\mu \qquad \Rightarrow \qquad \theta = \theta(\mu) = (1-p)^{-1}\mu^{1-p} + m \tag{3.9}$$

Then we find dual cumulant function $\phi(.)$ by integrating Equation 3.3 and using $\theta(\mu)$ given in Equation 3.9

$$\phi(\mu) = \int \theta(\mu)\partial\mu \tag{3.10}$$

$$= \int \left((1-p)^{-1}\mu^{1-p} + m\right)\partial\mu \tag{3.11}$$

$$= \left((1-p)(2-p)\right)^{-1}\mu^{2-p} + m\mu + d \tag{3.12}$$

This function is not defined for $p = 1$ and $p = 2$, since then for these two cases we seek the limits. Consider choosing the constants m and d appropriately as

$$m = -(1-p)^{-1} \qquad d = (2-p)^{-1} \tag{3.13}$$

so that the dual cumulant function becomes

$$\phi(\mu) = ((1-p)(2-p))^{-1} \left(\mu^{2-p} - (2-p)\mu + (1-p)\right) \tag{3.14}$$

In this way we can write the dual cumulant function as a ratio of two functions with the free variable p

$$\phi(p) = \frac{g_1(p)}{g_2(p)} = \frac{\mu^{2-p} - (2-p)\mu + (1-p)}{(1-p)(2-p)} \tag{3.15}$$

The fact that for $p = 1$ and $p = 2$ both $g_1(p)$ and $g_2(p)$ become zero, enable us to use the l'Hôpital's rule for seeking the limits. Here, the appropriate choice of the constants m and d enables $g_1()$ to provide this property. Then taking the derivatives of $g_1()$ and $g_2()$ w.r.t. p we obtain

$$\frac{g_1'(p)}{g_2'(p)} = \frac{-\mu^{2-p}\log\mu + \mu - 1}{2p-3} \tag{3.16}$$

Finally, plugging in $p = 1$ and $p = 2$ as well as the general case $p \neq 1, 2$ we obtain general dual cumulant function

$$\phi(\mu) = \begin{cases} \big((1-p)(2-p)\big)^{-1}\mu^{2-p} + m\mu + d & \text{for } p \neq 1, 2 \\ \mu\log\mu - \mu + 1 & \text{for } p = 1 \\ -\log\mu + \mu - 1 & \text{for } p = 2 \end{cases} \tag{3.17}$$

Note that with the constant values $m = -1/(1-p)$ and $d = 1/(2-p)$ and equating the index parameter $2-p = \beta$ this function exactly matches to [32] and also to [64,65].

To check this result note that inverse of the second derivative is equal to the

variance function (VF) where this result is expected since

$$\frac{\partial^2 \phi(\mu)}{\partial \mu^2} = \frac{\partial}{\partial \mu}\frac{\partial \phi(\mu)}{\partial \mu} = \frac{\partial}{\partial \mu}\theta = \mu^{-p} \tag{3.18}$$

while in the following analysis we do not need the second derivative. We only use the first derivative of $\phi(\mu)$ to obtain beta and alpha divergence, which is as follows

$$\frac{\partial \phi(\mu)}{\partial \mu} = \theta(\mu) = (1-p)^{-1}\mu^{1-p} + m \tag{3.19}$$

3.3.2. Beta divergence

In this section we obtain the beta divergence between x and y by using the definition of the Bregman divergence for $\phi()$ obtained in the previous section as

$$d_\phi(x,y) = \phi(x) - \phi(y) - (x-y)\frac{\partial \phi(y)}{\partial y} \tag{3.20}$$

After substituting the related equalities we obtain

$$\begin{aligned} d_\phi(x,y) &= \left((1-p)(2-p)\right)^{-1} x^{2-p} + mx + d && (3.21)\\ &\quad - \left((1-p)(2-p)\right)^{-1} y^{2-p} - my - d && (3.22)\\ &\quad - (x-y)\left\{(1-p)^{-1}y^{1-p} + m\right\} && (3.23)\\ &= \left((1-p)(2-p)\right)^{-1} x^{2-p} && (3.24)\\ &\quad - \left((1-p)(2-p)\right)^{-1} y^{2-p} && (3.25)\\ &\quad - (x-y)(1-p)^{-1}y^{1-p} && (3.26) \end{aligned}$$

which is equal to the beta divergence proposed by [27,28] (also related to density power divergence of [26])

$$d_\beta(x,y) = \left((1-p)(2-p)\right)^{-1}\left\{x^{2-p} - y^{2-p} - (x-y)(2-p)y^{1-p}\right\} \tag{3.27}$$

As the special cases, this expression reduces to the Euclidean distance for $p = 0$ while for $p = 1, 2$ it reduces to the KL and IS divergences in the limits

$$d_\beta(x, y) = \begin{cases} \frac{1}{2}(x-y)^2 & p = 0 \\ x \log \frac{x}{y} + y - x & p = 1 \\ \frac{x}{y} - \log \frac{x}{y} - 1 & p = 2 \end{cases} \tag{3.28}$$

Here we note that some authors [31, 63] have already used some variants of the dual cumulant function $\phi()$ in Equation 3.12 to obtain beta divergence as a special case of the Bregman divergence. Even [63] has used the same function to obtain alpha divergence as a special case of the f-divergence. However, the synthesis of this function and its link to the exponential dispersion models, to our knowledge, is shown first in this thesis.

Remark 3.2. We remark that even in case of ignoring the initial conditions, i.e. setting $m = d = 0$ and obtaining ϕ as follows

$$\phi(\mu) = ((1-p)(2-p))^{-1} \mu^{2-p} \tag{3.29}$$

We still obtain beta divergence since the remaining terms are cancelled smoothly by the third term of the Bregman divergence. Interestingly, the distribution induced by the dual of cumulant function ϕ is independent of the constants, i.e. initial conditions m and d. Indeed, [25] re-parametrized the natural (canonical) parameter from θ to $\theta_1 = \theta + m$ in Equation 3.9 that changes $\psi(\theta_1) = \psi(\theta + m)$ and we use $\psi(\theta)$ and θ rather than $\psi(\theta_1)$ and θ_1. This is also confirmed by [23] as class of distributions induced does not depend on the initial conditions. This leads to the theorem that a *linear exponential family* distribution is characterized by its PVF [23] .

Remark 3.3. The definition of index parameter p in power variance function is somewhat arbitrary. Thus we can equally define the variance function as

$$v(\mu) = \frac{\partial \mu}{\partial \theta} = \mu^{2-p} \tag{3.30}$$

and come up with the beta divergence with different index parameter as

$$d_\beta(x,y) = (p(p-1))^{-1}\left\{x^p - y^p - p(x-y)y^{p-1}\right\} \tag{3.31}$$

$$= (p(p-1))^{-1}\left\{x^p - (1-p)y^p - pxy^{p-1}\right\} \tag{3.32}$$

where here $p = 2, 1, 0$ is for EU, KL and IS cost functions. We note that this parametrization is more common in the literature [19, 32, 49, 63].

3.3.3. Alpha Divergence

Alpha divergence is a special case of the f-divergence where f function is chosen to be the dual cumulant function ϕ obtained above. The connection between beta and alpha divergence has already been pointed out by [63]. By definition, for any convex function f providing that $f(1) = 0$, the f-divergence of y from x is given by [66]

$$d_f(x,y) = yf(\frac{x}{y}) \tag{3.33}$$

noting that we intentionally drop the grand sum needed for the matrix variables for the notational simplicity. Then taking ϕ in Equation 3.12 as $\phi() = f()$ we come up with

$$d_\alpha(x,y) = y\left\{((1-p)(2-p))^{-1}(x/y)^{2-p} + m(x/y) + d\right\} \tag{3.34}$$

Using the values for the constants m and d

$$m = (2-p) \qquad d = (1-p) \tag{3.35}$$

we obtain the alpha divergence as

$$d_\alpha(x,y) = ((1-p)(2-p))^{-1}\left\{x^{2-p}y^{1-p} + (2-p)x + (1-p)y\right\} \tag{3.36}$$

or equivalently

$$d_\alpha(x,y) = (p(p-1))^{-1}\left\{x^p y^{p-1} + px + (p-1)y\right\} \tag{3.37}$$

3.4. Likelihood, Deviance and β-divergence

3.4.1. Derivation of Cumulant Function

In this section we compute cumulant function $\psi()$ for EDMs. It will be used for the computation of the deviance for EDMs. Recall that we have the identities

$$\int \partial\theta = \int \mu^{-p}\partial\mu \tag{3.38}$$

$$\theta = \theta(\mu) = (1-p)^{-1}\mu^{1-p} + m \tag{3.39}$$

then inverting Equation 3.39, i.e. solving for the expectation parameter μ as

$$\mu = \mu(\theta) = [(1-p)(\theta - m)]^{1/(1-p)} \tag{3.40}$$

Then we obtain the cumulant function as follows

$$\frac{\partial\psi(\theta)}{\partial\theta} = \mu = \mu(\theta) \qquad \Rightarrow \qquad \int \partial\psi(\theta) = \int \mu(\theta)\partial\theta \tag{3.41}$$

$$\int \partial\psi(\theta) = \int [(1-p)(\theta-m)]^{1/(1-p)}\,\partial\theta \tag{3.42}$$

$$\psi(\theta) = (1-p)^{-1}\frac{1-p}{2-p}\left[(1-p)(\theta-m)\right]^{(2-p)/(1-p)} + d \tag{3.43}$$

$$= (2-p)^{-1}\left[(1-p)(\theta-m)\right]^{(2-p)/(1-p)} + d \tag{3.44}$$

where m and d are usual constants. Then, by following [25] we may get rid of m and d by re-parametrizing the natural (canonical) parameter from θ to $\theta_1 = \theta - m$ that changes the cumulant function from $\psi(\theta)$ to $\psi_1(\theta_1) = \psi(\theta_1 + m)$ and we use $\psi(\theta)$ and

θ rather than $\psi_1(\theta_1)$ and θ_1. As a summary, starting from the following

$$\frac{\partial\theta}{\partial\mu} = \mu^{-p} \qquad \frac{\partial\psi(\theta)}{\partial\theta} = \mu = \mu(\theta) \tag{3.45}$$

we end up with

$$\theta = \theta(\mu) = (1-p)^{-1}\mu^{1-p} \tag{3.46}$$

$$\mu = \mu(\theta) = [(1-p)\theta]^{1/(1-p)} \tag{3.47}$$

$$\psi(\theta) = (2-p)^{-1}\left[(1-p)\theta\right]^{(2-p)/(1-p)} \tag{3.48}$$

For the next section where we analyze beta divergence and deviance relationship we will need the cumulant function in terms of μ as $\psi(\mu)$. This is indeed given in Equation 3.48 and we simply use θ given in Equation 3.46 in Equation 3.48

$$\psi(\theta) \equiv \psi(\theta(\mu)) = (2-p)^{-1}\mu^{2-p} \tag{3.49}$$

Then we find $-x\theta$ where we simply use Equation 3.46

$$-x\theta = -x(1-p)^{-1}\mu^{1-p} \tag{3.50}$$

Remark 3.4. Recall that an alternative expression for Bregman divergence is given as Equation 2.85

$$d(x,\mu) = \phi(x) + \psi(\theta(\mu)) - x\theta(\mu) \tag{3.51}$$

Then, we obtain each term in Equation 3.51 in terms of μ and add them up to find the divergence function. The cumulant function is already computed in Equation 3.49 and the dual of cumulant function $\phi(x)$ is already computed in Equation 3.12 while the

product $-x\theta$ is $-x\theta = -x(1-p)^{-1}\mu^{1-p}$. Then we add them up to get the divergence

$$d(x,\mu) = \left\{((1-p)(2-p))^{-1}\, x^{2-p}\right\} + \left\{(2-p)^{-1}\mu^{2-p}\right\} + \left\{-x(1-p)^{-1}\mu^{1-p}\right\} \quad (3.52)$$

$$= ((1-p)(2-p))^{-1}\left\{(1-p)\mu^{2-p} - x(2-p)\mu^{1-p} + x^{2-p}\right\} \quad (3.53)$$

3.4.2. Log Likelihood of Exponential Dispersion Models

In this section we compute the log-likelihood of the EDMs which we will use during the computation of both the entropy and the deviance. Recall the definition of the EDM

$$p(x|\theta,\varphi) = h(x,\varphi)exp\left\{\varphi\left(\theta x - \psi(\theta)\right)\right\} \quad (3.54)$$

Then the log-likelihood in terms of θ and μ is

$$\mathcal{L}(\theta) = \mathcal{L}(x|\theta) = \varphi(\theta x - \psi(\theta)) + \log h(x,\varphi) \quad (3.55)$$

$$\mathcal{L}(\theta(\mu)) = \mathcal{L}(x|\theta(\mu)) = \varphi(\theta(\mu)x - \psi(\theta(\mu))) + \log h(x,\varphi) \quad (3.56)$$

For the last equation, i.e. for $\mathcal{L}(\theta(\mu))$, substituting $\theta(\mu)$ and $\psi(\theta(\mu))$ from Equation 3.46 and Equation 3.48 we compute the log-likelihood in terms of $\mathcal{L}(\theta(\mu))$

$$\mathcal{L}(\theta(\mu)) = \varphi((1-p)^{-1}\mu^{1-p}x - (2-p)^{-1}\left[(1-p)\theta\right]^{(2-p)/(1-p)}) + \log h(x,\varphi) \quad (3.57)$$

$$\mathcal{L}(\mu) = \mathcal{L}(x|\mu) = \varphi((1-p)^{-1}\mu^{1-p}x - (2-p)^{-1}\mu^{2-p}) + \log h(x,\varphi) \quad (3.58)$$

One very important quantity is the derivative of the log-likelihood $\mathcal{L}$ w.r.t. the expectation parameter μ. The useful point in this derivation is that the the base measure $h()$ is independent of the expectation (and also the canonical) parameter and base measure disappears smoothly

$$\frac{\partial\mathcal{L}(\mu)}{\partial\mu} = \varphi\mu^{-p}(x-\mu) \quad (3.59)$$

3.4.3. Entropy of Exponential Dispersion Models

In this section we compute the entropy of EDMs. This analysis is similar to Menendez's proof where he computes the entropy for exponential families [52]. By definition the Shannon's entropy is

$$H[\theta] = -\int p(x|\theta) \log p(x|\theta)\ d\mu(x) \tag{3.60}$$

We plug in $\log p(x|\theta)$ that has already been computed in Equation 3.55

$$H[\theta] = -\int p(x|\theta)\big(\varphi(x\theta - \psi(\theta) + \log h(x,\varphi))\big)\ d\mu(x) \tag{3.61}$$

Then taking the constant terms outside of the integral we compute the entropy for EDM

$$H[\theta] = -\varphi\Big(\theta \int p(x|\theta)x\ d\mu(x) - \psi(\theta)\int p(x|\theta) + \int p(x|\theta)\log h(x,\varphi)\ d\mu(x)\Big) \tag{3.62}$$

$$= -\varphi\Big(\theta\mu - \psi(\theta) + \langle \log h(x,\varphi)\rangle\Big) \tag{3.63}$$

and in terms of expectation parameter

$$H[\mu] = -\varphi\Big(\phi(\mu) + \langle \log h(x,\varphi)\rangle\Big) \tag{3.64}$$

Example 3.1. *To check the entropy for EDMs given in Equation 3.63 consider unknown mean and known variance of the Gaussian case $\mathcal{N}(x;a,b)$ with the following equalities*

$$\varphi = \frac{1}{b} \quad \theta(\mu) = a \quad \psi(\theta) = \frac{1}{2}\theta^2 = \frac{a^2}{2} \quad \log h(x,\varphi) = -\frac{x^2}{2} - \frac{b}{2}\log 2\pi - \frac{b}{2}\log b \tag{3.65}$$

where we obtain them from Example 2.1 for unknown mean known variance Gaussian of exponential family by dividing by the factor $\varphi = \frac{1}{b}$. Then we apply Equation 3.63 to

find the entropy as

$$H[\theta] = -\varphi\Big(a^2 - \frac{a^2}{2} + \Big\langle -\frac{x^2}{2} - \frac{b}{2}\log 2\pi - \frac{b}{2}\log b \Big\rangle\Big) \tag{3.66}$$

$$= -\varphi\Big(a^2 - \frac{a^2}{2} - \frac{\langle x^2\rangle}{2} - \frac{b}{2}\log 2\pi - \frac{b}{2}\log b\Big) \tag{3.67}$$

$$= -\varphi\Big(a^2 - \frac{a^2}{2} - \frac{a^2}{2} - \frac{b}{2} - \frac{b}{2}\log 2\pi - \frac{b}{2}\log b\Big) \tag{3.68}$$

$$= -\varphi\Big(-\frac{b}{2} - \frac{b}{2}\log 2\pi - \frac{b}{2}\log b\Big) \tag{3.69}$$

$$= \frac{1}{2} + \frac{1}{2}\log 2\pi + \frac{1}{2}\log b \tag{3.70}$$

where the expectation of the sufficient statistics is $\langle x^2\rangle = a^2 + b$.

3.4.4. The Deviance and the β-divergence

Deviance is a statistical term to qualify the fit of the statistics to the model [50] and defined as [23] (adapted notation)

$$D(x,\mu) = 2\left[\sup\{x\theta - \psi(\theta)\} - \{x\theta(\mu) - \psi(\theta(\mu))\}\right] \tag{3.71}$$

The deviance is equal to 2 times of the *log-likelihood ratio* of the full model compared to the reduced (observation) model. A full model is the model where each sample in the observation is represented by a parameter. In GLM [67], the deviance is used to compare two models

$$D(\mu_f,\mu)\varphi = 2\big(\mathcal{L}(\mu_f) - \mathcal{L}(\mu)\big) = 2(\log p(x|\mu_f) - \log p(x|\mu)) \tag{3.72}$$

where μ_f is for the parameters of the full model which is as large as the samples. On the other hand, the likelihood of the full model with parameter μ_f can be taken as substituting x for μ in the likelihood formula for EDM (3.58) as

$$\mathcal{L}(\mu_f) \equiv \mathcal{L}(x) = \varphi((1-p)^{-1}x^{1-p}x - (2-p)^{-1}x^{2-p}) + \log h(x,\varphi) \tag{3.73}$$

Then the log likelihood ratio $\mathcal{L}(x) - \mathcal{L}(\mu)$ becomes

$$= \varphi\Big((1-p)^{-1}x^{1-p}x - (2-p)^{-1}x^{2-p}\Big) - \varphi\Big((1-p)^{-1}\mu^{1-p}x - (2-p)^{-1}\mu^{2-p}\Big) \quad (3.74)$$

$$= \varphi\left((1-p)(2-p)\right)^{-1}\Big(x^{2-p} - ((2-p)^{-1}\mu^{1-p}x + (1-p)^{-1}\mu^{2-p})\Big) \quad (3.75)$$

Interestingly, this result is exactly equivalent to the beta divergence that we derive in Equation 3.27 except the inverse of the dispersion parameter φ. Hence the beta divergence is equal to the half of *scaled deviance* [67] (dividing the deviance by the inverse of the dispersion parameter)

$$D(x,\mu) = 2\varphi d_\beta(x,\mu) = 2\big(\mathcal{L}(\mu_f) - \mathcal{L}(\mu)\big) \quad (3.76)$$

where $D()$ is for the deviance and $d_\beta()$ is for the beta divergence.

One significance of this analysis is that beta divergence can be used in a model selection framework and its measure of discrepancy of goodness of fit can be qualified via $\mathcal{X}^2$. This is due to the fact that the deviance itself is $\mathcal{X}^2$ distributed random variable for Gaussian models and it can be approximated to $\mathcal{X}^2$ for non-Gaussian models as number of observations gets larger, i.e. for large n. Even better approximation is for the difference of two deviance for comparing two models, that is

$$\begin{aligned} D(x,\mu_1) - D(x,\mu_2) &= 2\big(\mathcal{L}(\mu_f) - \mathcal{L}(\mu_1)\big) - 2\big(\mathcal{L}(\mu_f) - \mathcal{L}(\mu_2)\big) \\ &= 2\big(\mathcal{L}(\mu_2) - \mathcal{L}(\mu_1)\big) \end{aligned} \quad (3.77)$$

is $\mathcal{X}^2$ distributed [67]. Hence, when beta divergence is used in model selection framework we have a measure of goodness of fit as

$$2\varphi d_\beta(x,\mu) \sim \mathcal{X}^2_{n-m} \quad (3.78)$$

where n is the size of observations $x_1, \ldots, x_n$ and m is the number of parameters in μ, i.e. the number of components to compute the mean.

One other interesting quantity in this analysis is the log-likelihood of the full model. It is indeed closely related to a well-known quantity; the entropy. The likelihood of the full model is the maximum achievable likelihood and it is a quantity of data given the data and hence independent of the parameters. Note that, on the other hand, the entropy is independent of the location parameter. Also check that in Equation 3.73

$$\mathcal{L}(\mu_f) \equiv \mathcal{L}(x) = \varphi((1-p)^{-1}x^{1-p}x - (2-p)^{-1}x^{2-p}) + \log h(x,\varphi) \tag{3.79}$$

$$= \varphi\Big(\big((1-p)(2-p)\big)^{-1}x^{2-p}\Big) + \log h(x,\varphi) \tag{3.80}$$

$$= \varphi\phi(x) + \log h(x,\varphi) \tag{3.81}$$

and recalling the likelihood of the estimated model

$$\mathcal{L}(\theta(\mu)) = \mathcal{L}(x|\theta(\mu)) = \varphi(\theta(\mu)x - \psi(\theta(\mu))) + \log h(x,\varphi) \tag{3.82}$$

where their difference recovers the divergence equation

$$\mathcal{L}(\mu_f) - \mathcal{L}(\mu) = \varphi(\phi(x) - \theta(\mu)x + \psi(\theta(\mu))) \tag{3.83}$$

3.5. ML and MAP Optimization for EDMs

For the *inference* or the learning of the parameters we want to optimize μ w.r.t. the log likelihood of an EDM. For this we take the maximum likelihood approach and take the derivative of log likelihood w.r.t. μ

$$\log p(x|\theta,\varphi) = \log h(x,\varphi) + \{\varphi\left(\theta x - \psi(\theta)\right)\} \tag{3.84}$$

Then taking the derivative w.r.t. θ as

$$\frac{\partial \log p(x|\theta,\varphi)}{\partial\theta} = \varphi\left(x - \frac{\partial\psi(\theta)}{\partial\theta}\right) = \varphi(x-\mu) \tag{3.85}$$

since $\mu = \frac{\partial \psi(\theta)}{\partial \theta}$. For the derivative w.r.t. the expectation parameter μ, we use the chain rule as

$$\frac{\partial \log p(x|\theta,\varphi)}{\partial \theta}\frac{\partial \theta}{\partial \mu} = \varphi(x-\mu)\frac{\partial \theta}{\partial \mu} = \varphi(x-\mu)\mu^{-p} \tag{3.86}$$

since $\frac{\partial \theta}{\partial \mu} = \mu^{-p}$. Note that it is

$$\frac{\partial \log p(x|\theta,\varphi)}{\partial \mu} = \varphi(x-\mu)\mu^{-p} = \frac{(x-\mu)}{Var(x)} \tag{3.87}$$

since $\varphi\mu^{-p} = \frac{1}{Var(x)}$ where this result matches to the result in Equation 3.59.

Remark 3.5. The second way to get the same result is to use derivative of the Bregman divergence. Note that we can re-write the log likelihood of EDMs in terms of Bregman divergence as follows

$$\log p(x|\theta,\varphi) = \log h(x,\varphi) + \varphi\phi(x) - \{\varphi\left(\phi(x) - \theta(\mu)x + \psi(\theta(\mu))\right)\} \tag{3.88}$$

$$= b(x,\varphi) - \varphi d_\phi(x,\mu) \tag{3.89}$$

where $b(x,\varphi) = \log h(x,\varphi) + \varphi\phi(x)$ is independent of the canonical (θ) and expectation (μ) parameters. Then taking the derivative of Equation 3.89 w.r.t. μ

$$\frac{\partial \log p(x|\theta,\varphi)}{\partial \mu} = \frac{\partial - \varphi d_\phi(x,\mu)}{\partial \mu} = -\varphi\frac{\partial d_\phi(x,\mu)}{\partial \mu} \tag{3.90}$$

which is the derivative of Bregman divergence which respect to the second parameter

$$\frac{\partial \log p(x|\theta,\varphi)}{\partial \mu} = -\varphi\Big((x-\mu)\frac{\partial^2\phi(\mu)}{\partial \mu^2}\Big) \tag{3.91}$$

Note that $\phi(\mu) = ((1-p)(2-p))^{-1}\mu^{2-p}$ and $\partial^2\phi(\mu)/\partial\mu^2 = \mu^{-p}$ where we obtain Equation 3.86.

For the MAP optimization the conjugate priors for the EDM family are given as

$$p(\theta|n_0, x_0) = h(n_0, x_0) \exp\left\{n_0 \left(x_0\theta - \psi(\theta)\right)\right\} \tag{3.92}$$

with n_0, x_0 being the hyperparameters corresponding to belief about a prior sample size and prior expectation of the mean parameter [68]. Here the function h is as usual for the normalization and disappears during optimization. For the M-step we take the logarithm and derivative w.r.t. to μ

$$\frac{\partial \log p(\theta|n_0, x_0)}{\partial \mu} = \frac{\partial \log p(\theta|n_0, x_0)}{\partial \theta}\frac{\partial \theta}{\partial \mu} = n_0(x_0 - \mu)\frac{\partial \theta}{\partial \mu} = (x_0 - \mu)n_0\mu^{-p} \tag{3.93}$$

It is interesting to compare this MAP equation with that of ML optimization where we quote as follows

$$\frac{\partial \log p(\theta|n_0, x_0)}{\partial \mu} = n_0(x_0 - \mu)\mu^{-p} \tag{3.94}$$

$$\frac{\partial \log p(x|\theta, \varphi)}{\partial \mu} = \varphi(x - \mu)\mu^{-p} \tag{3.95}$$

3.6. ML and MAP Optimization of the Data Augmentation Model

The main motivation of this section is that we want to optimize the data augmentation model $x = \sum_r s_r$ where x is observed array while s is the latent. The result of this section will be used in Chapter 4 for the generalization of the tensor factorization to the beta divergence.

3.6.1. EM Algorithm for the Exponential Dispersion Models

Now, consider the data augmentation model $x = \sum_r s_r$ where s is a random vector with the *exponential dispersion models* (EDM) [23]

$$p(s_r|\theta, \varphi_s) = h(s_r, \varphi_s) \exp\left(\varphi_s \left(\theta s_r - \psi(\theta)\right)\right) \tag{3.96}$$

Here, as usual θ is the canonical parameter, φ_s^{-1} is the *dispersion parameter* which is common for all s_r by definition. Also, ψ is the cumulant function. The mean parameter, now is denoted by λ_r rather than μ and is tied to θ as $\lambda_r = \partial\psi(\theta)/\partial\theta$. Note that the distribution can also be characterized by the mean and the dispersion as $p(s_r|\lambda_r, \varphi_s^{-1})$. The *power variance function* [23–25] is given as $v(\lambda_r) = \lambda_r^p$ which is defined as $\partial\theta/\partial\lambda_r = v(\lambda_r)^{-1}$ [23]. Finally the variance is equal to $Var(s_r) = \varphi_s^{-1} v(\lambda_r) = \varphi_s^{-1}\lambda^p$. Recall that the variable x is augmented as $x = \sum_r s_r$ with the mean parameter $\hat{x} = \sum_r \lambda_r$. The objective is that x is observed and we optimize the model for specific λ_r. We formulate the EM for ML estimation as

$$\text{(E step)} \qquad {s_r}^{(n+1)} = \left\langle s_r | x, \lambda^{(n)} \right\rangle \tag{3.97}$$

$$\text{(M step)} \qquad {\lambda_r}^{(n+1)} = \arg\max_{\lambda_r} \log p({s_r}^{(n+1)}|\lambda, \varphi_s) \tag{3.98}$$

The M-step requires the derivative of the log likelihood $\mathcal{L}(\lambda) = \log p(s|\lambda, \varphi_s)$

$$\frac{\partial \log p(s|\lambda, \varphi_s)}{\partial\lambda} = \frac{\partial \log p(s|\theta, \varphi_s)}{\partial\theta}\frac{\partial\theta}{\partial\lambda} \tag{3.99}$$

$$= \varphi_s(s-\lambda)\frac{\partial\theta}{\partial\lambda} = \frac{(s-\lambda)}{Var(s)} = (s-\lambda)\varphi_s\lambda^{-p} \tag{3.100}$$

The E-step, on the other hand, introduces a difficult problem as it requires identification of the posterior expectation $\left\langle s_r|x, \lambda^{(n)}\right\rangle$ in the general form. We derive it by exploiting the tightness of the EM lower bound and outline it in the following theorem.

Theorem 3.1. *Let s be a random vector with the mean λ from the EDM family with the power variance functions and let x be a variable with the mean $\hat{x}$ given as $x = \sum_r s_r$. Then the posterior expectation for the variable s_r is given as*

$$\langle s_r|x, \lambda\rangle = \lambda_r + \frac{{\lambda_r}^p}{\sum_r {\lambda_r}^p}(x - \hat{x}) \tag{3.101}$$

Proof. We start with the expression for the EM lower bound for the likelihood as

$$\log p(x|\hat{x}) \geq \langle \log p(x|s_r)\rangle_{q(s_r)} + \langle \log p(s_r|\lambda_r)\rangle_{q(s_r)} + H[q(s_r)] \tag{3.102}$$

with $q(s_r)$ being any arbitrary distribution and $H[q(s_r)]$ being the entropy such that whenever the distribution $q(s_r)$ becomes equal to be true posterior $q(s_r) = p(s_r|x, \lambda)$, the bound becomes tight and the inequality turns to an equality [46]. If we replace $q(s_r)$ with the true posterior and take the derivative w.r.t. λ_r by using the chain rule and noting that $\partial\hat{x}/\partial\lambda_r = 1$

$$\frac{\partial \log p(x|\hat{x})}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial \lambda_r} = \left\langle \frac{\partial \log p(x|s_r)}{\partial \lambda_r} \right\rangle_{p(s_r|x,\lambda)} + \left\langle \frac{\partial \log p(s_r|\lambda_r)}{\partial \lambda_r} \right\rangle_{p(s_r|x,\lambda)} + \frac{\partial H[p(s_r|x,\lambda)]}{\partial \lambda_r} \tag{3.103}$$

Then regarding the differentiation w.r.t. λ_r in an M-step of EM setting where λ_r is independent of the approximating distribution $q(s_r)$ we end up with

$$\frac{\partial \log p(x|\hat{x})}{\partial \hat{x}} = \left\langle \frac{\partial \log p(s_r|\lambda_r)}{\partial \lambda_r} \right\rangle_{p(s_r|x,\lambda)} \tag{3.104}$$

$$\frac{x - \hat{x}}{Var(x)} = \frac{\langle s_r|x, \lambda\rangle - \lambda_r}{Var(s_r)} \tag{3.105}$$

Finally we solve for $\langle s_r|x, \lambda\rangle$ as

$$\langle s_r|x, \lambda\rangle = \lambda_r + \frac{Var(s_r)}{Var(x)}(x - \hat{x}) \tag{3.106}$$

$$\tag{3.107}$$

and using the fact that x is aggregation of multiple s_r and using power variance functions, i.e. pluging $v(\lambda_r) = \lambda_r^p$ and relating $Var(x)$ to $Var(s)$ as

$$Var(s_r) = {\varphi_s}^{-1}{\lambda_r}^p \qquad \Rightarrow \qquad Var(x) = \sum_r Var(s_r) = \varphi_s^{-1} \sum_r {\lambda_r}^p \tag{3.108}$$

we end up with

$$\langle s_r|x, \lambda\rangle = \lambda_r + \frac{{\lambda_r}^p}{\sum_r {\lambda_r}^p}(x - \hat{x}) \tag{3.109}$$

□

Theorem 3.1 gives the general form of the posterior expectations for the posterior distribution $p(s_r|x,\lambda)$ from EDM family, without explicitly constructing the posterior density.

Lemma 3.1. *Let φ_s^{-1} and φ_x^{-1} be the dispersion parameters of the EDM variables s and x and ρ be their ratio as $\rho = \varphi_s^{-1}/\varphi_x^{-1}$. Then the posterior expectation for EDM family with $p = \{0,1,2\}$ is given as*

$$\langle s_r|x,\lambda\rangle = \lambda_r + \rho\frac{{\lambda_r}^p}{\hat{x}^p}(x-\hat{x}) \tag{3.110}$$

The convolution property implies that $x|\hat{x}$ and $s|\lambda$ have the same class [25] of distributions implying the power parameter p values are the same. Hence $Var(x) = \varphi_x^{-1}\hat{x}^p$. Note that Gaussian and Poisson distributions are closed under arbitrary convolution properties [23], while for the gamma common scale parameter is required, i.e. let $s_r \sim \mathcal{G}(s_r;\lambda_r,\gamma)$ and if $x = \sum_r s_r$ then $x \sim \mathcal{G}(x;\hat{x},\gamma)$ with $\hat{x} = \sum_r \lambda_r$. Thus the ratio of dispersions is $\rho = \{1/|r|, 1, \hat{x}/\lambda_r\}$ for $p = \{0,1,2\}$.

3.6.2. Updates via EM, MUR and Alternating Projections for ML

In this section we consider x to be a random vector given as $x_i = \sum_r s_{ir}$ with the expectation $\hat{x}_i = \sum_r \lambda_{ir}$ where $\lambda_{ir} = u_i z_r$. The dispersion φ_s^{-1} is common for all s_r. We optimize the model for z_r noting that optimization for u_i is identical.

Theorem 3.2. *The optimization for z_r via EM_{ML} for EDM family with $p = \{0,1,2\}$ is given as*

$$z_r \leftarrow z_r + \frac{z_r \sum_i \rho\,(x_i - \hat{x}_i)\,\hat{x}_i^{-p} u_i}{\sum_i \lambda_{ir}^{1-p} u_i} \qquad \textit{where} \;\; u_i = \frac{\partial \lambda_{ir}}{\partial z_r} \tag{3.111}$$

where for implementation we use the following form to eliminate dependency on λ_{ir}

$$z_r \leftarrow z_r + \frac{z_r^p \sum_i \rho\,(x_i - \hat{x}_i)\,\hat{x}_i^{-p} u_i}{z_r^{1-p} \sum_i u_i^{2-p}} \tag{3.112}$$

Proof. For optimizing z_r the M-step in Equation 3.98 requires multiplication by the derivative of $\partial\lambda_{ir}/\partial z_r$ due to the chain rule. Then by substituting $\langle s_{ir}|x_i,\lambda_i\rangle$ we obtain EM_{ML} update given in Equation 3.111

$$\frac{\partial\mathcal{L}}{\partial z_r}=\sum_i(\langle s_{ir}|x_i,\lambda_i\rangle-\lambda_{ir})\varphi_s\lambda_{ir}^{-p}u_i=0 \qquad \Rightarrow \qquad z_r=\frac{\sum_i\langle s_{ir}|x_i,\lambda_i\rangle\ u_i^{1-p}}{\sum_i\ u_i^{2-p}} \tag{3.113}$$

□

Theorem 3.3. *The optimization for z_r via MUR_{ML} for EDM family with $p=\{0,1,2\}$ is given as*

$$z_r=z_r\frac{\sum_i x_i\hat{x}_i^{-p}u_i}{\sum_i\hat{x}_i^{1-p}u_i} \tag{3.114}$$

Proof. We set up the gradient ascent iterative optimization schema with the adaptive learning rate η_r similar to [5, 32] to cancel out the negative term and ρ as

$$\frac{\partial\mathcal{L}}{\partial z_r}=\sum_i\rho\left(x_i-\hat{x}_i\right)\hat{x}_i^{-p}u_i \qquad \eta_r=\frac{z_r}{\sum_i\rho\hat{x}_i^{1-p}u_i} \qquad z_r\leftarrow z_r+\eta_r\frac{\partial\mathcal{L}}{\partial z_r} \tag{3.115}$$

□

The convergence proof of the update Equation 3.114 for the matrices is given by [49] while pointing out that plugging $\lambda_{ir}^{1-p}=\rho\hat{x}_i^{1-p}$ in the denominator of the EM Equation 3.111, i.e. replacing the denominator with $\sum_i\rho\hat{x}_i^{1-p}u_i$ turns the EM update into the MUR update Equation 3.114. Interestingly for $p=0$, the relationship turns to $\lambda_{ir}=\hat{x}_i/|r|$ recalling $\hat{x}_i=\sum_r\lambda_{ir}$. A direct solution by setting the gradient to zero and solving directly in an *alternating projection* manner is also available. This method is equivalent to *alternating least square* (ALS) for EU cost ($p=0$)

$$\frac{\partial\mathcal{L}}{\partial z_r}=\sum_i\rho\left(x_i-\hat{x}_i\right)\hat{x}_i^{-p}u_i=0 \qquad \Rightarrow \qquad \sum_i x_i\hat{x}_i^{-p}u_i=\sum_i\hat{x}_i^{1-p}u_i \tag{3.116}$$

3.6.3. Conjugate priors and MAP Estimation

The conjugate priors for the EDM family are given as

$$p(\theta|n_0, s_0) = h(n_0, s_0) \exp\left\{n_0 \left(s_0\theta - \psi(\theta)\right)\right\} \tag{3.117}$$

with n_0, s_0 being the hyperparameters corresponding to belief about a prior sample size and prior expectation of the mean parameter [68]. Here the function h is for the normalization and disappears during optimization. For the M-step we take the logarithm and derivative w.r.t. to λ

$$\frac{\partial \log p(\theta|n_0, s_0)}{\partial \lambda} = \frac{\partial \log p(\theta|n_0, s_0)}{\partial \theta}\frac{\partial \theta}{\partial \lambda} = n_0(s_0 - \lambda)\frac{\partial \theta}{\partial \lambda} = (s_0 - \lambda)n_0\lambda^{-p} \tag{3.118}$$

With this, the M-step turns to be MAP optimization, i.e. optimizing w.r.t. the log posterior. Note that the prior distribution is factorized as $p(\lambda_{ir}) = p(z_r)p(u_i)$ and after taking the derivative w.r.t. z_r M-step becomes

$$\text{(M step)} \qquad \lambda_{ir}^{(n+1)} = \arg\max_{\lambda_{ir}} \log p(s_{ir}^{(n+1)}|\lambda, \varphi_s) + \log p(z_r|n_0, z_0) \tag{3.119}$$

3.6.4. Updates via EM and MUR for MAP

Theorem 3.4. *The MAP optimization via the EM with the conjugate prior* $p(z_r|n_0, z_0)$ *is*

$$z_r = \frac{n_0 z_0 + \sum_i \langle s_{ir}|x_i, \lambda_i\rangle \varphi_s u_i^{1-p}}{n_0 + \sum_i \varphi_s u_i^{2-p}} \tag{3.120}$$

Proof. We optimize z_r in the M-step by maximizing the log posterior

$$\frac{\partial \mathcal{L}_{MAP}}{\partial z_r} = \left\{ \sum_i (\langle s_{ir}|x_i, \lambda_i\rangle - \lambda_{ir})\varphi_s \lambda_{ir}^{-p} u_i \right\} + (z_0 - z_r) n_0 z_r^{-p} = 0 \tag{3.121}$$

$$= \left\{ \sum_i (\langle s_{ir}|x_i, \lambda_i\rangle - \lambda_{ir})\varphi_s u_i^{1-p} \right\} + (z_0 - z_r) n_0 \tag{3.122}$$

$$= \sum_i \langle s_{ir}|x_i, \lambda_i\rangle \varphi_s u_i^{1-p} - z_r \left\{ n_0 + \sum_i \varphi_s u_i^{-p} \right\} + z_0 n_0 \tag{3.123}$$

and we solve for z_r. □

Just after substituting $\langle s_{ir}|x_i, \lambda_i\rangle$ for the EDM family with $p = \{0, 1, 2\}$ into Equation 3.120 the EM_{MAP} and the MUR_{MAP} updates turn out

$$EM_{MAP}: \qquad z_r \leftarrow \frac{n_0 z_0 + z_r \sum_i \left\{ \varphi_s u_i^{1-p} + z_r^{p-1} \varphi_x (x_i - \hat{x}_i) \hat{x}_i^{-p} \right\} u_i}{n_0 + \sum_i \varphi_s u_i^{2-p}} \tag{3.124}$$

$$MUR_{MAP}: \qquad z_r \leftarrow \frac{n_0 z_0 + z_r^p \sum_i \varphi_x u_i x_i \hat{x}_i^{-p}}{n_0 + z_r^{p-1} \sum_i \varphi_x u_i \hat{x}_i^{1-p}} \tag{3.125}$$

where MUR_{MAP} is obtained by heuristic plugging $\lambda_{ir}^{1-p} = \rho \hat{x}_i^{1-p}$ in EM_{MAP} Equation 3.124 just after multiplying both numerator and denominator by z_r^{1-p} and identifying $(z_r u_i)^{1-p}$ as λ_{ir}^{1-p}. This effectively puts back the z_r^{-p} cancelled in Equation 3.121.

3.7. Solution for Non-linearity

In the previous section we optimize the data augmentation model $x = \sum_r s_r$ w.r.t. the factor z where $p(s_r|\lambda_r)$ is an EDM type distribution and the parameter λ_r is further factorized as $\lambda_r = zu$ (ignoring the other subscripts). This is simply a linear model and we make use the linearity when we need the derivative

$$\frac{\partial \hat{x}}{\partial \lambda_r} = 1 \tag{3.126}$$

Here in this section we do not assume a linear model and extract update equations for the factor z under certain assumptions. While we neither extend the derivation here to the tensor case nor derive equations for MAP case, it can easily be extended to the tensors similar to Chapter 4.

We continue from Equation 3.103 while we leave the derivation term as is

$$\frac{\partial \log p(x|\hat{x})}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial \lambda_r} = \left\langle \frac{\partial \log p(s_r|\lambda_r)}{\partial \lambda_r} \right\rangle_{p(s_r|x,\lambda)} \tag{3.127}$$

$$\frac{x - \hat{x}}{Var(x)} \frac{\partial \hat{x}}{\partial \lambda_r} = \frac{\langle s_r|x,\lambda\rangle - \lambda_r}{Var(s_r)} \tag{3.128}$$

and we solve for the posterior expectation $\langle s_r|x,\lambda\rangle$ as

$$\langle s_r|x,\lambda\rangle = \lambda_r + \frac{Var(s_r)}{Var(x)}(x - \hat{x})\frac{\partial \hat{x}}{\partial \lambda_r} \tag{3.129}$$

where $Var(x)$ and $Var(s)$ are as

$$Var(s_r) = {\varphi_s}^{-1} v(s_r) \tag{3.130}$$

$$Var(x) = \varphi_x^{-1} v(x) \tag{3.131}$$

Here $v(s_r)$ and $v(x)$ are the variance functions. Noting that in this section for the generalization purpose we do not use power variance functions and use variance functions without and further assumptions.

3.7.1. Matching Link and Loss Functions

In Equation 3.129 the term that needs to be computed is the derivative $\partial\hat{x}/\partial\lambda_r$. In this section we use one assumption that this derivative can be further simplified. The assumption is called as matching link and loss function [39,67,69] and is explained in this section. In addition the table 3.2 lists canonical link, variance function and other related quantities for the major class of the EDM family distributions.

Table 3.2. Certain parameters and functions for major class of probability distributions. Adapted from [67]

.

	Gaussian	**Poisson**	**Gamma**	**Inv. Gaussian**
Canonical link name	identity	log	inverse	inverse squared
Canonical link: $\theta(\mu)$	μ	$\log\mu$	μ^{-1}	μ^{-2}
Derivative: $\partial\theta(\mu)/\partial\mu$	1	μ^{-1}	μ^{-2}	μ^{-3}
Variance func. $v(\mu)$	1	μ	μ^2	μ^3
p for $v(\mu)=\mu^p$	0	1	2	3
$v(\mu)\times\partial\theta(\mu)/\partial\mu$	1	1	1	1

Here we treat non-linearity very similar to that of *generalized linear models* [50,67] (GLM). We will be using GLM in Chapter 6 extensively, but here we use a small part of its terminology to construct the non-linear model. In GLM terminology for the equality

$$g(\hat{x}) = \kappa \tag{3.132}$$

$\hat{x}$ is a random component as the expectation of the probabilistic model (ignoring the subscript i for now), κ is a systematic component and $g(.)$ is the *link function* that links both. In our case the systematic component is to be

$$\kappa = \sum_r \lambda_r \tag{3.133}$$

We write the link between random and systematic component in terms of inverse link function

$$g(\hat{x}) = \kappa \qquad \Rightarrow \qquad \hat{x} = g^{-1}(\kappa) \tag{3.134}$$

Then taking the derivative

$$\frac{\partial \hat{x}}{\partial \lambda_r} = \frac{\partial g^{-1}(\kappa)}{\partial \lambda_r} = \frac{\partial g^{-1}(\kappa)}{\partial \kappa}\frac{\partial \kappa}{\partial \lambda_r} \tag{3.135}$$

$$= \left(\frac{\partial g(\hat{x})}{\partial \hat{x}}\right)^{-1}\frac{\partial \kappa}{\partial \lambda_r} \tag{3.136}$$

where here we make use of simple mathematical equality for $f(x) = y$ and $f^{-1}(y) = x$

$$\frac{df^{-1}(y)}{dy} = \frac{dx}{df(x)} \tag{3.137}$$

Then the derivatives $\frac{\partial g(\hat{x})}{\partial \hat{x}}$ and $\frac{\partial \kappa}{\partial \lambda_r}$ are left to be identified. The second derivative is quite easy to compute since

$$\frac{\partial \kappa}{\partial \lambda_r} = \frac{\partial \sum_r \lambda_r}{\partial \lambda_r} = 1 \tag{3.138}$$

For the first derivative, we do not compute it directly and instead it will disappear from Equation 3.129 after substitution when we assume a matching link and loss function. In EDMs the loss is represented by the relation between the canonical parameter θ and the expectation parameter μ and is expressed by the function $\theta(\mu)$ (here in this section μ is $\hat{x}$, and hence $\theta(\hat{x})$). Then the matching assumption states that the canonical parameter function $\theta(\hat{x})$ is to be equal to the canonical link function

$$g(\mu) \equiv g(\hat{x}) = \theta(\hat{x}) \tag{3.139}$$

and then taking derivative w.r.t. the expectation parameter $\hat{x}$

$$\frac{\partial g(\hat{x})}{\partial \hat{x}} = \frac{\partial \theta(\hat{x})}{\partial \hat{x}} = v(\hat{x})^{-1} \tag{3.140}$$

which is by definition inverse of the variance function. Now we may return to Equation 3.129 and make use of the matching assumption to simplify it but first we substitute

the definition of variances $var(x), var(s_r)$ and the derivative $\frac{\partial \hat{x}}{\partial \lambda_r}$

$$\langle s_r|x,\lambda\rangle = \lambda_r + \frac{{\varphi_s}^{-1} v(s_r)}{\varphi_x^{-1} v(x)}(x-\hat{x})\frac{1}{\frac{\partial g(\hat{x})}{\partial \hat{x}}}\frac{\partial \kappa}{\partial \lambda_r} \tag{3.141}$$

and after noting that as we just derive

$$v(x)\frac{\partial g(\hat{x})}{\partial \hat{x}} = 1 \qquad \text{and} \qquad \frac{\partial \kappa}{\partial \lambda_r} = 1 \tag{3.142}$$

the expectation of posterior under matching condition turns to be

$$\langle s_r|x,\lambda\rangle = \lambda_r + \frac{{\varphi_s}^{-1} v(s_r)}{\varphi_x^{-1}}(x-\hat{x}) \tag{3.143}$$

noting that depending on the context we may use two more alternative forms as

$$\langle s_r|x,\lambda\rangle = \lambda_r + \frac{var(s_r)}{\varphi_x^{-1}}(x-\hat{x}) \tag{3.144}$$

$$= \lambda_r + \rho v(s_r)(x-\hat{x}) \tag{3.145}$$

where in the last form the ratio of inverse dispersion parameters is already defined as $\rho = \varphi_s^{-1}/\varphi_x^{-1}$.

3.7.2. Optimizing for the Factors

Now, we may optimize for the individual factors. Recall the M-step optimization yields the equation (without the sum)

$$\frac{s_r - \lambda_r}{Var(s_r)} = 0 \tag{3.146}$$

and recall that we consider x to be a random vector given as $x_i = \sum_r s_{ir}$ with the expectation $\hat{x}_i = \sum_r \lambda_{ir}$ where $\lambda_{ir} = u_i z_r$. The dispersion φ_s^{-1} is common for all s_r. We optimize the model for z_r whereas optimization for u_i is identical. Hence the M-step

optimization turns to the equation

$$\sum_i \frac{\langle s_r|x,\lambda\rangle - \lambda_r}{Var(s_r)} u_i = 0 \tag{3.147}$$

where u_i is the derivative $\partial\lambda_{ir}/\partial z_r$. We explicitly mark λ_r as $\lambda_r^{(m)}$ where superscript index m indicates that it is an M-step term. This is needed since an E-Step $\lambda_r^{(e)}$ will appear just after substituting expectation of posterior $\langle s_r|x,\lambda\rangle$ as

$$\sum_i \frac{\left\{\lambda_{ir}^{(e)} + \frac{Var(s_r)}{\varphi_x^{-1}}(x-\hat{x})\right\} - \lambda_{ir}^{(m)}}{Var(s_r)} u_i = 0 \tag{3.148}$$

Then the solution

$$\sum_i \frac{\lambda_{ir}^{(e)}}{Var(s_r)} u_i + \sum_i \varphi_x(x-\hat{x})u_i = \sum_i \frac{\lambda_{ir}^{(m)}}{Var(s_r)} u_i \tag{3.149}$$

Note that $\sum_i \lambda_{ir}^{(m)}$ can always be written as $z_r^{(m)} \sum_i u_i$ as

$$\sum_i \lambda_{ir}^{(e)} = z_r^{(e)} \sum_i u_i \tag{3.150}$$

then

$$z_r^{(e)} \sum_i \frac{u_i}{Var(s_r)} u_i + \sum_i \varphi_x(x-\hat{x})u_i = z_r^{(m)} \sum_i \frac{u_i}{Var(s_r)} u_i \tag{3.151}$$

which is

$$z_r^{(m)} = z_r^{(e)} + \frac{\sum_i \varphi_x(x-\hat{x})u_i}{\sum_i \frac{u_i}{Var(s_r)} u_i} \tag{3.152}$$

As a special case of power variance functions, i.e. $Var(s_r) = \varphi_s^{-1}\lambda_{ir}^p$

$$z_r \leftarrow z_r + \frac{\sum_i \rho(x-\hat{x})u_i}{\sum_i \lambda_{ir}^{-p} u_i^2} \tag{3.153}$$

For a comparison with the identity cost function for $p = \{0, 1, 2\}$ recall that the update equation is given as

$$z_r \leftarrow z_r + \frac{z_r \sum_i \rho\,(x_i - \hat{x}_i)\,\hat{x}_i^{-p} u_i}{\sum_i \lambda_{ir}^{1-p} u_i} \qquad \text{where} \;\; u_i = \frac{\partial \lambda_{ir}}{\partial z_r} \tag{3.154}$$

that can be turn to

$$z_r \leftarrow z_r + \frac{\sum_i \rho\,(x_i - \hat{x}_i)\,\hat{x}_i^{-p} u_i}{\sum_i \lambda_{ir}^{-p} u_i^2} \tag{3.155}$$

Hence the main difference is the inverse variance function of x, i.e. $v(x)^{-1} = x^{-p}$ in the numerator.

3.8. Summary

In this chapter, we presented the theory of exponential dispersion models in various respects, that has been used as the main machinery behind our cost function generalization perspective. The main points of this chapter are as follows.

We link the alpha and the beta divergences from power variance functions of EDMs after deriving dual cumulant function for the EDMs and applying Bregman divergence and f-divergence. The constants m and d in dual cumulant function disappear in beta divergence while we specifically set for certain values for the alpha divergence.

$$\phi(x) = \left((1-p)(2-p)\right)^{-1} x^{2-p} + mx + d \tag{3.156}$$

$$d_\beta(x, y) = \phi(x) - \phi(y) - (x - y)\frac{\partial \phi(y)}{\partial y} \qquad \text{for any } m, d \tag{3.157}$$

$$d_\alpha(x, y) = y\phi(\frac{x}{y}) \qquad \text{for } m = 2 - p \quad d = 1 - p \tag{3.158}$$

We also showed that the beta divergence is equal to the half of *scaled deviance*

$$D(x, \mu) = 2\varphi d_\beta(x, \mu) = 2\big(\mathcal{L}(\mu_f) - \mathcal{L}(\mu)\big) \tag{3.159}$$

where $D()$ is for the deviance and $d_\beta()$ is for the beta divergence. Then we obtain generic equation for entropy of EDMs

$$H[\mu] = -\varphi\Big(\phi(\mu) + \langle \log h(x,\varphi)\rangle\Big) \tag{3.160}$$

In the last part of this chapter we obtain equalities that we use for the tensor factorization for the optimization of the factors via EM. First is the derivative of the log likelihood

$$\frac{\partial \log p(x|\theta,\varphi)}{\partial \theta} = \varphi(x-\mu) \qquad \frac{\partial \log p(x|\theta,\varphi)}{\partial \mu} = \frac{(x-\mu)}{Var(x)} \tag{3.161}$$

while the last derivative with conjugate priors turns to be

$$\frac{\partial \log p(\theta|n_0,x_0)}{\partial \mu} = n_0(x_0-\mu)\frac{\partial \theta}{\partial \mu} = (x_0-\mu)n_0\mu^{-p} \tag{3.162}$$

Perhaps one of the most interesting equation in this chapter is the general equation for the expectation of the posterior distribution, which is required for the E-step of the EM

$$\langle s_r|x,\lambda\rangle = \lambda_{ir} + \frac{\lambda_{ir}^p}{\sum_r \lambda_{ir}^p}(x-\hat{x}) \tag{3.163}$$

$$= \lambda_{ir} + \rho\frac{\lambda_{ir}^p}{\hat{x}^p}(x-\hat{x}) \qquad \text{with } \rho = \varphi_s^{-1}/\varphi_x^{-1} \text{ for } p=0,1,2 \tag{3.164}$$

where the last equation is a special case for the Gaussian, Poisson and gamma(under certain condition), i.e. for the distributions that exhibit arbitrary convolution property

The rest of the chapter derives the fixed point update equations via EM for ML and MAP. For the notational simplicity derivations assume two factors z and u matrices, while in Chapter 4 we generalize results to arbitrary number of latent tensors.

4. PROBABILISTIC LATENT TENSOR FACTORIZATION

4.1. Introduction

In this chapter we first propose and outline a general probabilistic framework for tensor factorization with suitable notation. Then we give maximum likelihood (ML) and maximum a posterior (MAP) estimations based on expectation maximization (EM) solution, i.e. fixed point update equations for latent factors for various cost function such as Kullback-Leibler (KL), Euclidean (EU) or Itakura-Saito (IS). Besides EM, we derive alternative algorithms with multiplicative update rules (MUR) and alternating projections. Our approach inspires a novel link between probabilistic graphical models and tensor factorization models. We formulate the fixed point updates as a message passing algorithm on a graph where vertices correspond to indices and cliques represent factors of the tensor factorization.

This chapter is organized as follows: Section 4.2 introduces the notation and the probability model for PLTF and derives the update equation for the KL divergence as the special case. Here we also show the link between tensor factorization and graphical models and represent TF models as the notion of the graphs. In addition prior knowledge in the form of conjugate priors is incorporated into the update equations. Section 4.3 generalizes the TF problem for the beta divergence as the cost function by the optimization of the likelihood of the exponential dispersion models including the use of conjugate priors. In this section besides EM based ML and MAP update rules we also derive updates for multiplicative update rules as alternating least square (for Euclidean cost) methods. Section 4.4 introduces a method for matricizing and tensorizing the element-wise update equations developed in earlier sections along with the examples.

4.2. Latent Tensor Factorization Model

We define a *tensor* Λ as a multiway array with an index set $\mathcal{V} = \{i_1, i_2, \ldots, i_N\}$ where each index $i_n = 1 \ldots |i_n|$ for $n = 1 \ldots N$. Here, $|i_n|$ denotes the cardinality of the index i_n. An *element of the tensor* Λ is a scalar that we denote by $\Lambda(i_1, i_2, \ldots, i_N)$ or $\Lambda^{i_1, i_2, \ldots, i_N}$ or as a shorthand notation by $\Lambda(v)$. Here, v will be a particular configuration from the product space of all indices in $\mathcal{V}$. For our purposes, it will be convenient to define *a collection of tensors*, $Z_{1:N} = \{Z_\alpha\}$ for $\alpha = 1 \ldots N$, sharing a set of indices $\mathcal{V}$. Here, each tensor Z_α has a corresponding index set $\mathcal{V}_\alpha$ such that $\cup_{\alpha=1}^{N} \mathcal{V}_\alpha = \mathcal{V}$. Then, v_α denotes a particular configuration of the indices for Z_α while $\bar{v}_\alpha$ denotes a configuration of the complement $\bar{\mathcal{V}}_\alpha = \mathcal{V}/\mathcal{V}_\alpha$.

A *tensor contraction* or *marginalization* is simply adding the elements of a tensor over a given index set, i.e., for two tensors Λ and $\hat{X}$ with index sets $\mathcal{V}$ and $\mathcal{V}_0$ we write $\hat{X}(v_0) = \sum_{\bar{v}_0} \Lambda(v)$ or $\hat{X}(v_0) = \sum_{\bar{v}_0} \Lambda(v_0, \bar{v}_0)$. To clarify our notation, we present the following example

Example 4.1. *Consider the ordinary matrix multiplication*

$$\hat{X}(i, j) = \sum_k Z_1(i, k) Z_2(k, j) \tag{4.1}$$

which is a tensor contraction operation. Although never done in practical computation, formally we can define $\Lambda(i, j, k) = Z_1(i, k) Z_2(k, j)$ *and sum over the index* k *to find the result. In our formalism, we define* $\mathcal{V} = \{i, j, k\}$*, where* $\mathcal{V}_0 = \{i, j\}$*,* $\mathcal{V}_1 = \{i, k\}$ *and* $\mathcal{V}_2 = \{k, j\}$*. Hence,* $\bar{\mathcal{V}}_0 = \{k\}$ *and we write* $\hat{X}(v_0) = \sum_{\bar{v}_0} Z_1(v_1) Z_2(v_2)$*.*

A *tensor factorization* (TF) model is the product of a collection of tensors $Z_{1:N} = \{Z_\alpha\}$ for $\alpha = 1 \ldots N$, each defined on the corresponding index set $\mathcal{V}_\alpha$, collapsed over a set of indices $\mathcal{V}_0$. Given a particular TF model, the latent TF problem is to estimate

a set of latent tensors $Z_{1:N}$

$$\text{minimize } D(X||\hat{X}) \text{ s.t. } \hat{X}(v_0) = \sum_{\bar{v}_0} \prod_{\alpha} Z_\alpha(v_\alpha) \tag{4.2}$$

where X is an observed tensor and $\hat{X}$ is the prediction. Here, both objects are defined over the same index set $\mathcal{V}_0$ and are compared elementwise. The function $D(\cdot||\cdot) \geq 0$ is a cost function. In this chapter, we use first the Kullback-Leibler (KL) divergence as our cost. Later in the generalization section we introduce a form of the β-divergence that unifies Euclidean (EU), KL and Itakura-Saito (IS) cost functions.

Example 4.2. *The TUCKER3 factorization [12, 13] aims to find Z_α for $\alpha = 1 \dots 4$ that solves the following optimization problem*

$$\textit{minimize } D(X||\hat{X}) \textit{ s.t. } \hat{X}^{i,j,k} = \sum_{p,q,r} Z_1^{i,p} Z_2^{j,q} Z_3^{k,r} Z_4^{p,q,r} \qquad \forall i,j,k \tag{4.3}$$

Both for visualization and for efficient computation, it is useful to introduce a graphical notation to represent the factorization implied by a particular TF model. We define an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$ and associate each index with a vertex of $\mathcal{G}$. For each pairs of indices appearing in a factor index set $\mathcal{V}_\alpha$ for $\alpha = 1 \dots N$, we add an edge to the edge set of the graph. Consequently, each clique (fully connected subgraph) of $\mathcal{G}$ corresponds to the factor index set $\mathcal{V}_\alpha$.

Table 4.1 illustrates graphical representation of popular TF models such as Matrix Factorization(MF), CP, and TUCKER3. Any TF model can be visualized using the semantics of undirected graphical models where cliques (fully connected subgraphs) correspond to individual factors and vertices correspond to indices. Whilst algebraically identical to a probabilistic undirected graphical model, our representation merely captures the factorization of the TF model and does not have a probabilistic semantics (we do not represent a probability measure and the indices are not random variables).

As an alternative graphical model representation TF models can be visualized

and processed as *factors graphs* rather than directed acyclic graphs. [70] illustrates NMF, NMFD, NMF2D and SF-SSNTF as factor graphs.

In the following section, we introduce a probabilistic model where we cast the minimization problem into an equivalent maximum likelihood estimation problem, i.e., solving the TF problem given in Equation 4.2 will be equivalent to maximization of $\log p(X|Z_{1:N})$ with respect to Z_α [31, 39].

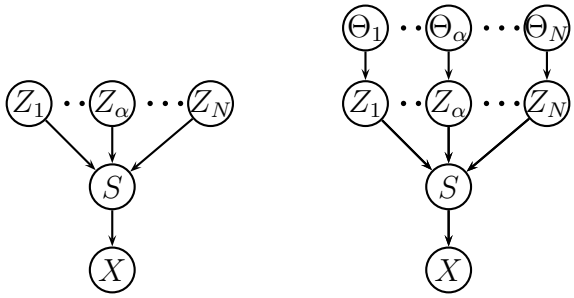
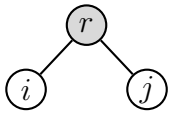
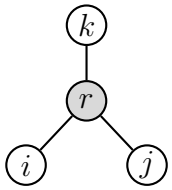
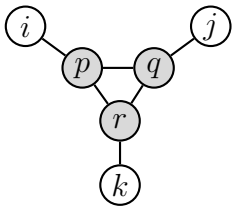
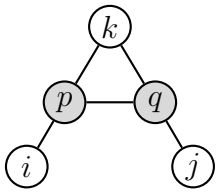


Figure 4.1. The underlying probability model of PLTF in the conventional sense is given by the DAG with and without prior knowledge (as a Bayesian network model). Here X is the observed multiway data and Z_α's are the model parameters. The latent tensor S allows us to treat the problem in a data augmentation setting and apply the EM algorithm.

Table 4.1. Graphical representation of popular TF models such as Matrix Factorization(MF), CP, and TUCKER3. The shaded vertices are latent, i.e., correspond to indices that are not elements of $\mathcal{V}_0$, the index set of X.

Model	**Graph**	**Indices**	
MF		$\mathcal{V} = \{i, j, r\}$ $\mathcal{V}_0 = \{i, j\}$ $\bar{\mathcal{V}}_0 = \{r\}$	$\mathcal{V}_1 = \{i, r\}$ $\mathcal{V}_2 = \{j, r\}$
		$\hat{X}^{i,j} = \sum_r Z_1^{i,r} Z_2^{j,r}$	
CP		$\mathcal{V} = \{i, j, k, r\}$ $\mathcal{V}_0 = \{i, j, k\}$ $\bar{\mathcal{V}}_0 = \{r\}$	$\mathcal{V}_1 = \{i, r\}$ $\mathcal{V}_2 = \{j, r\}$ $\mathcal{V}_3 = \{k, r\}$
		$\hat{X}^{i,j,k} = \sum_r Z_1^{i,r} Z_2^{j,r} Z_3^{k,r}$	
TUCKER3		$\mathcal{V} = \{i, j, k, p, q, r\}$ $\mathcal{V}_0 = \{i, j, k\}$ $\bar{\mathcal{V}}_0 = \{p, q, r\}$	$\mathcal{V}_1 = \{i, p\}$ $\mathcal{V}_2 = \{j, q\}$ $\mathcal{V}_3 = \{k, r\}$ $\mathcal{V}_4 = \{p, q, r\}$
		$\hat{X}^{i,j,k} = \sum_{p,q,r} Z_1^{i,p} Z_2^{j,q} Z_3^{k,r} Z_4^{p,q,r}$	
PARATUCK2		$\mathcal{V} = \{i, j, k, p, q\}$ $\mathcal{V}_0 = \{i, j, k\}$ $\bar{\mathcal{V}}_0 = \{p, q\}$ $\mathcal{V}_1 = \{i, p\}$	$\mathcal{V}_2 = \{j, q\}$ $\mathcal{V}_3 = \{k, p\}$ $\mathcal{V}_4 = \{k, q\}$ $\mathcal{V}_5 = \{p, q\}$
		$\hat{X}^{i,j,k} = \sum_{p,q} Z_1^{i,p} Z_2^{j,q} Z_3^{k,p} Z_4^{k,q} Z_5^{p,q}$	

4.2.1. Probability Model

The PLTF generative model represented as the directed acyclic graph (DAG) in Figure 4.1 is defined as below

$$\Lambda(v) = \prod_{\alpha}^{N} Z_{\alpha}(v_{\alpha}) \qquad \text{intensity} \tag{4.4}$$

$$S(v) \sim \mathcal{PO}(S; \Lambda(v)) \qquad \text{KL cost} \tag{4.5}$$

$$X(v_0) = \sum_{\bar{v}_0} S(v) \qquad \text{observation} \tag{4.6}$$

$$\hat{X}(v_0) = \sum_{\bar{v}_0} \Lambda(v) \qquad \text{parameter} \tag{4.7}$$

$$M(v_0) = \begin{cases} 0 & X(v_0) \text{ is missing} \\ 1 & \text{otherwise} \end{cases} \qquad \text{mask array} \tag{4.8}$$

Here, $\Lambda(v)$ the product of the factors is *intensity* or *latent intensity field*, $S(v)$ is latent source, and $X(v_0)$ is augmented data. There is a probability distribution associated with $S(v)$. $\mathcal{PO}(s; \lambda)$ denotes the Poisson density. Also note that the Poisson distribution implies non negativity. Due to the reproductivity property [71] of the Poisson density, the observation $X(v_0)$ has the same type of distribution as $S(v)$. Moreover, missing data is handled smoothly as in the likelihood [29, 30]

$$p(X, S|Z) = \prod_{v} \Big(p(X(v_0)|S(v))\; p(S(v)|\Lambda(v)) \Big)^{M(v_0)} \tag{4.9}$$

Note that maximizing Poisson likelihood is equivalent to minimizing KL cost function as

$$D(X||\hat{X}) = \sum_{v} X(v_0) \log \frac{X(v_0)}{\hat{X}(v_0)} - X(v_0) + \hat{X}(v_0) \tag{4.10}$$

4.2.2. Fixed Point Update Equation for KL Cost

The log likelihood $\mathcal{L}_{KL}$ is given as

$$\mathcal{L}_{KL} = \sum_{v} M(v_0) \left(S(v) \log \Lambda(v) - \Lambda(v) - \log(S(v)!) \right) \tag{4.11}$$

subject to the constraint $X(v_0) = \sum_{\bar{v}_0} S(v)$ whenever $M(v_0) = 1$. We can easily optimise $\mathcal{L}_{KL}$ for Z_α by an EM algorithm. In the E-step we calculate the posterior expectation $\langle S(v)|X(v_0)\rangle$ by identifying the posterior $p(S|X,Z)$ as a multinomial distribution [30,71]. In the M-step we solve the optimization problem $\partial\mathcal{L}_{KL}/\partial Z_\alpha(v_\alpha) = 0$ to get the fixed point update

$$\text{(E step)} \qquad \langle S(v)|X(v_0)\rangle = \frac{X(v_0)}{\hat{X}(v_0)}\Lambda(v) \tag{4.12}$$

$$\text{(M step)} \qquad Z_\alpha(v_\alpha) = \frac{\sum_{\bar{v}_\alpha} M(v_0)\langle S(v)|X(v_0)\rangle}{\sum_{\bar{v}_\alpha} M(v_0)\frac{\partial\Lambda(v)}{\partial Z_\alpha(v_\alpha)}} \tag{4.13}$$

with the following equalities

$$\frac{\partial\Lambda(v)}{\partial Z_\alpha(v_\alpha)} = \partial_\alpha\Lambda(v) = \prod_{\alpha'\neq\alpha} Z_{\alpha'}(v_{\alpha'}) \qquad \Lambda(v) = Z_\alpha(v_\alpha)\partial_\alpha\Lambda(v) \tag{4.14}$$

After substituting Equation 4.12 in Equation 4.13 and noting that $Z_\alpha(v_\alpha)$ being independent of the sum $\sum_{\bar{v}_\alpha}$ we obtain the following multiplicative fixed point iteration for Z_α

$$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha)\frac{\sum_{\bar{v}_\alpha} M(v_0)\frac{X(v_0)}{\hat{X}(v_0)}\partial_\alpha\Lambda(v)}{\sum_{\bar{v}_\alpha} M(v_0)\partial_\alpha\Lambda(v)} \tag{4.15}$$

Example 4.3. *This algorithm specialises the well-known KL-NMF multiplicative update rules [5] when* $\mathcal{V} = \{i,j,k\}$, $\mathcal{V}_0 = \{i,j\}$, $\mathcal{V}_1 = \{i,k\}$, $\mathcal{V}_2 = \{j,k\}$ *and the observa-*

tion is given as the matrix $X^{i,j}$

$$\hat{X}^{i,j} = \sum_k Z_1^{i,k} Z_2^{k,j} \tag{4.16}$$

$$Z_1^{i,r} \leftarrow Z_1^{i,r} \frac{\sum_j (M^{i,j} X^{i,j} / \hat{X}^{i,j}) Z_2^{r,j}}{\sum_j M^{i,j} Z_2^{r,j}} \tag{4.17}$$

Example 4.4. *The multiplicative update rule for CP is generated by* $PLTF_{KL}$ *with the setting* $N = 3$, $\mathcal{V} = \{i, j, k, r\}$, $\mathcal{V}_0 = \{i, j, k\}$, $\mathcal{V}_1 = \{i, r\}$, $\mathcal{V}_2 = \{j, r\}$ *and* $\mathcal{V}_3 = \{k, r\}$. *The model estimate and the fixed point equation for* Z_1 *are as*

$$\hat{X}^{i,j,k} = \sum_r Z_1^{i,r} Z_2^{j,r} Z_3^{k,r} \tag{4.18}$$

$$Z_1^{i,r} \leftarrow Z_1^{i,r} \frac{\sum_{j,k} (M^{i,j,k} X^{i,j,k} / \hat{X}^{i,j,k}) Z_2^{j,r} Z_3^{k,r}}{\sum_{j,k} M^{i,j,k} Z_2^{j,r} Z_3^{k,r}} \tag{4.19}$$

Example 4.5. *The multiplicative update rule for TUCKER3 is generated by* $PLTF_{KL}$ *with the setting* $N = 4$, $\mathcal{V} = \{i, j, k, p, q, r\}$, $\mathcal{V}_0 = \{i, j, k\}$, $\mathcal{V}_1 = \{i, p\}$, $\mathcal{V}_2 = \{j, q\}$ $\mathcal{V}_3 = \{k, r\}$ *and for core tensor* $\mathcal{V}_4 = \{p, q, r\}$. *The model estimate, and the fixed point update equations for* Z_1 *and for the core tensor* Z_4 *are as*

$$\hat{X}^{i,j,k} = \sum_{p,q,r} Z_1^{i,p} Z_2^{j,q} Z_3^{k,r} Z_4^{p,q,r} \tag{4.20}$$

$$Z_1^{i,p} \leftarrow Z_1^{i,p} \frac{\sum_{j,k,q,r} (M^{i,j,k} X^{i,j,k} / \hat{X}^{i,j,k}) Z_2^{j,q} Z_3^{k,r} Z_4^{p,q,r}}{\sum_{j,k,q,r} M^{i,j,k} Z_2^{j,q} Z_3^{k,r} Z_4^{p,q,r}} \tag{4.21}$$

$$Z_4^{p,q,r} \leftarrow Z_4^{p,q,r} \frac{\sum_{i,j,k} (M^{i,j,k} X^{i,j,k} / \hat{X}^{i,j,k}) Z_1^{i,p} Z_2^{j,q} Z_3^{k,r}}{\sum_{i,j,k} M^{i,j,k} Z_1^{i,p} Z_2^{j,q} Z_3^{k,r}} \tag{4.22}$$

4.2.3. Priors and Constraints

In this section we incorporate the prior information into the $PLTF$. Recall that the conjugate prior for the Poisson observation model is the Gamma density

$$Z_\alpha(v_\alpha) \sim \mathcal{G}\Big(Z_\alpha(v_\alpha); A_\alpha(v_\alpha), B_\alpha(v_\alpha)\Big) \tag{4.23}$$

where hyper-parameter arrays A_α and B_α match the size of Z_α. Now complete data likelihood becomes as given [29, 30]

$$p(X|S)p(S|Z_{1:N})p(Z_{1:N}|\Theta_{1:N}) = \tag{4.24}$$

$$\prod_v \Big(p(X(v_0)|S(v))\; p(S(v)|Z_{1:N}(v_\alpha))\Big)^{M(v_0)} \prod_\alpha^N p(Z_\alpha(v_\alpha)|A_\alpha(v_\alpha), B_\alpha(v_\alpha)) \tag{4.25}$$

where N is the number of factors. The log-likelihood $\mathcal{L}_{KL}$ is

$$\mathcal{L}_{KL} = \sum_v M(v_0)\left(S(v)\log\Lambda(v) - \Lambda(v) - \log(S(v)!)\right) \tag{4.26}$$

$$+ \sum_\alpha \sum_{v_\alpha} (A_\alpha(v_\alpha) - 1)\log Z_\alpha(v_\alpha) \tag{4.27}$$

$$- B_\alpha(v_\alpha)Z_\alpha(v_\alpha) - \log\Gamma(A_\alpha(v_\alpha)) + A_\alpha(v_\alpha)\log B_\alpha(v_\alpha) \tag{4.28}$$

We optimize the log posterior for $Z_\alpha(v_\alpha)$ as below noting that the first part is due to the log likelihood of the Poisson while the second part is due to the conjugate gamma prior

$$\frac{\partial\mathcal{L}}{\partial Z_\alpha(v_\alpha)} = \left(\sum_{\bar{v}_\alpha} M(v_0)\frac{S(v)-\Lambda(v)}{\Lambda(v)}\partial_\alpha\Lambda(v)\right) + \left(\frac{A_\alpha(v_\alpha)-1}{Z_\alpha(v_\alpha)} - B_\alpha(v_\alpha)\right) \tag{4.29}$$

$$Z_\alpha(v_\alpha) \leftarrow \frac{(A_\alpha(v_\alpha)-1) + Z_\alpha(v_\alpha)\sum_{\bar{v}_\alpha} M(v_0)\frac{X(v_0)}{\hat{X}(v_0)}\partial_\alpha\Lambda(v)}{B_\alpha(v_\alpha) + \sum_{\bar{v}_\alpha} M(v_0)\partial_\alpha\Lambda(v)} \tag{4.30}$$

This update converges to the mode of the posterior distribution $p(Z_{1:N}|X)$. This simply computes the mode of the full conditional $p(Z_\alpha|X, Z_{-\alpha})$, that is a gamma distribution

for each element $Z_\alpha(v_\alpha)$. Here, $Z_{-\alpha}$ denotes all other factors $Z_{\alpha'}$ for $\alpha' = 1 \dots N$, such that $\alpha \neq \alpha'$.

This prior can be used to impose sparsity: we can take the prior $\mathcal{G}(x; a, a/m)$ with mean m, and variance m^2/a. For small α, most of the elements of Z are expected to be around zero, with only a few large ones [30].

4.2.4. Relation to Graphical Models

An important observation that leads to computational savings and compact representation of the element-wise update equation is that the update in Equation 4.15 consists of structurally similar terms in both the denominator and the numerator. For large TF models, this structure needs to be exploited for computational efficiency. Therefore, we define the following tensor valued function, where $|Q|$ denote the dimensionality of the object Q

Definition 4.1. *We define a tensor valued function* $\Delta_\alpha(Q) : \mathbb{R}^{|Q|} \to \mathbb{R}^{|Z_\alpha|}$

$$\Delta_\alpha^p(Q) = \left[\sum_{\bar{v}_\alpha} Q(v_0) \ (\partial_\alpha \Lambda(v))^p \right] \tag{4.31}$$

where $\Delta_\alpha(Q)$ and the variable Q are tensors with the same size as Z_α and of the observation X respectively. The nonnegative integer p on the right side denotes the element-wise power while on the left, it should be interpreted as a parameter of the function. When $p = 1$ we will omit it and use as Δ_α instead of Δ_α^1. Using this notation, we rewrite the update equation for KL cost in Equation 4.15 compactly as

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\Delta_\alpha(M \circ X/\hat{X})}{\Delta_\alpha(M)} \tag{4.32}$$

where $\circ$ and / stand for element-wise multiplication and division respectively. We note that the computation of $\Delta_\alpha(Q)$ is model specific and needs to be computed for MF, CP and TUCKER3 separately. Later we develop the explicit matrix forms of these

updates. Again, using this new definition, we rewrite the fixed point update equation with conjugate priors for KL cost in Equation 4.30 compactly as

$$Z_\alpha \leftarrow \frac{(A_\alpha - 1) + Z_\alpha \circ \Delta_\alpha(M \circ X/\hat{X})}{B_\alpha + \Delta_\alpha(M)} \tag{4.33}$$

Interestingly, computing $\Delta_\alpha(Q)$ is equivalent to computing a tensor contraction, marginal potential over vertices $\mathcal{V}_\alpha$. This operation can be viewed as the computation of certain marginal densities given an undirected graphical model [8,46]. More precisely, we construct a joint tensor $P = X(v_0)\prod_\alpha Z_\alpha(v_\alpha)$. For each α, we remove Z_α from P, which we denote as P/Z_α, and compute the *marginal tensor* by summing over $\bar{\mathcal{V}}_\alpha$. An example for the TUCKER model is shown in Figure 4.2. In the figure, (a) Joint tensor P defined as $P = X\prod_\alpha Z_\alpha$, (b) Graph for P/Z_1 used for computation of Δ_1 where $\mathcal{V}_1 = \{i, p\}$, (c) Graph for P/Z_4 used for Δ_4 corresponding to the core tensor where $\mathcal{V}_4 = \{p, q, r\}$. . Rather than using a naive direct approach, the contraction can be computed efficiently by distributing the summation over the product, a procedure that is algebraically identical to variable elimination for inference in probabilistic graphical models. (d) This graph is model for P/X where it is interesting to compare it with that of core tensor P/G in (c). Notice the symmetry between two components X and G such that the nodes i, j, k are replaced by p, q, r. The symmetry is also observable from the matricized equations $X_{(1)} = AG_{(1)}(C \otimes B)^T$ and $G_{(1)} = A^T X_{(1)}(C \otimes B)$ where matrix transpose operations can be related to the replacement of the nodes i, j, k by p, q, r in the graph.

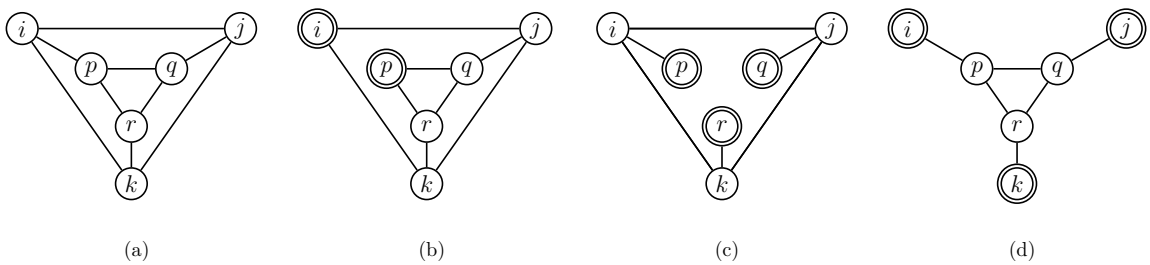


Figure 4.2. Undirected graphical models for the computation of the Δ_α of TUCKER3 decomposition. Double circled indices are not summed over.

An interesting observation is that with the semantics of $\Delta()$ function $\hat{X}$ can just be regarded as another factor just like Z_α. Note the similarity of the two equations

$$Z_\alpha \equiv \Delta_\alpha(\hat{X}) = \left[\sum_{\bar{v}_\alpha}\left(\hat{X}(v_0)\prod_{\alpha'\neq\alpha} Z_{\alpha'}(v_{\alpha'})\right)\right] \tag{4.34}$$

$$\hat{X} \equiv \Delta_{\hat{X}}(\mathbf{1}) = \left[\sum_{\bar{v}_0}\left(\prod_{\alpha} Z_\alpha(v_\alpha)\right)\right] \tag{4.35}$$

with $\mathbf{1}$ is the tensor of all ones. Here Δ function for the component Z_α is indeed marginalization of the intensity potentials over the rest of the variables $\bar{v}_\alpha$. Similarly $\Delta_{\hat{X}}$ for $\hat{X}$ is marginalization of the intensity potentials over the configuration variables $\bar{v}_0$. Also note that for the examples when we replace the notation Z_α as A, B, C, G as in the case of TUCKER3 we simply write the function Δ for A as $\Delta_A()$. Table 4.2 lists Δ functions for popular TF models.

We will never compute the Δ functions as element-wise but instead we will build matrix primitive and make the computation as product of matrix objects. However, if element-wise computation is considered we can use the graphical models methodology; the *variable elimination* where we push the sums as far as possible [46]. In the following example for the TUCKER3, note that the inner-most sum $\sum_q B^{j,q}G^{p,q,r}$ can be saved as $T^{j,p,r}$ temporarily and is used in the left most sums

$$A^{i,p} \equiv \sum_{jkqr} \hat{X}^{i,j,k} B^{j,q} C^{k,r} G^{p,q,r} \tag{4.36}$$

$$= \sum_{jk} \hat{X}^{i,j,k} \sum_{r} C^{k,r} \sum_{q} B^{j,q} G^{p,q,r} \tag{4.37}$$

$$= \sum_{jk} \hat{X}^{i,j,k} \sum_{r} C^{k,r} T^{j,p,r} \tag{4.38}$$

4.3. Generalization of TF for β-Divergence

In the previous section we developed update equations for PLTF for KL cost by modeling the latent tensor S as $S(v) \sim \mathcal{PO}(S; \Lambda(v))$. In this section we continue to

Table 4.2. Model output and Δ_α functions for the factors.

MODEL	$\hat{X}_{(1)}$	$\Delta_\alpha^Z(Q)$
NMF	$\hat{X}_{(1)} = AB$	$\Delta_A(Q) = XB^T$
CP	$\hat{X}_{(1)} = A(C \odot B)^T$	$\Delta_A(Q) = X_{(1)}(C \odot B)$
TUCKER3	$\hat{X}_{(1)} = AG_{(1)}(C \otimes B)^T$	$\Delta_A(Q) = X_{(1)}(C \otimes B)G_{(1)}^T$
TUCKER3 (CORE)	$\hat{X}_{(1)} = AG_{(1)}(C \otimes B)^T$	$\Delta_G(Q) = A^T X_{(1)}(C \otimes B)$

obtain PLTF update equations for KL, EU, and IS costs in single expressions as the special case of the β-divergence [32, 49]

$$d_\beta(x, y) = \frac{x^{2-p} + (1-p)y^{2-p} - (2-p)xy^{1-p}}{(2-p)(1-p)} \tag{4.39}$$

where for $p = 0$ and as the limiting cases for $p = 1, 2$ it reduces to

$$d_\beta(x, y) = \begin{cases} \frac{1}{2}(x-y)^2 & p = 0 \ \ \text{(EU)} \\ x \log \frac{x}{y} + y - x & p = 1 \ \ \text{(KL)} \\ \frac{x}{y} - \log \frac{x}{y} - 1 & p = 2 \ \ \text{(IS)} \end{cases} \tag{4.40}$$

Recall that in Chapter 3 we derive the β-divergence from the power variance functions of the exponential dispersion models [23] which relate the variance of the distributions to some power p of their means $\Lambda(v)$ given as $Var(S(v)) = \varphi_s^{-1} v(\Lambda(v))$ with φ_s^{-1} being the dispersion parameter and $v(\Lambda(v)) = \Lambda(v)^p$ being the *variance function* [23].

This section follows the similar outline as the previous section where we obtain various update rules for the TF models with β-divergence via the EM. Here we present the main results and leave technical details to Chapter 3. In addition, to keep the notation simpler, we will drop iteration indices from the update equations and ignore the missing mask array M, assuming all the elements of X are observed.

4.3.1. EM Updates for ML Estimate

Adapting the development in Chapter 3 on the EM for EDMs, we start with the E-step where the posterior expectation $\langle S(v)|X(v_0)\rangle$ for the EDMs with $p = 0, 1, 2$ is identified as

$$\text{(E step)} \qquad \langle S(v)|X(v_0)\rangle = \Lambda(v) + \rho\frac{\Lambda(v)^p}{\hat{X}(v_0)^p}\Big(X(v_0) - \hat{X}(v_0)\Big) \tag{4.41}$$

Here ρ is the ratio of the dispersion parameters identified as

$$\rho = \frac{\varphi_s^{-1}}{\varphi_x^{-1}} = \begin{cases} \frac{1}{K} & p = 0 \;\; \text{(Gaussian)} \\ 1 & p = 1 \;\; \text{(Poisson)} \\ \frac{\hat{X}(v_0)}{\Lambda(v)} & p = 2 \;\; \text{(Gamma)} \end{cases} \tag{4.42}$$

where for the Gaussian case K is the cardinality of the invisible vertex set $\bar{\mathcal{V}}_0$, i.e. $K = |\bar{\mathcal{V}}_0|$ recalling $\hat{X}(v_0) = \sum_{\bar{v}_0}\Lambda(v)$. Substituting the expectation in Equation 4.41 into the M-step gives the ML estimate for the factors

$$\text{(M step)} \qquad Z_\alpha(v_\alpha) = \frac{\sum_{\bar{v}_\alpha}\langle S(v)|X(v_0)\rangle\,\partial_\alpha\Lambda(v)^{1-p}}{\sum_{\bar{v}_\alpha}\partial_\alpha\Lambda(v)^{2-p}} \tag{4.43}$$

$$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha) + \frac{Z_\alpha(v_\alpha)^p\sum_{\bar{v}_\alpha}\rho\left(X(v_0) - \hat{X}(v_0)\right)\hat{X}(v_0)^{-p}\partial_\alpha\Lambda(v)}{\sum_{\bar{v}_\alpha}\partial_\alpha\Lambda(v)^{2-p}}$$

where, for $p = 1$ it recovers the update for KL derived in Section 4.2.

4.3.2. Multiplicative Updates for ML Estimate

Although the EM update can also be formulated in the multiplicative form as

$$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha)\frac{\sum_{\bar{v}_\alpha}\left(\rho\left(X(v_0) - \hat{X}(v_0)\right)\hat{X}(v_0)^{-p} + \Lambda(v)^{1-p}\right)\partial_\alpha\Lambda(v)}{\sum_{\bar{v}_\alpha}\Lambda(v)^{1-p}\partial_\alpha\Lambda(v)} \tag{4.44}$$

in general this is different from the *multiplicative update rule* (MUR) as in the sense of [5] due to the subtraction (potentially resulting to a negative value) in the numerator in Equation 4.44. The MUR equations were popularized by [5] for the NMF and extensively analyzed [32]. They ensure the non-negative parameter updates as long as starting with the non-negative initial values. For the PLTF models, the MUR update equation is given as

$$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha)\frac{\sum_{\bar{v}_\alpha} X(v_0)\hat{X}(v_0)^{-p}\partial_\alpha\Lambda(v)}{\sum_{\bar{v}_\alpha} \hat{X}(v_0)^{1-p}\partial_\alpha\Lambda(v)} \tag{4.45}$$

while this coincides with the EM update for the KL case ($p = 1$) it differs for general β-divergence. This equation successfully recovers the NMF updates in [5] and update for CP for β-divergence in [32]. The realization of the EM_{ML} and MUR_{ML} update equations for $p = \{0, 1, 2\}$ are given in Table 4.3. Further details can be found in 3.6.2.

The multiplicative update rules MUR_{ML} can be obtained by plugging the heuristic

$$\Lambda(v)^{1-p} = \rho\hat{X}(v_0)^{1-p} \tag{4.46}$$

in EM_{ML} update in Equation 4.44. A mathematical justification of this substitution would prove directly the convergence of MUR; while the convergence for the matrix factorization case is given by [49].

Figure 4.3 illustrates the monotone increase of the likelihood after each update iteratively for a TUCKER3 model for the EM and the MUR. For illustrative purposes, we only update an arbitrary single specific element (namely, $Z_2^{5,3}$) and compute the conditional likelihood for various p. The computation of the likelihood $\mathcal{L}(X|Z_{1:N})$ is based on the β-divergence in Equation 4.40 by exploiting the duality between the likelihood and the divergence [39] as $\mathcal{L}(X|Z_{1:N}) \equiv d_\beta(X, \hat{X})$ noting that $\hat{X}$ is a function of $Z_{1:N}$ and the divergence is decomposable as $d_\beta(X, \hat{X}) = \sum_{v_0} d_\beta\Big(X(v_0), \hat{X}(v_0)\Big)$. As expected, for $p = 1$ (KL cost) both method give identical likelihood for each iteration

(as MUR and EM are identical for the KL divergence). We observe monotonic convergence for both methods; for EM this follows directly from the theory, while for MUR with arbitrary p a formal proof is given by [49] for the matrix factorization case.

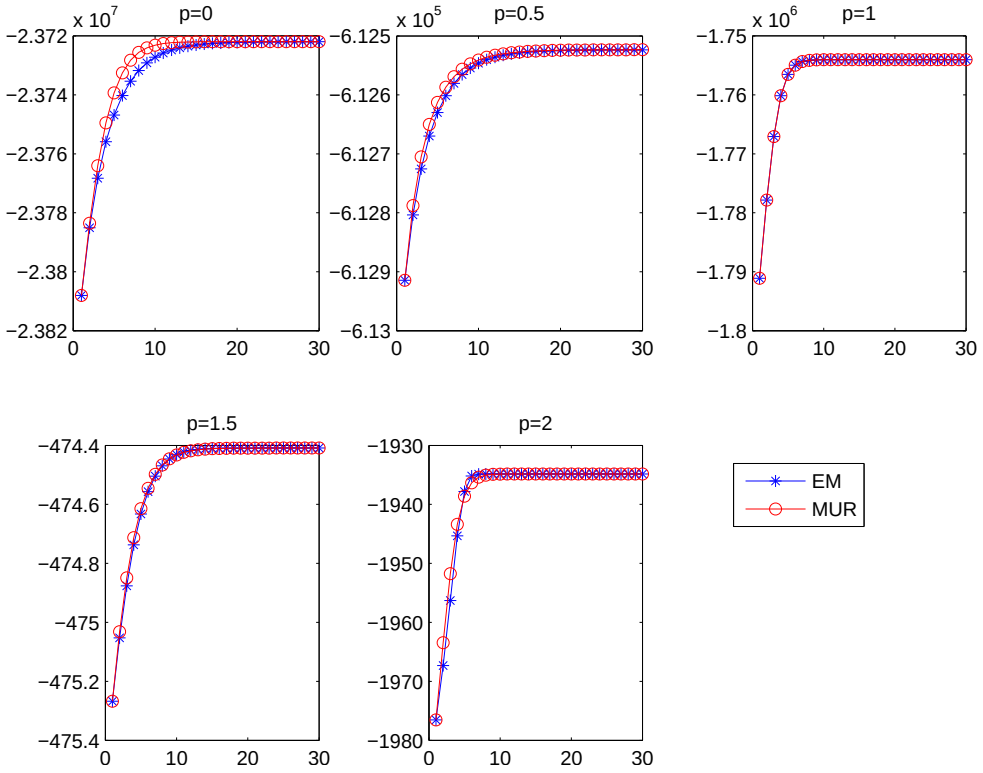


Figure 4.3. This figure illustrates the monotone increase of the likelihood after each update iteratively for a TUCKER3 model for the EM and the MUR.

Example 4.6. *MUR updates for the factor Z_1 for KL and EU costs for the CP factorization model are as follow*

$$Z_1^{i,r} \leftarrow Z_1^{i,r} \frac{\sum_{j,k}(X^{i,j,k}/\hat{X}^{i,j,k}) Z_2^{j,r} Z_3^{k,r}}{\sum_{j,k} Z_2^{j,r} Z_3^{k,r}} \qquad \textit{for KL} \qquad (4.47)$$

$$Z_1^{i,r} \leftarrow Z_1^{i,r} \frac{\sum_{j,k} X^{i,j,k} Z_2^{j,r} Z_3^{k,r}}{\sum_{j,k} \hat{X}^{i,j,k} Z_2^{j,r} Z_3^{k,r}} \qquad \textit{for EU} \qquad (4.48)$$

Example 4.7. *MUR updates for the factors Z_1 and Z_4 (core tensor) for EU cost for*

the TUCKER3 factorization model are as follow

$$Z_1^{i,p} \leftarrow Z_1^{i,p} \frac{\sum_{j,k,q,r} (X^{i,j,k} / \hat{X}^{i,j,k}) Z_2^{j,q} Z_3^{k,r} Z_4^{p,q,r}}{\sum_{j,k,q,r} Z_2^{j,q} Z_3^{k,r} Z_4^{p,q,r}} \tag{4.49}$$

$$Z_4^{p,q,r} \leftarrow Z_4^{p,q,r} \frac{\sum_{i,j,k} (X^{i,j,k} / \hat{X}^{i,j,k}) Z_1^{i,p} Z_2^{j,q} Z_3^{k,r}}{\sum_{i,j,k} Z_1^{i,p} Z_2^{j,q} Z_3^{k,r}} \tag{4.50}$$

Table 4.3. EM updates (multiplicative form), MUR updates and posterior expectations are listed.

	p	$v(\Lambda(v))$	EM Update	ρ	$\langle S(v)\|X(v_0)\rangle$
β		$\Lambda(v)^p$	$Z_\alpha(v_\alpha) = \frac{\sum \langle S(v)\|X(v_0)\rangle \, \partial_\alpha \Lambda(v)^{1-p}}{\sum \partial_\alpha \Lambda(v)^{2-p}}$	$\frac{\varphi_x}{\varphi_s}$	$\Lambda(v) + \rho \frac{\Lambda(v)^p}{\hat{X}(v_0)^p} \Big(X(v_0) - \hat{X}(v_0) \Big)$
EU	0	1	$Z_\alpha(v_\alpha) = \frac{\sum \langle S(v)\|X(v_0)\rangle \, \partial_\alpha \Lambda(v)}{\sum \partial_\alpha \Lambda(v)^2}$	$\frac{1}{K}$	$\Lambda(v) + \frac{1}{K} \Big(X(v_0) - \hat{X}(v_0) \Big)$
KL	1	$\Lambda(v)$	$Z_\alpha(v_\alpha) = \frac{\sum \langle S(v)\|X(v_0)\rangle}{\sum \partial_\alpha \Lambda(v)}$	1	$\Lambda(v) \frac{X(v_0)}{\hat{X}(v_0)}$
IS	2	$\Lambda(v)^2$	$Z_\alpha(v_\alpha) = \frac{\sum \langle S(v)\|X(v_0)\rangle \, \partial_\alpha \Lambda(v)^{-1}}{\sum 1}$	$\frac{\hat{X}(v_0)}{\Lambda(v)}$	$\Lambda(v) + \frac{\Lambda(v)}{\hat{X}(v_0)} (X(v_0) - \hat{X}(v_0))$
	EM Update			MUR Update	
	$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha) \frac{\sum_{\bar{v}_\alpha} \Big(\rho \big(X(v_0) - \hat{X}(v_0) \big) \hat{X}(v_0)^{-p} + \Lambda(v)^{1-p} \Big) \partial_\alpha \Lambda(v)}{\sum_{\bar{v}_\alpha} \Lambda(v)^{1-p} \partial_\alpha \Lambda(v)}$			$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha) \frac{\sum_{\bar{v}_\alpha} X(v_0) \hat{X}(v_0)^{-p} \, \partial_\alpha \Lambda(v)}{\sum_{\bar{v}_\alpha} \hat{X}(v_0)^{1-p} \, \partial_\alpha \Lambda(v)}$	
EU	— $Z_\alpha(v_\alpha) \leftarrow \frac{\sum_{\bar{v}_\alpha} \big(\Lambda(v) + \frac{1}{K} (X(v_0) - \hat{X}(v_0)) \big) \, \partial_\alpha \Lambda(v)}{\sum_{\bar{v}_\alpha} \, \partial_\alpha \Lambda(v)^2}$			$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha) \frac{\sum_{\bar{v}_\alpha} X(v_0) \, \partial_\alpha \Lambda(v)}{\sum_{\bar{v}_\alpha} \hat{X}(v_0) \, \partial_\alpha \Lambda(v)}$	
KL	$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha) \frac{\sum_{\bar{v}_\alpha} X(v_0) \hat{X}(v_0)^{-1} \partial_\alpha \Lambda(v)}{\sum_{\bar{v}_\alpha} \, \partial_\alpha \Lambda(v)}$			$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha) \frac{\sum_{\bar{v}_\alpha} X(v_0) \hat{X}(v_0)^{-1} \, \partial_\alpha \Lambda(v)}{\sum_{\bar{v}_\alpha} \, \partial_\alpha \Lambda(v)}$	
IS	$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha) \frac{\sum_{\bar{v}_\alpha} \big(X(v_0) - \hat{X}(v_0) \big) \hat{X}(v_0)^{-1} + 1}{\sum_{\bar{v}_\alpha} \, 1}$			$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha) \frac{\sum_{\bar{v}_\alpha} X(v_0) \hat{X}(v_0)^{-2} \, \partial_\alpha \Lambda(v)}{\sum_{\bar{v}_\alpha} \hat{X}(v_0)^{-1} \, \partial_\alpha \Lambda(v)}$	

4.3.3. Direct Solution via Alternating Least Squares

The method of *alternating least squares* (ALS) attempts to solve the estimation problem for the EU cost ($p = 0$) by optimizing one variable while keeping the others fixed [32]. It is a direct solution obtained by equating the gradient to zero and solving directly as

$$\sum_{\bar{v}_\alpha} X(v_0)\partial_\alpha \Lambda(v) = \sum_{\bar{v}_\alpha} \hat{X}(v_0)\partial_\alpha \Lambda(v) \tag{4.51}$$

In matricization section we show how this is practically implemented by use of the pseudo-inverse operation.

4.3.4. Update Rules for MAP Estimation

We can incorporate prior belief as in the form of conjugate prior

$$(Z_\alpha(v_\alpha)|N_\alpha^0(v_\alpha), Z_\alpha^0(v_\alpha))$$

with $N_\alpha^0(v_\alpha)$ and $Z_\alpha^0(v_\alpha)$ being the hyperparameters for $Z_\alpha(v_\alpha)$. As specified in [68], $N_\alpha^0(v_\alpha)$ might be thought of a prior sample size while $Z_\alpha^0(v_\alpha)$ is a prior expectation of the mean parameter. The exact definition of $N_\alpha^0(v_\alpha)$ and $Z_\alpha^0(v_\alpha)$ can be identified for each distribution separately by following the definition of a conjugate prior given in 3.6.3. The EM_{MAP} estimate is as

$$Z_\alpha(v_\alpha) = \frac{N_\alpha^0(v_\alpha)Z_\alpha^0(v_\alpha) + \sum_{\bar{v}_\alpha} \langle S(v)|X(v_0)\rangle \varphi_s \partial_\alpha \Lambda(v)^{1-p}}{N_\alpha^0(v_\alpha) + \sum_{\bar{v}_\alpha} \varphi_s \partial_\alpha \Lambda(v)^{2-p}} \tag{4.52}$$

$$\begin{aligned} Z_\alpha(v_\alpha) \leftarrow & \frac{N_\alpha^0(v_\alpha)Z_\alpha^0(v_\alpha)}{N_\alpha^0(v_\alpha) + \sum_{\bar{v}_\alpha} \varphi_s \partial_\alpha \Lambda(v)^{2-p}} \\ & + \frac{Z_\alpha(v_\alpha) \sum_{\bar{v}_\alpha} \left\{ \varphi_s \partial_\alpha \Lambda(v)^{1-p} + Z_\alpha(v_\alpha)^{p-1} \varphi_x (X(v_0) - \hat{X}(v_0))\hat{X}(v_0)^{-p} \right\} \partial_\alpha \Lambda(v)}{N_\alpha^0(v_\alpha) + \sum_{\bar{v}_\alpha} \varphi_s \partial_\alpha \Lambda(v)^{2-p}} \end{aligned}$$

while the multiplicative update rules MUR_{MAP} become

$$Z_\alpha(v_\alpha) \leftarrow \frac{N^0_\alpha(v_\alpha)Z^0_\alpha(v_\alpha) + Z_\alpha(v_\alpha)^p \sum_{\bar{v}_\alpha} \varphi_x X(v_0)\hat{X}(v_0)^{-p}\partial_\alpha\Lambda(v)}{N^0_\alpha(v_\alpha) + Z_\alpha(v_\alpha)^{p-1} \sum_{\bar{v}_\alpha} \varphi_x \hat{X}(v_0)^{1-p}\partial_\alpha\Lambda(v)} \tag{4.53}$$

Prior belief can always be specified in terms of $N^0_\alpha(v_\alpha)$ and $Z^0_\alpha(v_\alpha)$. However, if prior distribution is to be specified explicitly, they can be tied to the prior parameters similar to [72].

Example 4.8. *The gamma distribution is the conjugate prior for the Poisson $p(S|Z)$ as*

$$p(Z_\alpha(v_\alpha)|N^0_\alpha(v_\alpha), Z^0_\alpha(v_\alpha)) = \mathcal{G}(Z_\alpha(v_\alpha)|A_\alpha(v_\alpha), B_\alpha(v_\alpha)) \tag{4.54}$$

with

$$N^0_\alpha(v_\alpha) = B_\alpha(v_\alpha) \qquad\qquad N^0_\alpha(v_\alpha)Z^0_\alpha(v_\alpha) = A_\alpha(v_\alpha) - 1 \tag{4.55}$$

which for $p = 1$ EM_{MAP} and MUR_{MAP} successfully recover the update for KL in Equation 4.30 in Section 4.2.

Example 4.9. *The Gaussian distribution is the conjugate prior for the Gaussian $p(S|Z, \Sigma)$ with unknown mean and known variance Σ with*

$$p(Z_\alpha(v_\alpha)|N^0_\alpha(v_\alpha), Z^0_\alpha(v_\alpha)) = \mathcal{N}\left(Z_\alpha(v_\alpha)|A_\alpha(v_\alpha), B_\alpha(v_\alpha)\right) \tag{4.56}$$

$$N^0_\alpha(v_\alpha) = \Sigma/B_\alpha(v_\alpha) \qquad\qquad Z^0_\alpha(v_\alpha) = A_\alpha(v_\alpha) \tag{4.57}$$

4.3.5. Missing Data and Tensor Forms

It is straightforward to handle missing data in EM and MUR updates. We recall that the scalar value $M(v_0)$ is, indeed, simply a multiplier in the sum $\sum_{\bar{v}_\alpha}$ of the initial

M-step equation as

$$\sum_{\bar{v}_\alpha} M(v_0)\Big(\langle S(v)|X(v_0)\rangle - \Lambda(v)\Big)\varphi_s\Lambda(v)^{-p}\partial_\alpha\Lambda(v)$$

Hence we simply put it back inside the sums $\sum_{\bar{v}_\alpha}$. For example the MUR_{ML} update turns to be

$$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha)\frac{\sum_{\bar{v}_\alpha} M(v_0)X(v_0)\hat{X}(v_0)^{-p}\,\partial_\alpha\Lambda(v)}{\sum_{\bar{v}_\alpha} M(v_0)\hat{X}(v_0)^{1-p}\,\partial_\alpha\Lambda(v)} \tag{4.58}$$

As we did in Section 4.2 we represent all the update equations in tensor forms by use of the Δ_α abstraction. In Table 4.4, we give a summary of general update rules in tensor forms. For alternating projections, considering the least square approximation, it is natural to set $p = 0$ and by the use of the pseudo-inverse. We need to solve $\Delta_\alpha(X) = \Delta_\alpha(\hat{X})$. This equation leads to a linear matrix equation of form $L\ X\ R = L\ \hat{X}\ R$ where L and R are structured matrices determined by the form of the TF model. This equation can be solved via pseudo-inverses in a least square sense, when there is no missing data (i.e. $M = \mathbf{1}$). For general M, we could not derive a compact equation without considering tedious reindexings.

Table 4.4. Table lists the updates in tensor forms via $\Delta_\alpha()$ function.

EM	Q_1, Q_2	$Q_1 = M \circ \left(X - \hat{X}\right) \circ \hat{X}^{-p} \qquad Q_2 = M$
	ML	$Z_\alpha \leftarrow Z_\alpha + \frac{Z_\alpha^p \circ \Delta_\alpha(\rho Q_1)}{\Delta_\alpha^{2-p}(Q_2)}$
	MAP	$Z_\alpha \leftarrow \frac{N_\alpha^0 \circ Z_\alpha^0 + Z_\alpha \circ \Delta_\alpha^{1-p}(Q_2) + Z_\alpha^p \circ \Delta_\alpha(\varphi_x Q_1)}{N_\alpha^0 + \Delta_\alpha^{2-p}(\varphi_s Q_2)}$
MUR	Q_1, Q_2	$Q_1 = M \circ X \circ \hat{X}^{-p} \qquad Q_2 = M \circ \hat{X}^{1-p}$
	ML	$Z_\alpha \leftarrow Z_\alpha \circ \frac{\Delta_\alpha(Q_1)}{\Delta_\alpha(Q_2)}$
	MAP	$Z_\alpha \leftarrow \frac{N_\alpha^0 \circ Z_\alpha^0 + Z_\alpha^p \circ \Delta_\alpha(\varphi_x Q_1)}{Z_\alpha^0 + Z_\alpha^{p-1} \circ \Delta_\alpha(\varphi_x Q_2)}$
ALS	ML	solve $\Delta_\alpha(X) = \Delta_\alpha(\hat{X})$ assuming $M = \mathbf{1}$

4.4. Representation and Implementation

For the tensorized forms of the update equations such as the KL update

$$Z_\alpha = Z_\alpha \circ \Delta_\alpha(M \circ X/\hat{X})/\Delta_\alpha(M)$$

the main task is the computation of the $\Delta_\alpha()$ function. Below, we define a matricization procedure that converts $\Delta_\alpha()$ (any element-wise equation indeed) into the matrix form in terms of matrix multiplication abstracts such as the Kronecker product and Khati-Rao product.

4.4.1. Matricization

Matricization as defined originally in [12, 13] is the operation of converting a multiway array into a matrix by reordering the column fibers. In this paper we refer to this definition as *unfolding* and refer to *matricization* as the procedure to convert an element-wise equation into a corresponding matrix form. We use Einstein's summation convention where repeated indices are added over. The conversion rules are given in Table 4.5. Our notation is best illustrated with an example: consider a matrix $X^{i,j}$ with row index i and column index j. If we assume a column by column memory layout, we refer to the vectorization of $\mathbf{vec}\, X$ (vertical concatenation of columns) as X^{ji}; adopting a *the faster index last* convention and we drop the comma. Here i is the faster index since when traversing the elements of the matrix X in sequence i changes more rapidly. With this, we arrive at the following definition.

Definition 4.2. *Consider a multiway array $X \in \mathbb{R}^{I_1 \times \ldots \times I_M}$ with a generic element denoted by $X^{i_1,i_2,\ldots,i_M}$. The mode-n unfolding of X is the matrix $X_{(n)} \in \mathbb{R}^{I_n \times \prod_{k \neq n} I_k}$ with row index i_n as*

$$X_{(n)} \equiv X_{i_n}^{i_M \ldots i_{n-1} i_{n+1} \ldots i_2 i_1} \tag{4.59}$$

where the fastest index is in the order $i_i, i_2, \ldots, i_M$.

Here we follow the natural ordering, that is, for the mode-1 unfolding, mode-2 rows are placed before mode-3 rows and similarly for the mode-2 unfolding, mode-1 rows are placed before mode-3 rows and so on. Hence during matricization, we start with scalar terms, and we freely reorder the terms inside the sum, transpose them, unfold them, join them by using Kronecker and Khati-Rao product to get the desired indices (the outcome indices). In addition, if needed we may introduce ones matrix $\mathbf{1}$ of any size when the indices of the terms are insufficient as $Y_1^q = \sum_p Y_p^q = \sum_p \mathbf{1}_1^p Y_p^q = (\mathbf{1}Y)_1^q$ where the index p to be marginalize out.

Table 4.5. Index notation used to matricize an element-wise equation into the matrix form. Following Einstein convention, duplicate indices are summed over. Khatri-Rao product and mode-n unfolding are implemented in N-way Toolbox [73].

Equivalence	**Matlab**	**Remark**
$X_i^j \equiv X$	X	Matrix notation
$X_i^{kj} \equiv X_{(1)}$	$nshape(X, 1)$	Array (mode-1 unfolding)
$X_i^j \equiv (X^T)_j^i$	X'	Transpose
$\mathbf{vec}\, X_i^j \equiv (X)_{ji}^1$	$X(:)$	Vectorize
$X_i^j Y_j^p \equiv (XY)_i^p$	$X * Y$	Matrix product
$X_i^p Y_j^p \equiv (X \odot Y)_{ij}^p$	$krb(X, Y)$	Khatri-Rao product
$X_i^p Y_j^q \equiv (X \otimes Y)_{ij}^{pq}$	$kron(X, Y)$	Kronecker product
$X_i^j X_i^j \equiv (X \circ X)_i^j$	$X. * X$	Hadamard product

Here are the examples of matricization, MUR and ALS updates. Table 4.6 summarizes updates for known models. To get rid of the annoying subindices, here we relax the notation by using A, B, C, G for Z_α.

Example 4.10. *Mode-1 unfolding $\hat{X}_{(1)}$ for the TUCKER3 factorization is as follows*

$$\begin{aligned}
\hat{X}^{i,j,k} &= \sum_{pqr} G^{p,q,r} A^{i,p} B^{j,q} C^{k,r} && \text{start with element-wise equation} \\
(\hat{X}_{(1)})_i^{kj} &= (G_{(1)})_p^{rq} A_i^p B_j^q C_k^r && \text{place row/column indices} \\
&= (AG_{(1)})_i^{rq} (C \otimes B)_{kj}^{rq} && \text{reorder and form Kronecker product} \\
&= \left((AG_{(1)})(C \otimes B)^T\right)_i^{kj} && \text{transpose and drop matching indices} \\
\hat{X}_{(1)} &= AG_{(1)}(C \otimes B)^T && \text{remove remaining indices}
\end{aligned}$$

Example 4.11. *Similarly, mode-2 unfolding $\hat{X}_{(2)}$ is also handled smoothly as*

$$(\hat{X}_{(2)})_j^{ki} = (BG_{(2)})_j^{rq} \left((C \otimes A)^T\right)_{rq}^{ki} \tag{4.60}$$

$$\hat{X}_{(2)} = BG_{(2)}(C \otimes A)^T \tag{4.61}$$

Example 4.12. *For the update equations of TUCKER3 factorization, the $\Delta()$ functions $\Delta_A(Q)$ and $\Delta_G(Q)$ are as follows*

$$\Delta_A(Q) = (Q_{(1)})_i^{kj} B_j^q C_k^r G_p^{rq} = Q_{(1)}(C \otimes B)G_{(1)}^T \tag{4.62}$$

$$\Delta_G(Q) = (Q_{(1)})_i^{kj} A_i^p B_j^q C_k^r = A^T Q_{(1)}(C \otimes B) \tag{4.63}$$

- *If MUR_{ML} the general format of the update equation is $Z_\alpha \leftarrow Z_\alpha \circ \frac{\Delta_\alpha(Q_1)}{\Delta_\alpha(Q_2)}$ with $Q_1 = M \circ X \circ \hat{X}^{-p}$ and $Q_2 = M \circ \hat{X}^{1-p}$. Then*

$$A = A \circ \frac{Q_{1_{(1)}}(C \otimes B)G_{(1)}^T}{Q_{2_{(1)}}(C \otimes B)G_{(1)}^T} \qquad G_{(1)} = G_{(1)} \circ \frac{A^T Q_{1_{(1)}}(C \otimes B)}{A^T Q_{2_{(1)}}(C \otimes B)} \tag{4.64}$$

  where, for example, for KL ($p = 1$) we evaluate $Q_1 = M \circ (X/\hat{X})$ and $Q_2 = M$.
- *If MUR_{MAP} the update for the factor A with the hyperparameters $N_\alpha^0(v_\alpha), Z_\alpha^0(v_\alpha)$ is as below. Here Q_1, Q_2 are as $Q_1 = M \circ X \circ \hat{X}^{-p}$ and $Q_2 = M \circ \hat{X}^{1-p}$ as for*

the MUR_{ML}. A sample implementation is given in Algorithm 4.4.1.

$$A = \frac{N_\alpha^0(v_\alpha) Z_\alpha^0(v_\alpha) + A^p \circ \left(\varphi_x Q_{1_{(1)}} (C \otimes B) G_{(1)}^T\right)}{N_\alpha^0(v_\alpha) + A^{p-1} \circ \left(\varphi_x Q_{2_{(1)}} (C \otimes B) G_{(1)}^T\right)} \tag{4.65}$$

- *If ALS (for $p = 0$) and assuming no missing values we solve $X = \hat{X}$ for the core tensor G*

$$X_{(1)} = \hat{X}_{(1)} = A G_{(1)} (C \otimes B)^T \qquad \Rightarrow \qquad G_{(1)} = A^\dagger X_{(1)} \left((C \otimes B)^T\right)^\dagger \tag{4.66}$$

with $X^\dagger = X^T (X X^T)^{-1}$. Here we simply move all the unrelated factors to the other side of the equation while taking their pseudo-inverses.

Example 4.13. *After the factorization process ends we might consider factorization of the latent tensors deeper. PARATUCK2 [13, 32, 74] is a nice example to illustrate this. After inserting the scalar value 1_k^k PARATUCK2 model can be matricized as*

$$\hat{X}^{i,j,k} = \sum_{pq} A^{i,p} B^{j,q} F^{p,q,k} \tag{4.67}$$

$$\hat{X}_{(1)} = \left[A_i^p (B^T)_q^j F_p^{kq} \mathbf{1}_k^k\right] = A F_{(1)} (\mathbf{1} \otimes B^T) \tag{4.68}$$

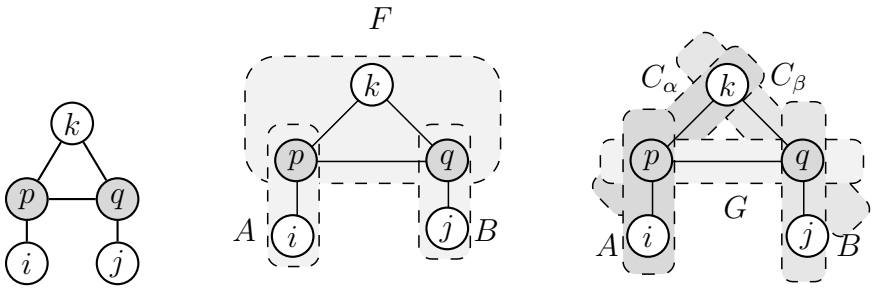


Figure 4.4. PARATUCK2 factorization model illustrates re-factorization of one of the latent tensors. The model output is $\hat{X}^{i,j,k} = \sum_{pq} G^{p,q} A^{i,p} B^{j,q} C_\alpha^{k,p} C_\beta^{k,q}$. The shaded nodes in the graph are latent.

Then, we consider further factorization of F, as below where at the last step we

transpose twice.

$$F_p^{kq} \equiv G_p^q C_{a_k}^p C_{b_k}^q \tag{4.69}$$

$$= ((C_a \odot G^T)^T)_p^{kq} (C_b^T)_1^{kq} \tag{4.70}$$

$$= \left((C_a \odot G^T)^T) \odot (C_b^T)\right)_p^{kq} \tag{4.71}$$

$$= \left((C_a \odot G^T) \odot C_b\right)^T \tag{4.72}$$

Table 4.6. The table lists model output, the Δ_α function, the MUR and ALS updates for the typical components of NMF, CP, TUCKER3 and PARATUCK2 models. For PARATUCK2 F is given as $F_{(1)} = \left((C_a \odot G^T) \odot C_b\right)^T$. The symbols $\otimes$, $\odot$ and $\circ$ are for Knonecker, Khatri-Rao and Hadamard products in the order and the division is of the element-wise type.

	Model	$\Delta_\alpha()$	**MUR Update**	**ALS** (for $p = 0$)
NMF	$\hat{X} = AB$	$\Delta_A(Q) = QB^T$	$A = A \circ \frac{Q_1 B^T}{Q_2 B^T}$	$A = XB^\dagger$
CP	$\hat{X}_{(1)} = A(C \odot B)^T$	$\Delta_A(Q) = Q_{(1)}(C \odot B)$	$A = A \circ \frac{Q_{1_{(1)}}(C \odot B)}{Q_{2_{(1)}}(C \odot B)}$	$A = X_{(1)}((C \odot B)^T)^\dagger$
TUCKER3	$\hat{X}_{(1)} = AG_{(1)}(C \otimes B)^T$	$\Delta_A(Q) = Q_{(1)}(C \otimes B)G_{(1)}^T$	$A = A \circ \frac{Q_{1_{(1)}}(C \otimes B)G_{(1)}^T}{Q_{2_{(1)}}(C \otimes B)G_{(1)}^T}$	$A = X_{(1)}\left(G_{(1)}(C \otimes B)^T\right)^\dagger$
		$\Delta_G(Q) = A^T Q_{(1)}(C \otimes B)$	$G_{(1)} = G_{(1)} \circ \frac{A^T Q_{1_{(1)}}(C \otimes B)}{A^T Q_{2_{(1)}}(C \otimes B)}$	$G_{(1)} = A^\dagger X_{(1)}\left((C \otimes B)^T\right)^\dagger$
PARATUCK2	$\hat{X}_{(1)} = AF_{(1)}(\mathbf{1} \otimes B^T)$	$\Delta_A(Q) = Q_{(1)}(\mathbf{1} \otimes B)F^T$	$A = A \circ \frac{Q_{1_{(1)}}(\mathbf{1} \otimes B)F^T}{Q_{2_{(1)}}(\mathbf{1} \otimes B)F^T}$	

Input: X, N0, Z0, Z, M, MAXITER, p, dX, N. Output: Z (latent factors)

for $e = 1$ to MAXITER **do**

  for $a = 1$ TO N **do**

    $\hat{X} = reshape(Z_1 * nshape(Z_4, 1) * (Z_3 \otimes Z_2)^T, i, j, k)$

    $Q_1 = dX * M \circ X \circ \hat{X}^{-p}$;

    $Q_2 = dX * M \circ \hat{X}^{1-p}$;

    if $(a == 1)$ **then**

      $UP = (nshape(Q_1, a) * (Z_3 \otimes Z_2) * nshape(Z_4, a)^T)$;

      $DW = (nshape(Q_2, a) * (Z_3 \otimes Z_2) * nshape(Z_4, a)^T)$;

    else if $(a == 2, 3)$ **then**

      similar to the condition $a == 1$

    else if $(a == 4)$ **then**

      $UP = Z_1^T * nshape(Q_1, 1) * (Z_3 \otimes Z_2)$;

      $DW = Z_1^T * nshape(Q_2, 1) * (Z_3 \otimes Z_2)$;

      $UP = reshape(UP, p, q, r)$;

      $DW = reshape(DW, p, q, r)$;

    end if

    $UP = N0_a \circ Z0_a + Z_a^p \circ UP$;

    $DW = N0_a + Z_a^{p-1} \circ DW$;

    $Z_a = UP./DW$;

  end for

end for

Figure 4.5. Matlab-like implementation for TUCKER3 MUR_{MAP} update for β-divergence. Only the update for Z_1 and Z_4 is shown.

4.4.2. Junction Tree for the Factors of TUCKER3

We already pointed out the close relation between TF and Graphical Models (GM). Here we illustrates with an example the relation between matricization in TF and *junction tree* in GM. For the illustration example we use the first factor A (or $A^{i,p}$ in element-wise) of the TUCKER3 factorization model. Recall the delta function for

$A^{i,p}$ in element-wise form and in tensor forms

$$A^{i,p} \equiv \sum_{jkqr} X^{i,j,k} B^{j,q} C^{k,r} G^{p,q,r} = \sum_{jk} X^{i,j,k} \sum_{r} C^{k,r} \sum_{q} B^{j,q} G^{p,q,r} \tag{4.73}$$

$$\Delta_A(X) = X_{(1)}(C \otimes B) G_{(1)}^T \tag{4.74}$$

Figure 4.6 illustrates the construction of the junction tree (JT) for the factor A of TUCKER3 factorization model. For JT construction we need to choose any random ordering of nodes where here in this example the node ordering is i, j, k, p, q, r in the order. The down subfigure illustrates the 1-1 correspondence between matricization via Δ function and JT message passing. Here in this subfigure the Kronecker product is represented as merging the two cliques.

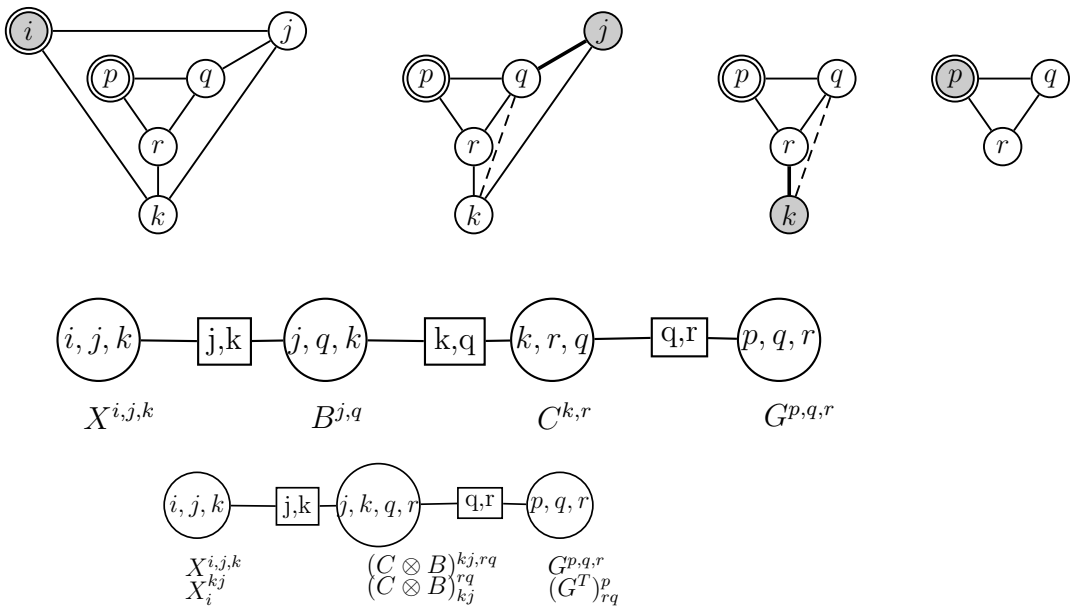


Figure 4.6. (Top) Construction of junction tree for factor A of TUCKER3. Node ordering is i, j, k, p, q, r in the order. (Middle) Associated junction tree. (Down) Message passing view of Delta Δ_A function for component A. There is 1-1 correspondence between matricization via Δ function and JT message passing. Below figure illustrates merging of the cliques.

4.5. Discussion

In Section 4.3 we derived the update equations for EU, KL and IS costs in a single expression by using the following posterior expectation $\langle S(v)|X(v_0)\rangle$ equation in the E-step

$$\langle S(v)|X(v_0)\rangle = \Lambda(v) + \rho\frac{\Lambda(v)^p}{\hat{X}(v_0)^p}\Big(X(v_0) - \hat{X}(v_0)\Big) \qquad \text{for } p = 0, 1, 2 \tag{4.75}$$

We note that this equation can be generalized for $p \geq 0$ by adapting the development in Chapter 3 as

$$\langle S(v)|X(v_0)\rangle = \Lambda(v) + \frac{\Lambda(v)^p}{\sum_{\bar{v}_0}\Lambda(v)^p}\Big(X(v_0) - \hat{X}(v_0)\Big) \tag{4.76}$$

The M-step remains the same. Hence by plugging the posterior expectation in the M-step we come up with a generic update equation for broader values of p, such as for the cases $p \in (1, 2)$ (the compound Poisson) or for $p = 3$ (inverse Gaussian).

4.6. Summary

In this chapter, we have developed a probabilistic framework for multiway analysis of high dimensional datasets. We start by introducing a notation that can represent any arbitrary factorization structure similar to the graphical models illustrated by the example for well-known model CP as the setting $N = 3$, $\mathcal{V} = \{i, j, k, r\}$, $\mathcal{V}_0 = \{i, j, k\}$, $\mathcal{V}_1 = \{i, r\}$, $\mathcal{V}_2 = \{j, r\}$ and $\mathcal{V}_3 = \{k, r\}$. Then we obtained the fixed point update equations for the latent factors. For the KL cost is is as

$$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha)\frac{\sum_{\bar{v}_\alpha} M(v_0)\frac{X(v_0)}{\hat{X}(v_0)}\partial_\alpha\Lambda(v)}{\sum_{\bar{v}_\alpha} M(v_0)\partial_\alpha\Lambda(v)} \tag{4.77}$$

where the model estimate is

$$\hat{X}(v_0) = \sum_{\bar{v}_0} \prod_{\alpha} Z_\alpha(v_\alpha) \tag{4.78}$$

Then we define a tensor valued function $\Delta_\alpha(Q) : \mathbb{R}^{|Q|} \rightarrow \mathbb{R}^{|Z_\alpha|}$, which is an important abstraction similar to the message passing of probabilistic graphical models. By use of it we express any fixed point update equation in terms of the Δ function

$$\Delta_\alpha^p(Q) = \left[\sum_{\bar{v}_\alpha} Q(v_0) \ (\partial_\alpha \Lambda(v))^p \right] \tag{4.79}$$

that simplifies the update for KL as follows

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\Delta_\alpha(M \circ X/\hat{X})}{\Delta_\alpha(M)} \tag{4.80}$$

Here M is the mask tensor for missing data. The use of prior knowledge in the form of conjugate priors is also introduced into the update equations. The following example is for KL cost for arbitrary structure where A, B are parameters of the gamma prior

$$Z_\alpha(v_\alpha) \leftarrow \frac{(A_\alpha(v_\alpha) - 1) + Z_\alpha(v_\alpha) \sum_{\bar{v}_\alpha} M(v_0) \frac{X(v_0)}{\hat{X}(v_0)} \partial_\alpha \Lambda(v)}{B_\alpha(v_\alpha) + \sum_{\bar{v}_\alpha} M(v_0) \partial_\alpha \Lambda(v)} \tag{4.81}$$

While the examples so far are for KL cost, as one of the main contribution of this chapter is that we obtained fixed point updates for major class of distributions in the same equation as given in the following ML estimate identified via EM

$$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha) + \frac{Z_\alpha(v_\alpha)^p \sum_{\bar{v}_\alpha} \rho \left(X(v_0) - \hat{X}(v_0) \right) \hat{X}(v_0)^{-p} \partial_\alpha \Lambda(v)}{\sum_{\bar{v}_\alpha} \partial_\alpha \Lambda(v)^{2-p}} \tag{4.82}$$

We also sketched a straightforward matricization procedure to convert element-wise equations into the matrix forms to ease implementation and compact representation. The use of the matricization is simple, easy and powerful that without any use of matrix algebra it is possible to derive the update equations mechanically in the corre-

sponding matrix forms. This is illustrated by the example for mode-2 unfolding $\hat{X}_{(2)}$ of TUCKER3

$$(\hat{X}_{(2)})_j^{ki} = (BG_{(2)})_j^{rq} \left((C \otimes A)^T\right)_{rq}^{ki} \quad \Rightarrow \quad \hat{X}_{(2)} = BG_{(2)}(C \otimes A)^T \tag{4.83}$$

Finally we generalized the approached that we build for KL models and obtained ML and MAP based update equations for other cost functions in the form of beta divergence.

5. MODEL SELECTION FOR NON-NEGATIVE TENSOR FACTORIZATION WITH KL

5.1. Introduction

This chapter constructs a model selection framework for arbitrary non-negative tensor factorization model for KL cost via a variational bound on the marginal likelihood [1, 75]. In this chapter, we explicitly focus on using the KL divergence and non-negative factorizations while the treatment in this chapter can be extended for other error measures and divergences noting that we already outline the general equations for model selection. Our probabilistic treatment generalizes the statistical treatment of NMF models described in [30, 31]. Here we also do asymptotic analysis in case of large data sample where we obtain BIC equations from VB [75] for TF models.

This chapter is organized as follows. Section 5.2 introduces Bayesian model selection and model selection with Variational methods. Section 5.3 describes variation methods for PLTF KL models. Section 5.4 is about computing a lower bound for marginal likelihood. Section 5.5 is about asymptotic analysis of model selection and BIC derivation for PLTF KL models. Then, finally Section 5.6 deals with the implementation issues followed by various experiments.

5.2. Model Selection for $PLTF_{KL}$ Models

For matrix factorization models the model selection problem becomes choosing the model order, i.e. the cardinality of the latent index, whereas for tensor factorization models selecting the right generative model among many alternatives can be a difficult task. The difficulty is due to the fact that it is not clear how to choose (i) the cardinality of the latent indices, (ii) the actual structure of the factorization. For example, given an observation $X^{i,j,k}$ with three indices one can propose a CP generative model as $\hat{X}^{i,j,k} = \sum_r Z_1^{i,r} Z_2^{j,r} Z_3^{k,r}$, or a TUCKER3 model $\hat{X}^{i,j,k} = \sum_{p,q,r} Z_1^{i,p} Z_2^{j,q} Z_3^{k,r} Z_4^{p,q,r}$ or

some arbitrary model as $\hat{X}^{i,j,k} = \sum_{p,q} Z_1^{i,p} Z_2^{j,p} Z_3^{k,q} Z_4^{p,q}$. On the other hand, a CP model is a special kind of TUCKER3 model where the cardinality of the latent indices are the same and the core tensor is superdiagonal [13] and hence a TUCKER3 model can fit to the CP generated data. However, this is arguable since over complex models cause a phenomenon known as over learning (or over fitting).

5.2.1. Bayesian Model Selection

For a Bayesian point of view, a model is associated with a random variable Θ and it interacts with the observed data X simply as $p(\Theta|X) \propto p(X|\Theta)p(\Theta)$. Then, for a model selection task we choose the model associated with Θ^* having the highest posterior probability such that $\Theta^* = \arg\max_\Theta p(\Theta|X)$. Meanwhile, assuming the model priors $p(\Theta)$ are equal the quantity $p(X|\Theta)$ becomes important since comparing $p(\Theta|X)$ is the same as comparing $p(X|\Theta)$. The quantity $p(X|\Theta)$ is called *marginal likelihood* [1] and it is the average over the space of the parameters, in our case, S and Z as [30]

$$p(X|\Theta) = \int_Z dZ \sum_S p(X|S, Z, \Theta)p(S, Z|\Theta) \tag{5.1}$$

In an EM optimization setup S contains the latent variables that are never directly observed and smoothly disappear from the update equations, while Z contains the model parameters that we want to find out.

5.2.2. Model Selection with Variational Methods

On the other hand, computation of this integral is itself a difficult task that requires averaging on several models and parameters. There are several approximation methods such as sampling or deterministic approximations such as Gaussian approximation. One other approximation method is to bound the log marginal likelihood by using variational inference [1,30,40] where an *approximating distribution* q is introduced

into the log marginal likelihood equation

$$\log p(X|\Theta) \geq \int_Z dZ \sum_S q(S,Z) \log \frac{p(X,S,Z|\Theta)}{q(S,Z)} \tag{5.2}$$

where the bound attains its maximum and becomes equal to the log marginal likelihood whenever $q(S,Z)$ is set as $p(S,Z|X,\Theta)$, that is the exact posterior distribution. However, the posterior is usually intractable, and rather, inducing the approximating distribution becomes easier. Here, the approximating distribution q is chosen such that it assumes no coupling between the hidden variables such that it factorizes into independent distributions as $q(S,Z) = q(S)q(Z)$.

5.3. Variational Methods for $PLTF_{KL}$ Models

For model comparison, the Bayesian approach offers an elegant solution based on computing marginal likelihood $p(X|\Theta)$, where latent variables and the parameters are integrated out. As exact computation is intractable, we will resort to standard variational Bayes approximations [1, 40]. The interesting result is that we get a belief propagation algorithm for marginal intensity fields rather than marginal probabilities.

Here we recall the generative Probabilistic Latent Tensor Factorization KL model

$$Z_\alpha(v_\alpha) \sim \mathcal{G}(Z_\alpha; A_\alpha(v_\alpha), B_\alpha(v_\alpha)/A_\alpha(v_\alpha)) \tag{5.3}$$

with the following iterative update equation for the component Z_α obtained via EM

$$Z_\alpha(v_\alpha) \leftarrow \frac{(A_\alpha(v_\alpha) - 1) + Z_\alpha(v_\alpha) \sum_{\bar{v}_\alpha} M(v_0) \frac{X(v_0)}{\hat{X}(v_0)} \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'})}{\frac{A_\alpha(v_\alpha)}{B_\alpha(v_\alpha)} + \sum_{\bar{v}_\alpha} M(v_0) \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'})} \tag{5.4}$$

where $\hat{X}(v_0)$ is the model estimate defined as earlier $\hat{X}(v_0) = \sum_{\bar{v}_0} \prod_\alpha Z_\alpha(v_\alpha)$. We note that the gamma hyperparameters $A_\alpha(v_\alpha)$ and $B_\alpha(v_\alpha)/A_\alpha(v_\alpha)$ are chosen for computational convenience for sparseness representation such that the distribution has a mean $B_\alpha(v_\alpha)$ and standard deviation $B_\alpha(v_\alpha)/\sqrt{A_\alpha(v_\alpha)}$ and for small $A_\alpha(v_\alpha)$ most of

the parameters are forced to be around zero favoring for a sparse representation [30]. Final note is that the gamma distribution convention here is

$$\mathcal{G}(Z_\alpha; A_\alpha(v_\alpha), B_\alpha(v_\alpha)/A_\alpha(v_\alpha)) = \exp\{(A_\alpha(v_\alpha)-1)Z_\alpha(v_\alpha) - (1/B_\alpha(v_\alpha))Z_\alpha(v_\alpha) + \ldots\}$$

Here we formulate the fixed point update equation for the update of the factor Z_α as an expectation of the approximated posterior distribution

Theorem 5.1. *Let A_α and B_α be the prior parameters for the gamma distributed variable Z_α, X is the observation and M is the missing mask. Then approximation distribution for the posterior $q(Z)$ is identified as the gamma distribution with the following parameters*

$$Z_\alpha(v_\alpha) \sim \mathcal{G}(Z_\alpha(v_\alpha); C_\alpha(v_\alpha), D_\alpha(v_\alpha)) \tag{5.5}$$

where the shape and scale parameters are

$$C_\alpha(v_\alpha) = A_\alpha(v_\alpha) + \sum_{\bar{v}_\alpha} M(v_0)\frac{X(v_0)}{\hat{X}_L(v_0)}\prod_\alpha L_\alpha(v_\alpha) \tag{5.6}$$

$$D_\alpha(v_\alpha) = \left(\frac{A_\alpha(v_\alpha)}{B_\alpha(v_\alpha)} + \sum_{\bar{v}_\alpha} M(v_0)\prod_{\alpha'\neq\alpha}\langle Z_{\alpha'}(v_{\alpha'})\rangle\right)^{-1} \tag{5.7}$$

while $L_\alpha(v_\alpha)$ and $\hat{X}_L(v_0)$ are defined as

$$L_\alpha(v_\alpha) = \exp\left(\langle\log Z_\alpha(v_\alpha)\rangle\right) = \exp\left(\psi\big(C_\alpha(v_\alpha)\big)\right)D_\alpha(v_\alpha) \tag{5.8}$$

$$\hat{X}_L(v_0) = \sum_{\bar{v}_0}\prod_\alpha L_\alpha(v_\alpha) \tag{5.9}$$

Here ψ is the digamma function.

Lemma 5.1. *The expectation of the factor Z_α is identified as the mean of the gamma*

distribution and given in the following iterative fixed point update equation

$$\langle Z_\alpha(v_\alpha)\rangle = C_\alpha(v_\alpha)D_\alpha(v_\alpha) \tag{5.10}$$

$$= \frac{A_\alpha(v_\alpha) + L_\alpha(v_\alpha)\sum_{\bar{v}_\alpha} M(v_0)\frac{X(v_0)}{\hat{X}_L(v_0)}\prod_{\alpha'\neq\alpha} L_{\alpha'}(v_{\alpha'}))}{\frac{A_\alpha(v_\alpha)}{B_\alpha(v_\alpha)} + \sum_{\bar{v}_\alpha} M(v_0)\prod_{\alpha'\neq\alpha} E_{\alpha'}(v_{\alpha'})} \tag{5.11}$$

where we take $L_\alpha(v_\alpha)$ out of the sum $\sum_{\bar{v}_\alpha}$ for the numerator to have the appropriate shape to apply Δ function, and in the denominator we used $E_{\alpha'}(v_{\alpha'})$ for notational clearness

$$E_\alpha(v_\alpha) = \langle Z_\alpha(v_\alpha)\rangle = C_\alpha(v_\alpha)D_\alpha(v_\alpha) \tag{5.12}$$

In the following we provide a proof for the Theorem 5.1.

Proof. We lower bound of the marginal likelihood for any arbitrary PLTF$_{KL}$ model based on variational Bayes; while clearly other Bayesian model selection such as based on MCMC can also be used. To bound the marginal log-likelihood, an approximating distribution $q(S, Z)$ over the hidden structure S and Z is introduced as

$$\mathcal{L}(\Theta) = \log p(X|\Theta) \geq \int_Z dZ \sum_S q(S, Z)\log\frac{p(X, S, Z|\Theta)}{q(S, Z)} \tag{5.13}$$

$$= \langle \log p(X, S, Z|\Theta)\rangle_{q(S,Z)} + H[q(S, Z)] \tag{5.14}$$

The bound is tight whenever q equals to the posterior as $q(S, Z) = p(S, Z|X, \Theta)$ but computing the posterior $p(S, Z|X, \Theta)$ is intractable.

At this point variational Bayes suggests approximating q. The simplest selection for q from the family of approximating distribution is the one which poses no coupling for the members of the hidden structure S, Z. That is, we take a factorized

approximation $q(S,Z) = q(S)q(Z)$ such that

$$q(S,Z) = \Big(\prod_{v_0} q\big(S(v_0,*)\big)\Big)\Big(\prod_{\alpha}\prod_{v_\alpha} q\big(Z_\alpha(v_\alpha)\big)\Big) \tag{5.15}$$

where $*$ symbol in $S(v_0,*)$ is used to indicate the slice of the array. That is $S(v_0,*)$ is the slice of latent tensor S as the observed variables in configurations v_0 are being fixed. For example, for TUCKER3 where S is the tensor over the indices i,j,k,p,q,r the slice could be the sub tensor $S(2,3,4,*)$ such that the observed indices i,j,k are fixed as $2,3,4$ while the unobserved indices ranges for the full configuration. Then,

$$q^{(n+1)}_{S(v_0,*)} \propto \exp\Big(\langle \log p(X,S,Z|\Theta)\rangle_{q^{(n)}/q_{S(v_0,*)}}\Big) \tag{5.16}$$

$$q^{(n+1)}_{Z_\alpha(v_\alpha)} \propto \exp\Big(\langle \log p(X,S,Z|\Theta)\rangle_{q^{(n+1)}/q_{Z_\alpha(v_\alpha)}}\Big) \tag{5.17}$$

where here (n) in the superscript is for iteration index. Now, we formulate the approximating distribution $q(S)$ and after $q(Z)$. When we expand the log and drop $\log P(Z|\Theta)$ and all other irrelevant S terms $q_{S(v_0,*)}$, disregarding (n), we end up with

$$\begin{aligned}
q_{S(v_0,*)} &\propto \exp\Big\{\langle \log p(X|S) + \log p(S|Z)\rangle_{q/q_{S(v_0,*)}}\Big\} \\
&\propto \exp\Big\{\sum_{\bar{v}_0}\Big(S(v)\langle\log\prod_{\alpha} Z_\alpha(v_\alpha)\rangle - \log\Gamma(S(v)+1)\Big) \\
&\qquad + \log\delta\left(X(v_0) - \sum_{\bar{v}_0} S(v)\right)\Big\} \\
&\propto \exp\left\{\sum_{\bar{v}_0}\left(S(v)\sum_{\alpha}\log\langle Z_\alpha(v_\alpha)\rangle - \log\Gamma(S(v)+1)\right)\right\}\delta\left(X(v_0) - \sum_{\bar{v}_0} S(v)\right)
\end{aligned}$$

Exactly, the slice $S(v_0,*)$ is sampled from the multinomial distribution as $X(v_0)$ is the total number of observations. As the joint posterior density of a vector s of *a priori* independent Poisson random variables s_i with intensity vector λ conditioned on the sum $x = \sum_i s_i$ is multinomial distributed with cell probabilities $p = \lambda/\sum_i \lambda_i$, denoted

by $\mathcal{M}(s; x, p)$. Finally we obtain the approximating distributions as

$$q_{S(v_0,*)} \sim \mathcal{M}(S(v_0,*), X(v_0), P(v_0,*)) \tag{5.18}$$

Then, the cell probabilities and sufficient statistics for $q_{S(v_0,*)}$ are

$$P(v) = \frac{\exp(\sum_\alpha \langle \log Z_\alpha(v_\alpha) \rangle)}{\sum_{\bar{v}_0} \exp(\sum_\alpha \langle \log Z_\alpha(v_\alpha) \rangle)} \tag{5.19}$$

$$\langle S(v) \rangle = X(v_0) P(v) \tag{5.20}$$

Interestingly the cell probabilities $P(v)$ can be further transformed into compact form, note that

$$P(v) = \frac{\exp(\sum_\alpha \langle \log Z_\alpha(v_\alpha) \rangle)}{\sum_{\bar{v}_0} \exp(\sum_\alpha \langle \log Z_\alpha(v_\alpha) \rangle)} \tag{5.21}$$

$$= \frac{\prod_\alpha \exp(\langle \log Z_\alpha(v_\alpha) \rangle)}{\sum_{\bar{v}_0} \prod_\alpha \exp(\langle \log Z_\alpha(v_\alpha) \rangle)} \tag{5.22}$$

$$= \frac{\prod_\alpha L_\alpha(v_\alpha)}{\sum_{\bar{v}_0} \prod_\alpha L_\alpha(v_\alpha)} \tag{5.23}$$

$$= \frac{\prod_\alpha L_\alpha(v_\alpha)}{\hat{X}_L(v_0)} \tag{5.24}$$

where we had specified $L_\alpha(v_\alpha)$ and $\hat{X}_L(v_0)$ in Equation 5.8 and Equation 5.9. Then the sufficient statistics $\langle S(v) \rangle$ turns to

$$S(v)\rangle = X(v_0) P(v) = \frac{X(v_0)}{\hat{X}_L(v_0)} \prod_\alpha L_\alpha(v_\alpha) \tag{5.25}$$

It is interesting to compare this sufficient statistics equation $\langle S(v) \rangle$ with the one we found in Chapter 4 in EM based Equation 4.12 which is $\langle S(v) \rangle = \frac{X(v_0)}{\hat{X}(v_0)} \prod_\alpha Z_\alpha(v_\alpha)$. As expected the only difference is that $Z_\alpha(v_\alpha)$ is replaced with $L_\alpha(v_\alpha)$ that also affects $\hat{X}$ to be replaced with $\hat{X}_L(v_0)$.

Now we turn to formulating $q(Z)$. The distribution $q_{Z_\alpha(v_\alpha)}$ is obtained similarly as after we expand the log and drop irrelevant terms it becomes proportional to (dis-

regarding the iteration index)

$$q_{Z_\alpha(v_\alpha)} \propto \exp\left(\langle \log p(S|Z) + \log p(Z|\Theta)\rangle_{q/q_{Z_\alpha(v_\alpha)}}\right) \tag{5.26}$$

$$\propto \log Z_\alpha(v_\alpha)\left(A_\alpha(v_\alpha) - 1 + \sum_{\bar{v}_\alpha}\langle S(v)\rangle\right) \tag{5.27}$$

$$- Z_\alpha(v_\alpha)\left(\frac{A_\alpha(v_\alpha)}{B_\alpha(v_\alpha)} + \sum_{\bar{v}_\alpha}\prod_{\alpha'\neq\alpha}\langle Z_{\alpha'}(v_{\alpha'})\rangle\right) \tag{5.28}$$

which is the distribution

$$q_{Z_\alpha(v_\alpha)} \sim \mathcal{G}(C_\alpha(v_\alpha), D_\alpha(v_\alpha)) \tag{5.29}$$

where the shape and scale parameters for $q_{Z_\alpha(v_\alpha)}$ are

$$C_\alpha(v_\alpha) = A_\alpha(v_\alpha) + \sum_{\bar{v}_\alpha}\langle S(v)\rangle \tag{5.30}$$

$$D_\alpha(v_\alpha) = \left(\frac{A_\alpha(v_\alpha)}{B_\alpha(v_\alpha)} + \sum_{\bar{v}_\alpha}\prod_{\alpha'\neq\alpha}\langle Z_{\alpha'}(v_{\alpha'})\rangle\right)^{-1} \tag{5.31}$$

Note that as expected the shape and scale parameters of the gamma distribution above, i.e. $C_\alpha(v_\alpha)$ and $D_\alpha(v_\alpha)$ have the same shape as those of the posterior distribution with n observations in Chapter 2 in Equation 2.98. Finally, sufficient statistics are obtained by definition of the gamma distribution as

$$E_\alpha(v_\alpha) = \langle Z_\alpha(V_\alpha)\rangle = C_\alpha(v_\alpha)D_\alpha(v_\alpha) \tag{5.32}$$

$$L_\alpha(v_\alpha) = \exp(\langle \log Z_\alpha(v_\alpha)\rangle) = \exp(\Psi(C_\alpha(v_\alpha)))D_\alpha(v_\alpha) \tag{5.33}$$

□

5.3.1. Tensor Forms via Δ Function

Similar to Chapter 4, we make use of Δ function to make the notation shorter and implementation friendly. However, before that we need to redefine and extend the definition of Δ function such that the tensor Z_α to be a variable instead of a constant, i.e. it is to be parameter of the Δ function since now we have quantities, besides Z_α, also based on $\langle Z_\alpha \rangle$ and even on $\exp \langle \log Z_\alpha \rangle$.

Definition 5.1. *A tensor valued $\Delta_\alpha^Z(Q) : \mathbb{R}^{|Q|} \to \mathbb{R}^{|Z_\alpha|}$ function associated with component Z_α is defined as*

$$\Delta_\alpha^Z(Q) \equiv \left[\sum_{\bar{v}_\alpha} \left(Q(v_0) \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'}) \right) \right] \tag{5.34}$$

where as usual $\circ$ and / stand for element wise multiplication (Hadamard product) and division respectively. Recall that $\Delta_\alpha^Z(Q)$ is an object the same size of Z_α while $\Delta_\alpha^Z(Q)(v_\alpha)$ refers to a particular element of $\Delta_\alpha^Z(Q)$. Here Z is clearly a parameter of the Δ function where we prefer writing it as superscript as $\Delta_\alpha^Z(Q)$ but it is to be considered to equal to $\Delta_\alpha(Z, Q)$.

Before giving tensor forms we want to identify important terms in the formulation of the shape and the scale parameters $C_\alpha(v_\alpha)$ and $D_\alpha(v_\alpha)$. For $C_\alpha(v_\alpha)$ we need to find out $\sum_{\bar{v}_\alpha} \langle S(v) \rangle$, which can be written into a form that the Δ functions can be used

$$\begin{aligned} \sum_{\bar{v}_\alpha} \langle S(v) \rangle &= \sum_{\bar{v}_\alpha} X(v_0) P(v) = \sum_{\bar{v}_\alpha} \frac{X(v_0)}{\hat{X}_L(v_0)} \prod_{\alpha} L_\alpha(v_\alpha) \\ &= L_\alpha(v_\alpha) \sum_{\bar{v}_\alpha} \frac{X(v_0)}{\hat{X}_L(v_0)} \prod_{\alpha' \neq \alpha} L_{\alpha'}(v_{\alpha'}) \end{aligned} \tag{5.35}$$

On the other hand, for computation of the gamma scale parameter $D_\alpha(v_\alpha)$, we note that

$$\sum_{\bar{v}_\alpha} \prod_{\alpha' \neq \alpha} \langle Z_{\alpha'}(v_{\alpha'}) \rangle = \sum_{\bar{v}_\alpha} \prod_{\alpha' \neq \alpha} E_{\alpha'}(v_{\alpha'}) = \Delta_\alpha^E(M)$$

Then, by use of $\Delta_\alpha^E(.)$ and $\Delta_\alpha^L(.)$ the update for hyperparameters C_α and D_α becomes

$$C_\alpha = A_\alpha + L_\alpha \circ \Delta_\alpha^L(M \circ X/\hat{X}_L) \tag{5.36}$$

$$D_\alpha = \left(\frac{A_\alpha}{B_\alpha} + \Delta_\alpha^E(M)\right)^{-1} \tag{5.37}$$

that, in turn, since $\langle Z_\alpha \rangle$ is $C_\alpha \circ D_\alpha$, E_α and L_α the sufficient statistics for $q(Z_\alpha)$ become

$$\langle Z_\alpha \rangle = E_\alpha \leftarrow \frac{A_\alpha + L_\alpha \circ \Delta_\alpha^L(M \circ X/\hat{X}_L)}{\frac{A_\alpha}{B_\alpha} + \Delta_\alpha^E(M)} \tag{5.38}$$

$$\exp\langle \log(Z_\alpha) \rangle = L_\alpha \leftarrow \exp(\psi(C_\alpha)) \circ D_\alpha \tag{5.39}$$

5.3.2. Handling Missing Data

Missing data is handled smoothly by the following observation model [29, 30]

$$p(X|S)p(S|Z_{1:N}) = \prod_{v_0}\prod_{\bar{v}_0}\left\{p(X(v_0)|S(v))\; p(S(v)|Z_{1:N})\right\}^{M(v_0)} \tag{5.40}$$

where mask tensor M is formed as

$$M(v_0) = \begin{cases} 0 & X(v_0) \text{ is missing} \\ 1 & \text{otherwise} \end{cases}$$

Then, slight modifications are needed to be done in VB based update equation. We start with the modification on the full joint. Since the priors are not part of the observation model they are not affected, the first two terms of $\langle \log p(X, S, Z|\Theta) \rangle_{q(S,Z)}$ become

$$\sum_{v_0} M(v_0) \left\langle \log \delta \left(X(v_0) - \sum_{\bar{v}_0} S(v) \right) \right\rangle \tag{5.41}$$

$$+ \sum_{v} M(v_0) \left(\langle S(v) \rangle \left\langle \log \prod_\alpha Z_\alpha(v_\alpha) \right\rangle - \prod_\alpha \langle Z_\alpha(v_\alpha) \rangle - \langle \log \Gamma(S(v)+1) \rangle \right) \ldots$$

and this results

$$q_{Z_\alpha(v_\alpha)} \propto \log \langle Z_\alpha(v_\alpha)\rangle \left(A_\alpha(v_\alpha) - 1 + \sum_{\bar{v}_\alpha} M(v_0)\langle S(v)\rangle \right) \tag{5.42}$$

$$- \langle Z_\alpha(v_\alpha)\rangle \left(\frac{A_\alpha(v_\alpha)}{B_\alpha(v_\alpha)} + \sum_{\bar{v}_\alpha} M(v_0) \prod_{\alpha' \neq \alpha} \langle Z_{\alpha'}(v_{\alpha'})\rangle \right) \tag{5.43}$$

$$\propto \mathcal{G}(C_\alpha(v_\alpha), D_\alpha(v_\alpha)) \tag{5.44}$$

This modifies the gamma parameters for $q(Z)$ in Equation 5.30 and in Equation 5.31 as to include the mask $M(v_0)$ as

$$C_\alpha(v_\alpha) = A_\alpha(v_\alpha) + \sum_{\bar{v}_\alpha} M(v_0)\langle S(v)\rangle \tag{5.45}$$

$$D_\alpha(v_\alpha) = \left(\frac{A_\alpha(v_\alpha)}{B_\alpha(v_\alpha)} + \sum_{\bar{v}_\alpha} M(v_0) \prod_{\alpha' \neq \alpha} \langle Z_{\alpha'}(v_{\alpha'})\rangle \right)^{-1} \tag{5.46}$$

The other terms are not affected since mask matrix is already in the definition of $C_\alpha(v_\alpha)$ and $D_\alpha(v_\alpha)$. Moreover, $A_\alpha(v_\alpha)$ and $B_\alpha(v_\alpha)$ are priors and not part of the observation model. For the bound equations in next section, on the other hand, the only modification is that X is replaced by $M \circ X$.

5.4. Likelihood Bound via Variational Bayes

Recall that in Equation 5.13 we lower bound the marginal log likelihood $\mathcal{L}(\Theta) = \log p(X|\Theta)$ for any arbitrary $PLTF_{KL}$ model based on variational Bayes

$$\begin{aligned}
\mathcal{L}(\Theta) \geq \mathcal{B} &= \langle \log p(X, S, Z|\Theta)\rangle_{q(S,Z)} + H[q(S, Z)] \\
&= \langle \log p(X|S, Z, \Theta) + \log p(S|Z, \Theta) + \log p(Z|\Theta)\rangle_{q(S,Z)} + H[q(S)] + H[q(Z)] \\
&= \langle \log p(X|S) + \log p(S|Z) + \log p(Z|\Theta)\rangle_{q(S,Z)} + H[q(S)] + H[q(Z)]
\end{aligned} \tag{5.47}$$

where the simplifications in the joint distribution are done in accordance with the PLTF graphical model by reading the conditional independence relations, and also dividing

the entropy into two terms is due to that the distribution $q(S, Z)$ is approximated and is factorized as $q(S, Z) = q(S)q(Z)$.

At this point due to the large number of terms let us write the bound terms as sum of likelihood and entropy parts and compute them separately, then add them all

$$\mathcal{B} = \mathcal{B}_L + \mathcal{B}_H \tag{5.48}$$

$$\mathcal{B}_L = \langle \log p(X|S) + \log p(S|Z) + \log p(Z|\Theta) \rangle_{q(S,Z)} \tag{5.49}$$

$$\mathcal{B}_H = H[q(S)] + H[q(Z)] \tag{5.50}$$

To find $\mathcal{B}_L$, we find out each expectation then add them up. Due to the fact that $p(X|S)$ is actually deterministic, while $S|Z \sim \mathcal{PO}$ and $Z|\Theta \sim \mathcal{G}$ we end up with the following expression for $\mathcal{B}_L$

$$\begin{aligned}
\mathcal{B}_L = &\sum_{v_0} \left\langle \log \delta \left(X(v_0) - \sum_{\bar{v}_0} S(v) \right) \right\rangle \\
&+ \sum_{v} \langle S(v) \rangle \left\langle \log \prod_{\alpha} Z_\alpha(v_\alpha) \right\rangle - \prod_{\alpha} \langle Z_\alpha(v_\alpha) \rangle - \langle \log \Gamma(S(v)+1) \rangle \\
&+ \sum_{\alpha} \sum_{v_\alpha} (A_\alpha(v_\alpha) - 1) \langle \log Z_\alpha(v_\alpha) \rangle - \frac{A_\alpha(v_\alpha)}{B_\alpha(v_\alpha)} \langle Z_\alpha(v_\alpha) \rangle \\
&\qquad - \log \Gamma(A_\alpha(v_\alpha)) - A_\alpha(v_\alpha) \log \frac{B_\alpha(v_\alpha)}{A_\alpha(v_\alpha)}
\end{aligned}$$

$\mathcal{B}_H$, on the other hand, is, by definition of (Shannon) entropy

$$\begin{aligned}
\mathcal{B}_H &= H[q(S)] + H[q(Z)] = \langle \log q(S) \rangle + \langle \log q(Z) \rangle \\
&= \sum_{v_0} \Big\{ - \log \Gamma(X(v_0)+1) - \sum_{\bar{v}_0} \langle S(v) \rangle \log P(v) + \sum_{\bar{v}_0} \langle \log \Gamma(S(v)+1) \rangle \\
&\qquad - \left\langle \log \delta \Big(X(v_0) - \sum_{\bar{v}_0} S(v) \Big) \right\rangle \Big\} \\
&+ \sum_{\alpha} \sum_{v_\alpha} \Big\{ - (C_\alpha(v_\alpha) - 1) \Psi(C_\alpha(v_\alpha)) + \log(D_\alpha(v_\alpha))) + C_\alpha(v_\alpha) + \log \Gamma(C_\alpha(v_\alpha)) \Big\}
\end{aligned}$$

where the first two terms comes from the entropy of $q(S(v_0, *)) \sim \mathcal{M}$ and the last one

comes from the entropy of $q(Z_\alpha(v_\alpha)) \sim \mathcal{G}(C_\alpha(v_\alpha), D_\alpha(v_\alpha))$. After cancellation of the common terms coming from entropy and free energy terms the bound becomes

$$\mathcal{B} = \sum_v \langle S(v)\rangle \left\langle \log \prod_\alpha Z_\alpha(v_\alpha) \right\rangle - \prod_\alpha \langle Z_\alpha(v_\alpha)\rangle \tag{5.51}$$

$$+ \sum_\alpha \sum_{v_\alpha} (A_\alpha(v_\alpha) - 1)\langle \log Z_\alpha(v_\alpha)\rangle - \frac{A_\alpha(v_\alpha)}{B_\alpha(v_\alpha)} \langle Z_\alpha(v_\alpha)\rangle \tag{5.52}$$

$$- \log \Gamma(A_\alpha(v_\alpha)) - A_\alpha(v_\alpha) \log \frac{B_\alpha(v_\alpha)}{A_\alpha(v_\alpha)}$$

$$+ \sum_{v_0} \Big\{ - \log \Gamma(X(v_0) + 1) - \sum_{\bar{v}_0} \langle S(v)\rangle \log P(v) \Big\} \tag{5.53}$$

$$+ \sum_\alpha \sum_{v_\alpha} -(C_\alpha(v_\alpha) - 1)\Psi(C_\alpha(v_\alpha)) + \log(D_\alpha(v_\alpha))) + C_\alpha(v_\alpha) + \log \Gamma(C_\alpha(v_\alpha))$$

Here we apply the equivalence $\sum_v \equiv \sum_{v_0} \sum_{\bar{v}_0}$ to Equation 5.51 which results to

$$\sum_v \prod_\alpha \langle Z_\alpha(v_\alpha)\rangle = \sum_{v_0} \sum_{\bar{v}_0} \prod_\alpha \langle Z_\alpha(v_\alpha)\rangle = \sum_{v_0} -\hat{X}_E(v_0) \tag{5.54}$$

where

$$\hat{X}_E(v_0) = \sum_{\bar{v}_0} \prod_\alpha E_\alpha(v_\alpha) \tag{5.55}$$

Next, we further simplify the bound by combining $\langle S(v)\rangle$ related terms, i.e. by combining Equation 5.51 and Equation 5.53 into $\mathcal{B}_{part}$ as

$$\begin{aligned} \mathcal{B}_{part} &= \sum_v \langle S(v)\rangle \left\langle \log \prod_\alpha Z_\alpha(v_\alpha) \right\rangle - \sum_{v_0} \sum_{\bar{v}_0} \langle S(v)\rangle \log P(v) \\ &= \sum_{v_0} X(v_0) \log \hat{X}_L(v_0) \end{aligned} \tag{5.56}$$

Then, the bound $\mathcal{B}$ becomes

$$\mathcal{B} = \sum_{v_0} -\hat{X}_E(v_0) - \log\Gamma(X(v_0)+1) + X(v_0)\log\hat{X}_L(v_0) \tag{5.57}$$

$$+ \sum_{\alpha}\sum_{v_\alpha}(A_\alpha(v_\alpha)-1)\langle\log Z_\alpha(v_\alpha)\rangle - \frac{A_\alpha(v_\alpha)}{B_\alpha(v_\alpha)}\langle Z_\alpha(v_\alpha)\rangle \tag{5.58}$$

$$- \log\Gamma(A_\alpha(v_\alpha)) - A_\alpha(v_\alpha)\log\frac{B_\alpha(v_\alpha)}{A_\alpha(v_\alpha)}$$

$$+ \sum_{\alpha}\sum_{v_\alpha} -(C_\alpha(v_\alpha)-1)\Psi(C_\alpha(v_\alpha)) + \log(D_\alpha(v_\alpha)) + C_\alpha(v_\alpha) + \log\Gamma(C_\alpha(v_\alpha))$$

On the other hand, recall that expectation of the gamma distribution is $\langle Z_\alpha(v_\alpha)\rangle = C_\alpha(v_\alpha)D_\alpha(v_\alpha)$, while $\langle\log Z_\alpha(v_\alpha)\rangle$ is $\psi(C_\alpha(v_\alpha)) + \log D_\alpha(v_\alpha)$ (for the gamma distribution convention we choose). Next, when we plug these three quantities in the bound expression we come up with new likelihood bound as

$$\mathcal{B} = \sum_{v_0} -\hat{X}_E(v_0) - \log\Gamma(X(v_0)+1) + X(v_0)\log\hat{X}_L(v_0) \tag{5.59}$$

$$+ \sum_{\alpha}\sum_{v_\alpha}(A_\alpha(v_\alpha)-1)\Big(\psi(C_\alpha(v_\alpha)) + \log D_\alpha(v_\alpha)\Big) - \frac{A_\alpha(v_\alpha)}{B_\alpha(v_\alpha)}\Big(C_\alpha(v_\alpha)D_\alpha(v_\alpha)\Big) \tag{5.60}$$

$$- \log\Gamma(A_\alpha(v_\alpha)) - A_\alpha(v_\alpha)\log\frac{B_\alpha(v_\alpha)}{A_\alpha(v_\alpha)}$$

$$+ \sum_{\alpha}\sum_{v_\alpha} -(C_\alpha(v_\alpha)-1)\Psi(C_\alpha(v_\alpha)) + \log D_\alpha(v_\alpha) + C_\alpha(v_\alpha) + \log\Gamma(C_\alpha(v_\alpha)) \tag{5.61}$$

This new bound is actual a well-known quantity that can be written compactly as

$$\mathcal{B} = \Big\{\sum_{v_0} -\hat{X}_E(v_0) - \log\Gamma(X(v_0)+1) + X(v_0)\log\hat{X}_L(v_0)\Big\} \tag{5.62}$$

$$- \sum_{\alpha}\sum_{v_\alpha} KL[q(Z)||p(Z|\Theta)]$$

where $q(Z)$ is the posterior as $q(Z) = \mathcal{G}(Z; C_\alpha(v_\alpha), D_\alpha(v_\alpha))$ and $p(Z|\Theta)$ is the prior as $p(Z) = \mathcal{G}(Z; A_\alpha(v_\alpha), B_\alpha(v_\alpha)/A_\alpha(v_\alpha))$.

Here we want to prove Equation 5.56 given as $\mathcal{B}_{part}$.

Proof. We first start with

$$\mathcal{B}_{part} = \sum_{v} \langle S(v) \rangle \left\langle \log \prod_{\alpha} Z_{\alpha}(v_{\alpha}) \right\rangle - \sum_{v_0} \sum_{\bar{v}_0} \langle S(v) \rangle \log P(v) \tag{5.63}$$

then by using the expected sufficient statistics

$$P(v) = \frac{\prod_{\alpha} L_{\alpha}(v_{\alpha})}{\hat{X}_L(v_0)} \qquad \text{and} \qquad \langle S(v) \rangle = \frac{X(v_0)}{\hat{X}_L(v_0)} \prod_{\alpha} L_{\alpha}(v_{\alpha}) \tag{5.64}$$

and the simplification

$$\sum_{v} \langle S(v) \rangle \left\langle \log \prod_{\alpha} Z_{\alpha}(v_{\alpha}) \right\rangle = \sum_{v} \langle S(v) \rangle \left\langle \sum_{\alpha} \log Z_{\alpha}(v_{\alpha}) \right\rangle \tag{5.65}$$

$$= \sum_{v} \langle S(v) \rangle \sum_{\alpha} \langle \log Z_{\alpha}(v_{\alpha}) \rangle \tag{5.66}$$

and also recall the equivalence of $\sum_v \equiv \sum_{v_0} \sum_{\bar{v}_0}$, $\mathcal{B}_{part}$ becomes

$$\mathcal{B}_{part} = \sum_{v} \langle S(v) \rangle \sum_{\alpha} \langle \log Z_{\alpha}(v_{\alpha}) \rangle - \sum_{v_0} \sum_{\bar{v}_0} \langle S(v) \rangle \log P(v) \tag{5.67}$$

$$= \sum_{v} \langle S(v) \rangle \Big\{ \sum_{\alpha} \langle \log Z_{\alpha}(v_{\alpha}) \rangle - \log \prod_{\alpha} L_{\alpha}(v_{\alpha}) + \log \hat{X}_L(v_0) \Big) \Big\} \tag{5.68}$$

Finally we identify another important equivalence

$$\langle \log Z_{\alpha}(v_{\alpha}) \rangle = \log \exp \langle \log Z_{\alpha}(v_{\alpha}) \rangle = \log L_{\alpha}(v_{\alpha}) \tag{5.69}$$

that simplifies $\mathcal{B}_{part}$ to as simple as $\sum_{v_0} X(v_0) \log \hat{X}_L(v_0)$

$$\mathcal{B}_{part} = \sum_{v} \langle S(v) \rangle \Big\{ \sum_{\alpha} \log L_\alpha(v_\alpha) - \sum_{\alpha} \log L_\alpha(v_\alpha) + \log \hat{X}_L(v_0) \Big) \Big\} \tag{5.70}$$

$$= \sum_{v} \frac{X(v_0)}{\hat{X}_L(v_0)} \prod_{\alpha} L_\alpha(v_\alpha) \Big\{ \log \hat{X}_L(v_0) \Big) \Big\} \tag{5.71}$$

$$= \sum_{v_0} \frac{X(v_0)}{\hat{X}_L(v_0)} \Big\{ \log \hat{X}_L(v_0) \Big) \Big\} \sum_{\bar{v}_0} \prod_{\alpha} L_\alpha(v_\alpha) \tag{5.72}$$

$$= \sum_{v_0} X(v_0) \log \hat{X}_L(v_0) \tag{5.73}$$

where we identify the term $\sum_{\bar{v}_0} \prod_{\alpha} L_\alpha(v_\alpha)$ as $\hat{X}_L(v_0)$ also note that $\hat{X}_L(v_0)$ is independent of the index $\hat{v}_0$ hence take it out of the sum $\sum_{\hat{v}_0}$. □

5.5. Asymptotic Analysis of Model Selection

Recall that we computed the bound $\mathcal{B}$ as

$$\mathcal{B} = \Big\{ \sum_{v_0} -\hat{X}_E(v_0) - \log \Gamma(X(v_0) + 1) + X(v_0) \log \hat{X}_L(v_0) \Big\} \tag{5.74}$$

$$- \sum_{\alpha} \sum_{v_\alpha} KL[q(Z)||p(Z|\Theta)] \tag{5.75}$$

When we have large amount of data $\hat{X}_L(v_0)$ becomes $\hat{X}_E(v_0)$ since

$$\hat{X}_L(v_0) = \sum_{\bar{v}_0} \prod_{\alpha} L_\alpha(v_\alpha) = \sum_{\bar{v}_0} \prod_{\alpha} \exp\big(\langle \log Z_\alpha(v_\alpha) \rangle\big) \tag{5.76}$$

$$= \sum_{\bar{v}_0} \prod_{\alpha} \exp\Big(\psi\big(C_\alpha(v_\alpha)\big)\Big) D_\alpha(v_\alpha) \tag{5.77}$$

and since asymptotically $\psi(C_\alpha(v_\alpha)) \simeq \log(C_\alpha(v_\alpha))$ we end up with

$$\sum_{\bar{v}_0} \prod_{\alpha} C_\alpha(v_\alpha) D_\alpha(v_\alpha) = \sum_{\bar{v}_0} \prod_{\alpha} \langle Z_\alpha(v_\alpha) \rangle = \hat{X}_E(v_0) \tag{5.78}$$

In addition, note that under large sample, KL divergence can also be approximately calculated. Here as we already know that KL divergence can be decomposed as

$$KL[q(Z)||p(Z|\Theta)] = -H[q(Z)] + \langle \log p(Z|\Theta) \rangle_{q(Z)} \tag{5.79}$$

where in this formulation $q(Z)$ is the posterior while $p(Z|\Theta)$ is the prior. In Chapter 2 we already formulated that asymptotically, i.e. when number of observations increases largely the cross entropy terms $\langle \log p(Z|\Theta) \rangle_{q(Z)}$ become constant and does not scale with this increase, and hence can be neglected. On the other hand, the entropy of the posterior distribution under large sample turns into the entropy of the Gaussian which we have already calculated in Chapter 2. Then the bound is

$$\mathcal{B} = \log \mathcal{PO}(X; \hat{X}_E) + \sum_{\alpha}^{|\alpha|} H[\mathcal{G}(Z_\alpha; C_\alpha, D_\alpha)] \tag{5.80}$$

We already computed the entropy of the gamma distribution under large sample in Equation 2.163, and finally the asymptotic bound in tensor form is

$$\mathcal{B} = \log \mathcal{PO}(X; \hat{X}_E) + \sum_{\alpha}^{|\alpha|} \frac{1}{2} \log \left(C_\alpha \circ D_\alpha^2 \right) \tag{5.81}$$

5.5.1. Generalization of Asymptotic Variational Lower Bound

The treatment in the previous chapter, i.e. the asymptotic log likelihood bound for $PLTF_{KL}$ models can be generalized for other PLTF models as well. Here we start with a variational lower bound equation for the model likelihood.

Theorem 5.2. *Let X be the observation, Θ be the hyperparameter and Z_{MAP} be the MAP estimate of the parameter Z. Then, the log-likelihood bound for the observation X is given by the following inequality*

$$\mathcal{L}(X|\Theta) \geq \mathcal{B} = \log p(X|Z_{MAP}, \Theta) + H[q(Z)] \tag{5.82}$$

where $q(Z)$ is the asymptotic posterior and $H[.]$ is the entropy.

Proof. Note that the bound for the log marginal likelihood given in Equation 5.13 can be written as [75]

$$\mathcal{L}(X|\Theta) \geq \left\langle \log \frac{p(X,S|Z,\Theta)}{q(Z)} \right\rangle_{q(S,Z)} - KL(q(Z)||p(Z|\Theta)) \tag{5.83}$$

Here the bound attains its maximum for $q(S,Z) = p(S,Z|X,\Theta)$ where it becomes the exact log marginal likelihood. On the other hand, q is a factorized distribution such that $q(S,Z) = q(S)q(Z)$ while the posterior over the parameters S and Z is factorized as $p(S|X,Z,\Theta)p(Z|X,\Theta)$. Since we are free to choose the distribution $q(S)$ and $q(Z)$, the assignment that $q(S) = p(S|X,Z,\Theta)$ and $q(Z) = p(Z|X,\Theta)$ yields the maximum lower bound. Hence plugging $q(S) = p(S|X,Z,\Theta)$ in the first part, it becomes as

$$\mathcal{L}(X|\Theta) \geq \langle \log p(X|Z,\Theta) \rangle_{q(Z)} - KL(q(Z)||p(Z|\Theta)) \tag{5.84}$$

This result is general and holds without requiring the asymptotic assumption. Then by assuming asymptotic normality we further simplify the KL term as entropy of the asymptotic posterior $H[q(Z)]$ and we arrive at the equation Equation 5.82. □

5.5.2. Variational Bound and Bayesian Information Criterion (BIC)

It is shown in [75] that the criterion BIC can be recovered form VB in the limit of large data. Here we obtain the BIC equivalent of the bound. We already showed that under large sample case, entropy of the posterior distribution is approximated as

$$H[q(Z)] \simeq \frac{1}{2} \log \frac{1}{N} + const \tag{5.85}$$

Hence the likelihood bound for general PLTF models turns to the following ignoring the constant

$$\mathcal{B} \simeq \log p(X|Z_{MAP}, \Theta) - \sum_{\alpha}^{|\alpha|} \frac{1}{2} M_\alpha \log N_\alpha \tag{5.86}$$

where number of parameters and number of dimensions are identified as

$$N_\alpha = |\bar{v}_\alpha| \qquad M_\alpha = |v_\alpha| \tag{5.87}$$

Note that for KL the posterior parameters are already identified as follows where N_α is identified as cardinality of $\bar{v}_\alpha$

$$C_\alpha(v_\alpha) = A_\alpha(v_\alpha) + \sum_{\bar{v}_\alpha} \langle S(v) \rangle \tag{5.88}$$

Finally, BIC can be written as $PLTF_{KL}$ models as

$$\mathcal{B} \simeq \log \mathcal{PO}(X; \hat{X}_E) - \sum_{\alpha}^{|\alpha|} \frac{1}{2} M_\alpha \log N_\alpha \tag{5.89}$$

The following example illustrates identification of the terms M_α and N_α.

Example 5.1. *For the CP model* $\hat{X}^{i,j,k} = \sum_r A^{i,r} B^{j,r} C^{k,r}$ *the number of parameters and the dimensionality's are as follows*

$$N_A = |j||k| \qquad N_B = |i||k| \qquad N_C = |i||j| \tag{5.90}$$

$$M_A = |i||r| \qquad M_B = |j||r| \qquad M_C = |k||r| \tag{5.91}$$

5.6. Experiments

To experiment the Bayesian model selection procedure for the tensor factorization we set up two sets of experiments. In the first set we test model order and structure

determinations, whereas in the second set we compare VB and BIC techniques and also test missing data case on the image dataset.

5.6.1. Experiment 1

In this experiment we generate synthetic data sets and test our findings on model order and structure determination. For the experiments in this section we use the algorithm that implements variational fixed point update equation and variational bound computation for the models.

The first experiment is about model order determination where the goal is to determine the cardinality of the latent indices p, q, r of the TUCKER3 model $X^{i,j,k} = \sum_{pqr} G^{p,q,r} A^{i,p} B^{j,q} C^{k,r}$. Here we denote the cardinality of an index i as $|i|$. For simplicity, we set $|r|$ to its true value, while we seek for the values $|p|$ and $|q|$ simultaneously. Each of $|p|$ and $|q|$ is set to be from 2 to 10 (ignoring 1) incremented by one gradually at each run. The score computation is repeated 10 times for the same observation data with different random initializations, and we pick up the highest bound score. As for the experiment settings, the iteration number is 2000, shape and scale parameters of the gamma priors are set to be 1. For the generated data set, the cardinality of the observed indices i, j, k are set to be $40, 40, 40$ (i.e. the size of the data) while the cardinality of the latent indices p, q, r are set to be $7, 4, 3$ as true model order. During this experiment we use the exact bound score. At the end, we obtain a matrix where each cell is a bound score of the TUCKER3 model with model orders $|p| = 2 \dots 10, |q| = 2 \dots 10$. As a result of this experiment, we observe the highest bound (lightest area) at around true model order of $|p| = 7, |q| = 4$ on Figure 5.2.

The second experiment is on model structure learning. Here our goal is to predict the underlying factorization structure for a given data set generated by either CP or TUCKER3 models. The true model order is not known in advance either. For a given data set we compute the bound for each of CP and TUCKER3 with all possible model orders. Then the model yielding the maximum is selected accordingly. For the experiment settings, the iteration number is 2000, shape and scale parameters of

Input: X (observation), M (mask array), A and B (priors), and $N = |\alpha|$

Output: E (expected value of factors), and $\mathcal{B}$ (bound)

for $\alpha = 1 \ldots N$ **do**

$L_\alpha \sim \mathcal{G}(A_\alpha, B_\alpha/A_\alpha)$

$E_\alpha \sim \mathcal{G}(A_\alpha, B_\alpha/A_\alpha)$

end for

Main loop

for epoch $= 1 \ldots$ MAXITER **do**

Compute $\hat{X}_L$ and $\hat{X}_E$

$\hat{X}_L = L_1(L_3 \odot L_2)^T$ *for CP*

$\hat{X}_L = L_1 L_{4_{(1)}}(L_3 \otimes L_2)^T$ *for TUCKER3*

Computation for $\hat{X}_E$ is similar and is omitted.

for $\alpha = 1 \ldots N$ **do**

$C_\alpha = A_\alpha + L_\alpha \circ \Delta_\alpha^L(M \circ X/\hat{X}_L)$

$D_\alpha = 1/((A_\alpha/B_\alpha) + \Delta_\alpha^E(M))$

$E_\alpha = C_\alpha \circ D_\alpha$

end for

for $\alpha = 1 \ldots N$ **do**

$L_\alpha = \exp(\psi(C_\alpha)) \circ D_\alpha$

end for

end for

1. Compute the bound (exact)

$\mathcal{B} = \left(-\hat{X}_E - \log\Gamma(X+1) + X \circ \log \hat{X}_L\right) - \sum_\alpha^N KL(\mathcal{G}(C_\alpha, D_\alpha)||\mathcal{G}(A_\alpha, B_\alpha))$

2. Compute the bound (approximated)

$\mathcal{B} = \left(-\hat{X}_E - \log\Gamma(X+1) + X \circ \log \hat{X}_E\right) - \sum_\alpha^N \frac{1}{2}\log(C_\alpha \circ D_\alpha^2)$

3. Compute the bound (approximated)

$\mathcal{B} = \left(-\hat{X}_E - \log\Gamma(X+1) + X \circ \log \hat{X}_E\right) - \sum_\alpha^N \frac{1}{2} M_\alpha \log N_\alpha$

Figure 5.1. Matlab-like implementation for the VB based model selection. For the Matlab $\log\Gamma$ is the function *gammaln*(), $\psi()$ is *psi*(). The unfolding operation and Khati-Rao product are implemented in the N-way Toolbox [73] as *nshape*() and *krb*().

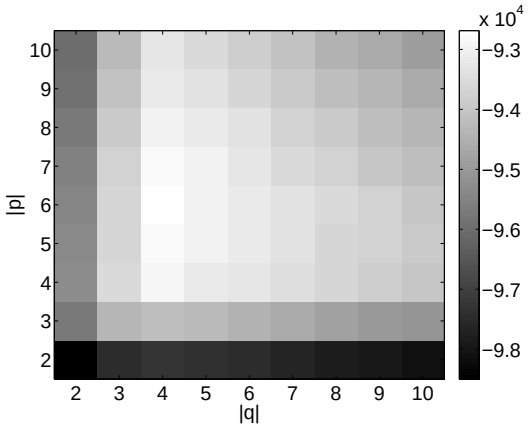


Figure 5.2. The figure illustrates model order determination for TUCKER3 model. The cardinality of the latent indices p and q are given in the Y and X dimensions respectively while the cardinality of r is taken to be constant and is not shown. The Z dimension is the variational bound yielded by the model. The true model order is as $|p| = 7$ and $|q| = 4$.

the gamma priors are set to be 1. The maximum cardinality of the latent indices, i.e. r for CP model and p, q for TUCKER3 is 10. For TUCKER3, $|r|$ is assumed to be known for simplicity. During this experiment we use the approximated bound score. This experiment is repeated for a total of 100 data sets where 50 of them are generated by the CP model and the rest are generated by the TUCKER3 model. As a result of this experiment, for the CP generated data sets, 50 out of 50 data sets are identified correctly whereas for TUCKER3 generated data sets, 48 out of 50 data sets are identified correctly.

5.6.2. Experiment 2

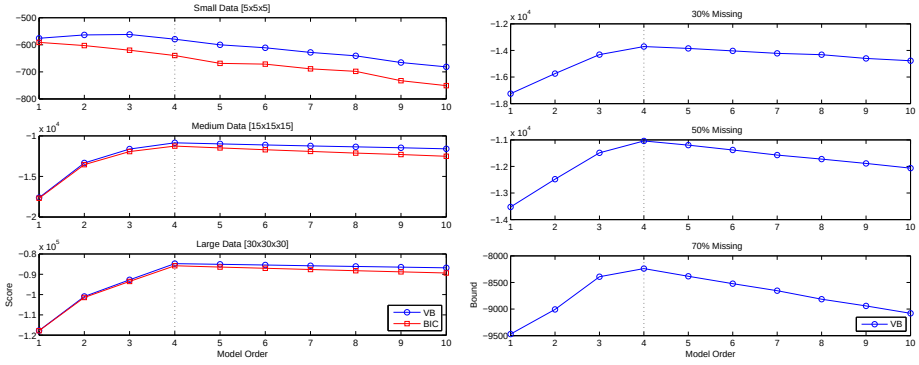
Here we compare VB and BIC for model selection of $PLTF_{KL}$ models and also test the missing data case on image dataset.

We use the following synthetic example for comparing VB and BIC for model selection for a CP factorization model. Three data sets are used to simulate small,

medium and large data sets that correspond to observed tensor sizes $5\times5\times5$, $15\times15\times15$ and $30 \times 30 \times 30$. The cardinality of the latent index (true model order) is set to 4 while the number of iteration is 300. The test is done up to model order 10. Each run is repeated 10 times and average is plotted in the figures as model order is on the x-axis while VB and BIC scores on the y-axes.

When there is large and moderate data in size VB and BIC give similar results. Both methods find the correct model order of 4. As model order increase, however, BIC score diverges from VB score. This is as expected, since increase in model order causes increase in the number of free parameters that, in turn, enlarges penalty term in BIC score.

On the other hand, model order selection performance under missing data case is shown in Figure 5.3. For this an observation array is multiplied by a mask array to simulate datasets with 30, 50 and 70 % missing data. True model order is set to 4 and the number of iterations is 300 as before.



(a) (b)

Figure 5.3. (a) is for model order selection comparison for CP generated data. Correct model order is 4 and data set sizes are $5 \times 5 \times 5$, $15 \times 15 \times 15$ and $30 \times 30 \times 30$ that correspond to insufficient, medium and large data sets. The gamma priors A and B are set to be 2,1 respectively. (b) is for model order selection under missing data case for CP generated data.

To illustrate the model selection and missing data performance of our model with

real data we use *Olivetti Faces* image database available at `http://www.cs.toronto.edu/~roweis/data.html`. The data set consists of 400 grayscale face images of 40 distinct people each with 10 different face orientations. The input tensor $X^{i,j,k}$ is constructed as i for 64×64 wide face image, j for face orientation index and k is for person index. As for the experiment settings, the gamma hyperparameters are set to 10 for scale and for shape for all the components, and the number of iterations is set to 1000. Again, the missing data is generated by 3 mask arrays generated randomly to simulated 30, 50 and 70% missing datasets. We test a CP model with model order $|r| = 7$ (as reported by best model order by BIC) and $|r| = 80$. Figure 5.4 and Figure 5.5 show the results.

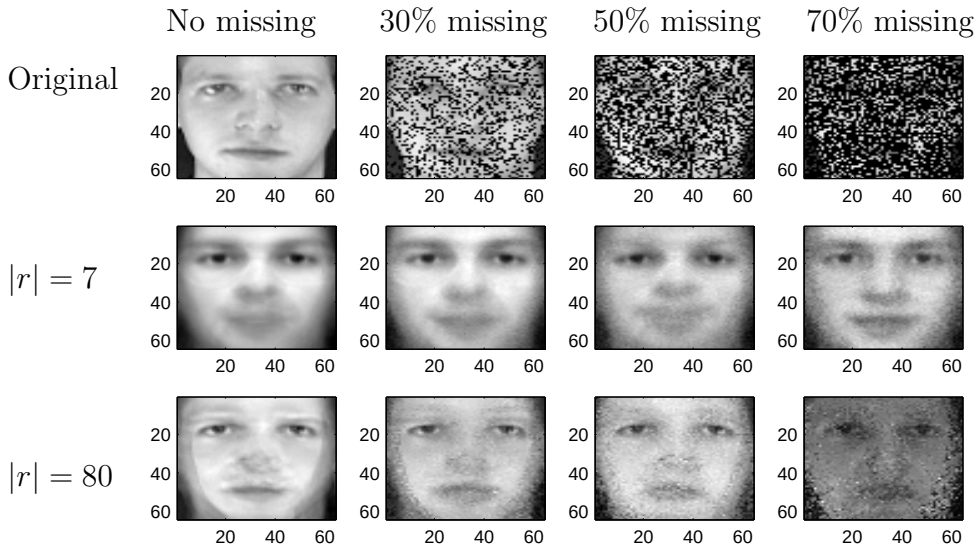


Figure 5.4. The figure shows missing data recovery by CP factorization model. Four images on the top row represent, original data and 3 incomplete data with 30%, 50% and 70% missing respectively without learning. Other two rows are the reconstructed images. Middle row is with model order set to 7, and bottom row is set to 80.

5.7. Summary

In this chapter we described a model selection framework for non-negative tensor factorization with KL cost from a probabilistic perspective that also handles the missing

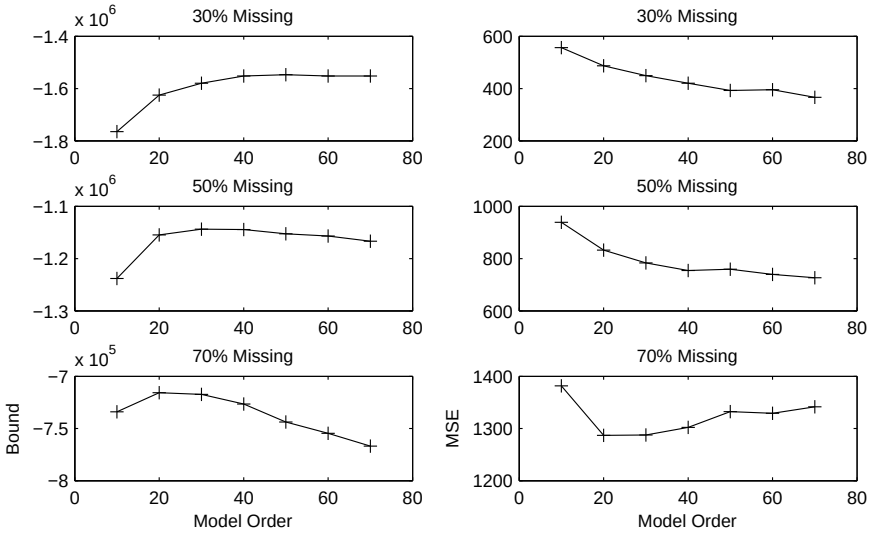


Figure 5.5. Left column represents the bound vs the model order while the right column shows MSE vs the model order. As expected when the amount of missing data increases the best model order decreases.

data naturally. The model comparison is based on variational Bayes that turns to be the criterion BIC asymptotically. There are three main parts in this chapter; the computation of approximation distribution, the likelihood bound and asymptotic analysis of the bound. First, practically we want to approximate the posterior by an approximation distribution $q(S, Z)$ that is assumed to be factorized as $q(S, Z) = q(S)q(Z)$. There is a prior knowledge on the parameters as the gamma distribution. Approximation of posterior $q(Z)$ is also identified as the gamma distribution

$$Z_\alpha(v_\alpha) \sim \mathcal{G}(Z_\alpha(v_\alpha); A_\alpha(v_\alpha), B_\alpha(v_\alpha)/A_\alpha(v_\alpha)) \qquad \text{prior} \tag{5.92}$$

$$Z_\alpha(v_\alpha) \sim \mathcal{G}(Z_\alpha(v_\alpha); C_\alpha(v_\alpha), D_\alpha(v_\alpha)) \qquad \text{posterior approximation} \tag{5.93}$$

where the shape and scale parameters for the posterior are

$$C_\alpha(v_\alpha) = A_\alpha(v_\alpha) + \sum_{\bar{v}_\alpha} M(v_0) \frac{X(v_0)}{\hat{X}_L(v_0)} \prod_\alpha L_\alpha(v_\alpha) \tag{5.94}$$

$$D_\alpha(v_\alpha) = \left(\frac{A_\alpha(v_\alpha)}{B_\alpha(v_\alpha)} + \sum_{\bar{v}_\alpha} M(v_0) \prod_{\alpha' \neq \alpha} \langle Z_{\alpha'}(v_{\alpha'}) \rangle \right)^{-1} \tag{5.95}$$

Then the expectation of the factor Z_α is identified as the mean of the gamma distribution and given in the following iterative fixed point update equation

$$\langle Z_\alpha(v_\alpha)\rangle = C_\alpha(v_\alpha)D_\alpha(v_\alpha) \tag{5.96}$$

where

$$L_\alpha(v_\alpha) = \exp\Big(\psi\big(C_\alpha(v_\alpha)\big)\Big)D_\alpha(v_\alpha) \qquad \text{and} \qquad \hat{X}_L(v_0) = \sum_{\bar{v}_0}\prod_{\alpha} L_\alpha(v_\alpha) \tag{5.97}$$

Secondly, we compute the variational bound $\mathcal{B}$ for the log marginal likelihood

$$\mathcal{L}(\Theta) \geq \langle \log p(X|S) + \log p(S|Z) + \log p(Z|\Theta)\rangle_{q(S,Z)} + H[q(S)] + H[q(Z)] \tag{5.98}$$

that is simplified and can be written compactly as

$$\mathcal{B} = \Big\{ \sum_{v_0} -\hat{X}_E(v_0) - \log\Gamma(X(v_0)+1) + X(v_0)\log\hat{X}_L(v_0)\Big\} \tag{5.99}$$

$$- \sum_{\alpha}\sum_{v_\alpha} KL[q(Z)||p(Z|\Theta)] \tag{5.100}$$

Thirdly, we further approximate the bound under large sample case. By dropping terms that do not scale asymptotically and approximating entropy of the posterior by the entropy of the Gaussian the bound becomes

$$\mathcal{B} = \log\mathcal{PO}(X;\hat{X}_E) + H[\mathcal{G}(Z;C_\alpha,D_\alpha)] \tag{5.101}$$

$$= \log\mathcal{PO}(X;\hat{X}_E) + \sum_{\alpha}^{|\alpha|}\frac{1}{2}\log\left(C_\alpha \circ D_\alpha^2\right) \tag{5.102}$$

Finally, BIC can be written as $PLTF_{KL}$ models as

$$\mathcal{B} \simeq \log\mathcal{PO}(X;\hat{X}_E) - \sum_{\alpha}^{|\alpha|}\frac{1}{2}M_\alpha \log N_\alpha \tag{5.103}$$

where the number of parameters and number of dimensions are identified as

$$N_\alpha = |\bar{v}_\alpha| \qquad\qquad M_\alpha = |v_\alpha| \tag{5.104}$$

6. GENERALIZED TENSOR FACTORIZATION

6.1. Introduction

The main motivation of this chapter is to construct a general and practical framework for computation of tensor factorizations, by extending the well-established theory of Generalized Linear Models (GLM). In Chapter 4 we take a similar approach to tensor factorizations via EM method, but that work is limited to Euclidean, KL and IS costs. We will extend the framework constructed here to address the coupled tensor factorization case which we analyse in detail in the next chapter.

The outline of this chapter is as follows. In Section 6.2 we briefly review GLM theory. Here we also discuss two equivalent representations of systems such that the vectorization and derivation of the tensors become equivalent and we make use of this equivalence to extend GLM to TF. In Section 6.3 we extend the theory of GLM to Generalized Tensor Factorization (GTF) by bounding the step size of the Fisher Scoring iteration of the GLM. We, then obtain two forms of fixed point update equations for real data and multiplicative updates for non-negative data.

6.2. Generalized Linear Models for Matrix/Tensor Factorization

To set the notation and our approach, we briefly review GLMs adapting the original notation of [76]. A GLM assumes that a data vector x has conditionally independently drawn components x_i according to an exponential dispersion model (EDM)

$$x_i \sim h(x_i, \varphi) \exp\left\{\varphi\left(\theta_i x_i - \psi(\theta_i)\right)\right\} \tag{6.1}$$

with the parameters

$$\langle x_i \rangle = \hat{x}_i = \frac{\partial \psi(\theta_i)}{\partial \theta_i} \qquad var(x_i) = \varphi^{-1} \frac{\partial^2 \psi(\theta_i)}{\partial \theta_i^2} \tag{6.2}$$

Here, to recall θ_i are the *canonical parameters* and φ^{-1} is a known dispersion parameter. $\langle x_i \rangle$ is the expectation of x_i and $\psi(\cdot)$ is the log partition function, enforcing normalization. The canonical parameters are not directly estimated, instead one assumes a link function $g(\cdot)$ that links the mean of the distribution $\hat{x}_i$ and assumes that $g(\hat{x}_i) = l_i^\top z$ where $l_i^\top$ is the ith row vector of a known model matrix L and z is the parameter vector to be estimated, $A^\top$ denotes matrix transpose of A. The model is linear in the sense that a function of the mean is linear in parameters, i.e., $g(\hat{x}) = Lz$. A *Linear Model (LM)* is a special case of GLM that assumes normality, i.e. $x_i \sim \mathcal{N}(x_i; \hat{x}_i, \sigma^2)$ as well as linearity that implies identity link function as $g(\hat{x}_i) = \hat{x}_i = l_i^\top z$ assuming l_i are known. Logistic regression assumes a log link, $g(\hat{x}_i) = \log \hat{x}_i = l_i^\top z$; here $\log \hat{x}_i$ and z have a linear relationship [76].

In addition, [67] defines three components in GLMs as follows.

(i) *Random component* : $\hat{x}_i$ is expected value of random variable x_i

$$E[x_i] = \langle x_i \rangle = \hat{x}_i \tag{6.3}$$

(ii) *Systematic component* : κ_i (also called as *linear predictor*)

$$\kappa_i = l_i^T z \tag{6.4}$$

(iii) *Link function* : $g()$ is a monotonic differential function that links the systematic component to the random component as

$$\kappa_i = g(\hat{x}_i) \tag{6.5}$$

The goal in classical GLM is to estimate the parameter vector z. This is typically achieved via a Gauss-Newton method (Fisher Scoring). The necessary objects for this computation are the log likelihood denoted by $\mathcal{L}$, its derivative and the Fisher Information (the expected value of negative of the Fisher Score). These are easily

derived as

$$\mathcal{L} = \sum_i \left\{ \varphi(x_i\theta_i - \psi(\theta_i)) + \log h(x_i, \varphi) \right\} \tag{6.6}$$

$$\frac{\partial \mathcal{L}}{\partial z} = \varphi \sum_i (x_i - \hat{x}_i) w_i g_{\hat{x}}(\hat{x}_i) l_i^\top \tag{6.7}$$

in matrix forms

$$\frac{\partial \mathcal{L}}{\partial z} = \varphi L^\top D G (x - \hat{x}) \qquad \left\langle \frac{\partial^2 \mathcal{L}}{\partial z^2} \right\rangle = \varphi L^\top D L \tag{6.8}$$

where w is a vector with elements w_i, D and G are the diagonal matrices as $D = diag(w)$, $G = diag(g_{\hat{x}}(\hat{x}_i))$ and

$$w_i = \left(v(\hat{x}_i) g_{\hat{x}}^2(\hat{x}_i) \right)^{-1} \qquad g_{\hat{x}}(\hat{x}_i) = \frac{\partial g(\hat{x}_i)}{\partial \hat{x}_i} \tag{6.9}$$

with $v(\hat{x}_i)$ being the *variance function* related to the observation variance by $\text{var}(x_i) = \varphi^{-1} v(\hat{x}_i)$. We note that Equation 6.7 is in accordance with the derivative of the log-likelihood in Equation 3.59 developed for the power variance functions in Chapter 3. Via Fisher Scoring, the general update equation in matrix form is written as

$$z \leftarrow z + \left(L^\top D L \right)^{-1} L^\top D G (x - \hat{x}) \tag{6.10}$$

Although this formulation is somewhat abstract, it covers a very broad range of model classes that are used in practice. For example, an important special case appears when the variance functions are in the form of $v(\hat{x}) = \hat{x}^p$. By setting $p = \{0, 1, 2, 3\}$ these correspond to Gaussian, Poisson, exponential/gamma, and inverse Gaussian distributions [67], which are special cases of the exponential family of distributions for any p named Tweedie's distributions [77] that are standardized by Jorgensen under the name exponential dispersion models [23].

6.2.1. Two Equivalent Representations via Vectorization

The key observation for expressing a tensor factorization model as a GLM is to identify equivalence of vector representation and derivation convention of a multilinear structure. To hide the notational complexity, we will give an example with a simple matrix factorization model; extension to tensors will require heavier notation, but are otherwise conceptually straightforward. Consider a MF model

$$\hat{X} = Z_1 Z_2 \qquad \text{in scalar} \qquad \hat{X}^{i,j} = \sum_r Z_1^{i,r} Z_2^{r,j} \tag{6.11}$$

where Z_1, Z_2 and $\hat{X}$ are matrices of compatible sizes. Indeed, by applying the **vec** operator (vectorization, stacking columns of a matrix to obtain a vector) to both sides of Equation 6.11 we obtain two equivalent representations of the same system

$$\mathbf{vec}(\hat{X}) = \mathbf{vec}(Z_1 Z_2) = (I_{|j|} \otimes Z_1)\,\mathbf{vec}(Z_2) \tag{6.12}$$

where $I_{|j|}$ denotes the $|j| \times |j|$ identity matrix, $\otimes$ denotes the Kronecker product [78]. Since Equation 6.11 and Equation 6.12 are two equivalent representations of the same system we have the following reverse implication

$$\mathbf{vec}(\hat{X}) = (I_{|j|} \otimes Z_1)\,\mathbf{vec}(Z_2) \qquad \Rightarrow \qquad \hat{X} = Z_1 Z_2 \tag{6.13}$$

On the other hand, definition of the matrix (and tensor) derivative is given as

$$\nabla_2 = \frac{\partial (Z_1 Z_2)}{\partial Z_2} = (I_{|j|} \otimes Z_1) \tag{6.14}$$

Thus, from Equation 6.12 and Equation 6.14 we conclude

$$\mathbf{vec}(\hat{X}) = (I_{|j|} \otimes Z_1)\,\mathbf{vec}(Z_2) = \frac{\partial (Z_1 Z_2)}{\partial Z_2}\,\mathbf{vec}(Z_2) = \frac{\partial \hat{X}}{\partial Z_2}\,\mathbf{vec}(Z_2) \equiv \nabla_2 \vec{Z}_2 \tag{6.15}$$

where $\mathbf{vec}\, Z \equiv \vec{Z}$ and ∇_2 is the gradient vector w.r.t. Z_2.

Considering the matrix derivation convention we use [78] which is quoted as 'Let F be a differentiable $m \times p$ real matrix function of an $n \times q$ matrix of real variables X. The Jacobian matrix of F at X is the $mp \times nq$ matrix'

$$DF(X) = \frac{\partial \, \mathbf{vec}\, F(X)}{\partial (\mathbf{vec}\, X)^T} \tag{6.16}$$

Here while the definition is for matrices, it can be easily extended for the tensors by regarding the vectorization of the tensor as vectorization of *mode*-1 unfolding [13] of the tensor, i.e. for the tensor Q it is as $\mathbf{vec}(Q_{(1)})$. Hence the definition turns to be

$$DF(Q) = \frac{\partial \ \mathbf{vec}\ F(Q)_{(1)}}{\partial \ (\mathbf{vec}\ Q_{(1)})^T} \tag{6.17}$$

6.3. Generalized Tensor Factorization

By using the development of previous section, we may use Tensors in GLM's. For this, we replace $\hat{X}$ by $g(\hat{X})$ and we simply consider a MF model

$$g(\hat{X}) = Z_1 Z_2 \qquad \text{in scalar} \qquad g(\hat{X})^{i,j} = \sum_r Z_1^{i,r} Z_2^{r,j} \tag{6.18}$$

where, in this case, Z_1, Z_2 and $g(\hat{X})$ are matrix versions of tensors (i.e. consider unfolding operation) of compatible sizes. Then, by applying the **vec** operator to both sides of Equation 6.18 we obtain

$$\mathbf{vec}(g(\hat{X})) = (I_{|j|} \otimes Z_1)\, \mathbf{vec}(Z_2) \tag{6.19}$$

$$= \frac{\partial (Z_1 Z_2)}{\partial Z_2}\, \mathbf{vec}(Z_2) = \frac{\partial g(\hat{X})}{\partial Z_2}\, \mathbf{vec}(Z_2) \equiv \nabla_2 \vec{Z}_2 \tag{6.20}$$

Clearly, this is a GLM where ∇_2 plays the role of a model matrix and $\vec{Z}_2$ is the parameter vector. Another key observation for expressing a tensor factorization model as a GLM is to use an *alternative optimization* approach [79]. By alternating between Z_1 and Z_2, we can maximize the log likelihood iteratively; indeed this alternating

maximization is standard for solving matrix factorization problems. In the sequel, we will show that a much broader range of algorithms can be readily derived in the GLM framework.

To extend this discussion to general tensor case, we need the equivalent of the model matrix, when updating Z_α. This is obtained by summing over the product of all remaining factors

$$g(\hat{X}(v_0)) = \sum_{\bar{v}_0} \prod_{\alpha} Z_\alpha(v_\alpha) \tag{6.21}$$

$$= \sum_{\bar{v}_0 \cap v_\alpha} \left\{ Z_\alpha(v_\alpha) \sum_{\bar{v}_0 \cap \bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'}) \right\} \tag{6.22}$$

$$= \sum_{\bar{v}_0 \cap v_\alpha} Z_\alpha(v_\alpha) L_\alpha(o_\alpha) \tag{6.23}$$

where $L_\alpha(o_\alpha)$ is defined as

$$L_\alpha(o_\alpha) = \sum_{\bar{v}_0 \cap \bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'}) \tag{6.24}$$

with the configuration

$$o_\alpha \equiv (v_0 \cup v_\alpha) \cap (\bar{v}_0 \cup \bar{v}_\alpha) \tag{6.25}$$

checking that in Equation 6.21 $Z_\alpha(v_\alpha)$ is independent of the sum $\sum_{\bar{v}_0 \cap \bar{v}_\alpha}$ and can be taken out of this sum.

Also noting that the two sums recover the original number of configurations $\bar{v}_0$

$$\sum_{\bar{v}_0} \equiv \sum_{\bar{v}_0 \cap v_\alpha} \sum_{\bar{v}_0 \cap \bar{v}_\alpha} \tag{6.26}$$

since $\bar{v}_0 = \big(\bar{v}_0 \cap v_\alpha\big) \cup \big(\bar{v}_0 \cap \bar{v}_\alpha\big)$ by observing

$$\big(\bar{v}_0 \cap v_\alpha\big) \cup \big(\bar{v}_0 \cap \bar{v}_\alpha\big) = \bar{v}_0 \cap \big(v_\alpha \cup \bar{v}_\alpha\big) = \bar{v}_0 \tag{6.27}$$

Remark 6.1. The actual reasoning for decomposing $\bar{v}_0$ is as follows. Since we will take the derivative w.r.t. $Z_\alpha(v_\alpha)$, for the inner sum, v_α terms will be irrelevant, and hence the inner sum has $\bar{v}_0/v_\alpha$ configurations. Then, the number of configurations left for the outer sum is the rest. i.e. $\bar{v}_0/(\bar{v}_0/v_\alpha)$.

$$\sum_{\bar{v}_0} \equiv \sum_{\bar{v}_0/(\bar{v}_0/v_\alpha)} \sum_{\bar{v}_0/v_\alpha} \tag{6.28}$$

Then by using the fact that for the sets A, B we have $A/B = A \cap \bar{B}$, we transform this equation to an equivalent representation in Equation 6.26.

To sum up, the derivations obtained so far we have the random component, the link function and systematic components tied as follows

$$g(\hat{X}(v_0)) = \sum_{\bar{v}_0 \cap v_\alpha} Z_\alpha(v_\alpha) L_\alpha(o_\alpha) \tag{6.29}$$

where here $L_\alpha(o_\alpha)$ is the model matrix while $Z_\alpha(v_\alpha)$ is the parameter to be identified. This equation, in addition, will help us get the derivative that we need in the update equation as

$$\frac{\partial g(\hat{X}(v_0))}{\partial Z_\alpha(v_\alpha)} = L_\alpha(o_\alpha) \tag{6.30}$$

since $(\bar{v}_0 \cap v_\alpha)/v_\alpha = (\bar{v}_0 \cap v_\alpha) \cap \bar{v}_\alpha = \{\}$ sum drops and only one configuration v_α is left.

One alternative analysis to get the derivative of the tensor $g(\hat{X})$ w.r.t. the latent

tensor Z_α denoted as ∇_α by following the convention [78]

$$\nabla_\alpha = \frac{\partial g(\hat{X})}{\partial Z_\alpha} = I_{|v_0 \cap v_\alpha|} \otimes L_\alpha \qquad \text{with } L_\alpha \in \mathbb{R}^{|v_0 \cap \bar{v}_\alpha| \times |\bar{v}_0 \cap v_\alpha|} \tag{6.31}$$

For the obtaining the dimensions of L_α, we start from the gradient object ∇_α which is $|v_0| \times |v_\alpha|$ object. Recall that for Kronecker product we have the Cartesian product of the configurations as $X_i^j \otimes Y_p^q = (X \otimes Y)_{ip}^{jq}$. Reducing $|v_0 \cap v_\alpha|$ the size of the identity matrix I from both dimension we end up with

$$Row : v_0/(v_0 \cap v_\alpha) = v_0 \cap (\bar{v}_0 \cup \bar{v}_\alpha) = v_0 \cap \bar{v}_\alpha \tag{6.32}$$

$$Col : v_\alpha/(v_0 \cap v_\alpha) = v_\alpha \cap (\bar{v}_0 \cup \bar{v}_\alpha) = \bar{v}_0 \cap v_\alpha \tag{6.33}$$

Also the configuration of the object L_α, i.e. an address of its cells is union of row and columns addressing

$$o_\alpha \equiv (\bar{v}_0 \cap v_\alpha) \cup (v_0 \cap \bar{v}_\alpha) = (v_0 \cup v_\alpha) \cap (\bar{v}_0 \cup \bar{v}_\alpha) \tag{6.34}$$

For example, for TUCKER3 the configuration of the object L_A, i.e. o_A is instantiations from the index set $\{j, k, p\}$.

The importance of L_α is that, all the update rules can be formulated by a product and subsequent contraction of L_α with another tensor Q having exactly the same index set of the observed tensor X. As a notational abstraction, it is useful to formulate the following function,

Definition 6.1. *The tensor valued function* $\Delta_\alpha(Q) : \mathbb{R}^{|v_0|} \to \mathbb{R}^{|v_\alpha|}$ *is defined as*

$$\Delta_\alpha^\varepsilon(Q) = \Big[\sum_{v_0 \cap \bar{v}_\alpha} Q(v_0)\ L_\alpha(o_\alpha)^\varepsilon \Big] \tag{6.35}$$

with $\Delta_\alpha(Q)$ being an object of the same order as Z_α and $o_\alpha \equiv (v_0 \cup v_\alpha) \cap (\bar{v}_0 \cup \bar{v}_\alpha)$. Here, on the right side, the non-negative integer ε denotes the element-wise

power, not to be confused with an index. On the left, it should be interpreted as a parameter of the Δ function. Arguably, Δ function abstracts away all the tedious reshape and unfolding operations [13]. This abstraction has also an important practical facet: the computation of Δ is algebraically (almost) equivalent to computation of marginal quantities on a factor graph, for which efficient message passing algorithms exist [8].

Example 6.1. *TUCKER3 is defined as* $\hat{X}^{i,j,k} = \sum_{p,q,r} A^{i,p}B^{j,q}C^{k,r}G^{p,q,r}$ *with* $\mathcal{V} = \{i,j,k,p,q,r\}$, $\mathcal{V}_0 = \{i,j,k\}$, $\mathcal{V}_A = \{i,p\}$, $\mathcal{V}_B = \{j,q\}$, $\mathcal{V}_C = \{k,r\}$, $\mathcal{V}_G = \{p,q,r\}$. *Then for the first factor A, the objects* L_A *and* $\Delta^{\varepsilon}_A()$ *are computed as follows*

$$L_A = \left[\frac{\partial \hat{X}^{i,j,k}}{\partial A^{i,p}}\right] = \left[\sum_{q,r} B^{j,q}C^{k,r}G^{p,q,r}\right] = \left[\left((C \otimes B)G^{\top}\right)^{p}_{k,j}\right] = \left[\left(L_A\right)^{p}_{k,j}\right] \quad (6.36)$$

$$\Delta^{\varepsilon}_A(Q) = \left[\sum_{j,k} Q^{k,j}_i \left(L^{\varepsilon}_A\right)^{p}_{k,j}\right] = \left[\left(QL^{\varepsilon}_A\right)^{p}_{i}\right] \equiv Q_{(1)}\Big((C \otimes B)G^{T}_{(1)}\Big)^{\varepsilon} \quad (6.37)$$

The index sets marginalized out for L_A *and* Δ_A *are* $\bar{\mathcal{V}}_0 \cap \bar{\mathcal{V}}_A = \{p,q,r\} \cap \{j,q,k,r\} = \{q,r\}$ *and* $\mathcal{V}_0 \cap \bar{\mathcal{V}}_A = \{i,j,k\} \cap \{j,q,k,r\} = \{j,k\}$.

For the core tensor $G^{p,q,r}$ *of TUCKER3*

$$L_G = \left[\frac{\partial \hat{X}^{i,j,k}}{\partial G^{p,q,r}}\right] = \left[\sum_{\{\}} A^{i,p}B^{j,q}C^{k,r}\right] = \left[A^{i,p}B^{j,q}C^{k,r}\right] \quad (6.38)$$

$$\Delta^{\varepsilon}_G(Q) = \left[\sum_{i,j,k} Q^{i,j,k}\left(A^{i,p}B^{j,q}C^{k,r}\right)^{\varepsilon}\right] = \left(A^{T}\right)^{\varepsilon} Q_{(1)}(C^{\varepsilon} \otimes B^{\varepsilon}) \quad (6.39)$$

The index sets marginalized out for L_G *and* Δ_G *are* $\bar{\mathcal{V}}_0 \cap \bar{\mathcal{V}}_G = \{p,q,r\} \cap \{i,j,k\} = \{\}$ *and* $\mathcal{V}_0 \cap \bar{\mathcal{V}}_G = \{i,j,k\} \cap \{i,j,k\} = \{i,j,k\}$.

Remark 6.2. When we plug $L_\alpha(o_\alpha)$ into the definition of the Δ function

$$\Delta^{\varepsilon}_\alpha(Q) = \Big[\sum_{v_0 \cap \bar{v}_\alpha} Q(v_0)\; L_\alpha(o_\alpha)^{\varepsilon}\Big] = \left[\sum_{v_0 \cap \bar{v}_\alpha} Q(v_0)\; \left\{\sum_{\bar{v}_0 \cap \bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'})\right\}^{\varepsilon}\right] \quad (6.40)$$

and when we ignore the element-wise power ϵ for a while, we note that $Q(v_0)$ can be

pushed into the inner sum, since the configuration v_0 is independent of the inner sum's index $\bar{v}_0 \cap \bar{v}_\alpha$ as

$$\Delta_\alpha^\varepsilon(Q) = \left[\sum_{v_0 \cap \bar{v}_\alpha} \sum_{\bar{v}_0 \cap \bar{v}_\alpha} Q(v_0) \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'})\right] = \left[\sum_{\bar{v}_\alpha} Q(v_0) \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'})\right] \tag{6.41}$$

and note the union of the configurations

$$(v_0 \cap \bar{v}_\alpha) \cup (\bar{v}_0 \cap \bar{v}_\alpha) = (v_0 \cup \bar{v}_0) \cap \bar{v}_\alpha = \bar{v}_\alpha \tag{6.42}$$

This form of Δ function where we marginalize out all the configurations $\bar{v}_\alpha$ is exactly what we defined in Chapter 4 for PLTF. However, there is still a difference in the definitions of both Δ functions as below

$$\Delta_\alpha(Q) = \left[\sum_{\bar{v}_\alpha} Q(v_0) \frac{\partial X(v_0)}{\partial Z_\alpha(v_\alpha)}\right] \qquad \text{GTF (this) definition} \tag{6.43}$$

$$\Delta_\alpha(Q) = \left[\sum_{\bar{v}_\alpha} Q(v_0) \frac{\partial \Lambda(v)}{\partial Z_\alpha(v_\alpha)}\right] \qquad \text{PLTF (Ch. 4) definition} \tag{6.44}$$

The difference is that in the PLTF version of Δ function the derivative $\frac{\partial \Lambda(v)}{\partial Z_\alpha(v_\alpha)}$ does not yield any sum while the derivative $\frac{\partial X(v_0)}{\partial Z_\alpha(v_\alpha)}$ may end up with a sum just as in the case of TUCKER3. The element-wise power ϵ of the derivative in the Δ function of GTF version prevents Q to push into the inner sum.

6.3.1. Iterative Solution for GTF

As we have now established a one to one relationship between GLM and GTF objects such as the observation $x \equiv$ vec X, the mean (and the model estimate) $\hat{x} \equiv$ vec $\hat{X}$, the model matrix $L \equiv L_\alpha$ and the parameter vector $z \equiv$ vec Z_α, we can write directly from Equation 6.10 as

$$\vec{Z}_\alpha \leftarrow \vec{Z}_\alpha + \left(\nabla_\alpha^\top D \nabla_\alpha\right)^{-1} \nabla_\alpha^\top DG(\vec{X} - \vec{\hat{X}}) \qquad \text{with } \nabla_\alpha = \frac{\partial g(\hat{X})}{\partial Z_\alpha} \tag{6.45}$$

with $D = diag(\vec{W})$, i.e. D is $|v_0| \times |v_0|$ diagonal matrix whose diagonal is formed by $\vec{W}$ with the order $|v_0| \times 1$. Note also that ∇_α is $|v_0| \times |v_\alpha|$ matrix where here we follow the matrix derivation convention in [78]. To check the orders of the matrices, the inversion is formed by multiplication of three matrices with $|v_\alpha| \times |v_0|$, $|v_0| \times |v_0|$, and $|v_0| \times |v_\alpha|$ in the order resulting to $|v_\alpha| \times |v_\alpha|$ after multiplication and matrix inversion. On the other hand, the remaining terms are formed by multiplication of four matrices with the dimensions $|v_\alpha| \times |v_0|$, $|v_0| \times |v_0|$, $|v_0| \times |v_0|$ and finally $|v_0| \times 1$ respectively resulting to $|v_\alpha| \times 1$ which can be unfolded or reshaped back to the related factor Z_α. The orders of the matrices are summarized in Table 6.1. Under these settings vec X turns to be

Table 6.1. Orders of various objects in Equation 6.45.

Object	**Size**
$\vec{Z}_\alpha$	$\|v_\alpha\| \times 1$
D	$\|v_0\| \times \|v_0\|$
∇_α	$\|v_0\| \times \|v_\alpha\|$
G	$\|v_0\| \times \|v_0\|$
$\vec{X} - \vec{\hat{X}}$	$\|v_0\| \times 1$

a random vector with each component $X(v_0)$ being independently drawn from

$$X(v_0) \sim h\big(X(v_0), \varphi\big) \exp\Big\{\varphi\Big(X(v_0)\Gamma(v_0) - \psi\big(\Gamma(v_0)\big)\Big)\Big\} \tag{6.46}$$

$$\langle X(v_0)\rangle = \hat{X}(v_0) = \frac{\partial\psi\big(\Gamma(v_0)\big)}{\partial\Gamma(v_0)} \tag{6.47}$$

$$var(X(v_0)) = \varphi^{-1}\frac{\partial^2\psi(\Gamma(v_0))}{\partial\Gamma(v_0)^2} \equiv \varphi^{-1}v(\hat{X}(v_0)) \tag{6.48}$$

with $\Gamma(v_0)$ being the canonical parameter and with $v(\hat{X}(v_0))$ being the *variance function*. The link function $g()$ links the mean of the distribution $\hat{X}$ and the linear form of the predictors as $g(\hat{X}(v_0)) = \sum_{\bar{v}_0}\prod_\alpha Z_\alpha(v_\alpha)$. D and G are the diagonal matrices as $D = diag(\text{vec } W)$ and $G = diag(\text{vec } g_{\hat{X}}(\hat{X}))$ such that $W = \big[W(v_0)\big]$,

$g_{\hat{X}}(\hat{X}) = \left[g_{\hat{X}}(\hat{X}(v_0))\right]$ those in turn

$$W(v_0) = \left(v(\hat{X}(v_0))g^2_{\hat{X}}(\hat{X}(v_0))\right)^{-1} \qquad g_{\hat{X}}(\hat{X}(v_0)) = \frac{\partial g(\hat{X}(v_0))}{\partial \hat{X}(v_0)} \tag{6.49}$$

There are at least two ways that this update can further simplified. We may assume an identity link function, or alternatively we may choose a matching link and lost functions such that they cancel each other smoothly [80]. In the sequel we consider identity link $g(\hat{X}) = \hat{X}$ that results to $g_{\hat{X}}(\hat{X}) = \mathbf{1}$. This implies G to be identity, i.e. $G = I$. We define a tensor W, that plays the same role as w in (6.9), which becomes simply the precision (inverse variance function), i.e. $W = 1/v(\hat{X})$ where for the Gaussian, Poisson, Exponential and Inverse Gaussian distributions we have simply $W = \hat{X}^{-p}$ with $p = \{0, 1, 2, 3\}$ [67]. Then, the update in Equation 6.45 is reduced to

$$\vec{Z}_\alpha \leftarrow \vec{Z}_\alpha + \left(\nabla_\alpha^\top D \nabla_\alpha\right)^{-1} \nabla_\alpha^\top D(\vec{X} - \vec{\hat{X}}) \tag{6.50}$$

After this simplification we obtain two update rules for GTF for non-negative and real data.

6.3.2. Update Rules for Non-Negative GTF

The update in Equation 6.50 can be used to derive multiplicative update rules (MUR) popularized by [5] for the nonnegative matrix factorization (NMF). MUR equations ensure the non-negative parameter updates as long as starting some non-negative initial values.

Theorem 6.1. *The update in Equation 6.50 for nonnegative GTF can be expressed in multiplicative form as*

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\Delta_\alpha(W \circ X)}{\Delta_\alpha(W \circ \hat{X})} \qquad s.t.\ Z_\alpha(v_\alpha) > 0 \tag{6.51}$$

For the special case of the Tweedie family where the precision is a function of the

mean as $W = \hat{X}^{-p}$ for $p = \{0, 1, 2, 3\}$ the update in Equation 6.50 is reduced to

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\Delta_\alpha(\hat{X}^{-p} \circ X)}{\Delta_\alpha(\hat{X}^{1-p})} \tag{6.52}$$

Example 6.2. *For example, to update Z_2 for the NMF model $\hat{X} = Z_1 Z_2$, Δ_2 is $\Delta_2(Q) = Z_1^\top Q$. Then for the Gaussian ($p = 0$) this reduces to NMF-EU as*

$$Z_2 \leftarrow Z_2 \circ \frac{Z_1^\top X}{Z_1^\top \hat{X}} \tag{6.53}$$

For the Poisson ($p = 1$) it reduces to NMF-KL as [5]

$$Z_2 \leftarrow Z_2 \circ \frac{Z_1^\top (X/\hat{X})}{Z_1^\top \mathbf{1}} \tag{6.54}$$

Proof. Consider and re-arrange the derivative $\partial\mathcal{L}/\partial Z_\alpha$

$$\frac{\partial \mathcal{L}}{\partial Z_\alpha} = \nabla_\alpha^\top D(\vec{X} - \vec{\hat{X}}) = -\nabla_\alpha^\top D\vec{\hat{X}} + b = -\big(\nabla_\alpha^\top D \nabla_\alpha\big)\vec{Z}_\alpha + b = -H\vec{Z}_\alpha + b$$

where b is a constant and the substitution $H = \nabla_\alpha^\top D \nabla_\alpha$ is for notational simplicity. Here we use the equality $\vec{\hat{X}} = \nabla_\alpha \vec{Z}_\alpha$. In addition, note that H as the coefficient of the variable Z_α is symmetric positive definite since D is a diagonal matrix as $D = diag(\vec{W})$, so that any step size bounded appropriately as below converges [81]

$$\eta_{max} \leq \frac{2}{\lambda_{max}(H)}$$

We further find another bound by use of the Perron-Frobenious theorem [82]

$$\eta_{max} \leq \frac{2}{\lambda_{max}(H)} \qquad \lambda_{max}(H) \leq \max_{v_\alpha} \frac{\big(H\vec{Z}_\alpha\big)(v_\alpha)}{Z_\alpha(v_\alpha)} \qquad \text{s.t. } Z_\alpha(v_\alpha) > 0 \tag{6.55}$$

where λ_{max} being the maximum eigenvalue of the matrix H noting that the product $H\vec{Z}_\alpha$ has the order $|v_\alpha| \times 1$. In addition λ_{max} is known as Perron root or the Perron-Frobenius eigenvalue equal to the spectral radius. Identifying the right hand side of

$H\vec{Z}_\alpha$ as $\vec{\hat{X}} = \nabla_\alpha \vec{Z}_\alpha$, i.e.

$$(\nabla_\alpha^T D \nabla_\alpha)\vec{Z}_\alpha = \nabla_\alpha^T D(\nabla_\alpha \vec{Z}_\alpha) = \nabla_\alpha^T D\vec{\hat{X}} \tag{6.56}$$

choosing the step size specifically as

$$\eta = \frac{\vec{Z}_\alpha}{\nabla_\alpha^T D\vec{\hat{X}}} < \frac{2\vec{Z}_\alpha}{\nabla_\alpha^T D\vec{\hat{X}}} \leq \frac{2}{\lambda_{max}(\nabla_\alpha^T D \nabla_\alpha)} \tag{6.57}$$

we end up with the iterative update as

$$\vec{Z}_\alpha \leftarrow \vec{Z}_\alpha + \frac{\vec{Z}_\alpha}{\nabla_\alpha^T D\vec{\hat{X}}} \circ \nabla_\alpha^T D(\vec{X} - \vec{\hat{X}}) \tag{6.58}$$

$$\leftarrow \vec{Z}_\alpha \circ \frac{\nabla_\alpha^T D\vec{X}}{\nabla_\alpha^T D\vec{\hat{X}}} \tag{6.59}$$

$$\leftarrow \vec{Z}_\alpha \circ \frac{\nabla_\alpha^T(\vec{W} \circ \vec{X})}{\nabla_\alpha^T(\vec{W} \circ \vec{\hat{X}})} \tag{6.60}$$

Then after identifying $\nabla_\alpha = (I \otimes L_\alpha)$ that, in turn, $\nabla_\alpha^T = (I^T \otimes L_\alpha^T)$ and plugging

$$\vec{Z}_\alpha \leftarrow \vec{Z}_\alpha \circ \frac{(I^T \otimes L_\alpha^T)(\vec{W} \circ \vec{X})}{(I^T \otimes L_\alpha^T)(\vec{W} \circ \vec{\hat{X}})} \tag{6.61}$$

Here we may drop vec operator by use of [83] and recall the equation

$$\mathbf{vec}(\hat{X}) = \mathbf{vec}(Z_1 Z_2) = (I_{|j|} \otimes Z_1)\,\mathbf{vec}(Z_2) \tag{6.62}$$

where we come up

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{L_\alpha^T(W \circ X)}{L_\alpha^T(W \circ \hat{X})} \tag{6.63}$$

Then after, by using the $\Delta_\alpha()$ function, i.e. by casting the product $L_\alpha^T(W \circ X)$ into $\Delta_\alpha()$ we arrive to Theorem 6.1. □

Remark 6.3. We already point out that L_α is an object of $|v_0 \cap \bar{v}_\alpha| \times |\bar{v}_0 \cap v_\alpha|$, and thus its transpose is $|\bar{v}_0 \cap v_\alpha| \times |v_0 \cap \bar{v}_\alpha|$. Next, $W \circ X$ is an object of $|v_0| \times 1$ (in vectorized dimension). $W \circ X$ is going to be reshaped as a matrix of $|v_0 \cap \bar{v}_\alpha| \times |v_0/(v_0 \cap \bar{v}_\alpha)|$ by the Δ function leaving the product $L_\alpha^T(W \circ X)$ as an object of

$$\{|\bar{v}_0 \cap v_\alpha| \times |v_0 \cap \bar{v}_\alpha|\} \times \{|v_0 \cap \bar{v}_\alpha| \times |v_0/(v_0 \cap \bar{v}_\alpha)|\} \tag{6.64}$$

$$\equiv |\bar{v}_0 \cap v_\alpha| \times |v_0/(v_0 \cap \bar{v}_\alpha)| \tag{6.65}$$

Clearly this is an object of the same size as Z_α since the shape is

$$\big(\bar{v}_0 \cap v_\alpha\big) \cup \big(v_0/(v_0 \cap \bar{v}_\alpha)\big) = \big(\bar{v}_0 \cap v_\alpha\big) \cup \big(v_0 \cap (\bar{v}_0 \cup v_\alpha)\big) \tag{6.66}$$

$$= \big(\bar{v}_0 \cap v_\alpha\big) \cup \big(v_0 \cap v_\alpha\big) = \big(\bar{v}_0 \cap v_\alpha\big) \cup \big(v_0 \cap v_\alpha\big) \tag{6.67}$$

$$= v_\alpha \tag{6.68}$$

where we replace multiplication $\times$ by union $\cup$ and replace cardinality symbol $|.|$ by the usual parenthesis $(.)$.

Since we work with multidimensional objects and their abstract dimensions and configurations we find useful to get the size of objects as first-level checking for correctness of the equations.

The important point to note that the MUR equation requires the non-negativity as $Z_\alpha(v_\alpha) > 0$ since Equation 6.55 holds only for positive values. In addition, otherwise for catching a positive value, starting from a negative initial point would never escape from the origin due to the nature of the multiplication.

6.3.3. General Update Rule for GTF

By dropping the non-negativity requirement we obtain the following update equation:

Theorem 6.2. *The update equation for GTF with real data can be expressed as*

$$Z_\alpha \leftarrow Z_\alpha + \frac{2}{\lambda_{\alpha/0}} \frac{\Delta_\alpha(W \circ (X - \hat{X}))}{\Delta_\alpha^2(W)} \tag{6.69}$$

where $\lambda_{\alpha/0}$ *is the cardinality of the configuration* $v_\alpha \cap \bar{v}_0$

$$\lambda_{\alpha/0} = |v_\alpha \cap \bar{v}_0| = |v_\alpha / v_0| \tag{6.70}$$

and the power is scalar type, i.e. $\Delta_\alpha^2(W) = \Delta_\alpha(W) \circ \Delta_\alpha(W)$

The update equation for $W = \hat{X}^{-p}$ is

$$Z_\alpha \leftarrow Z_\alpha + \frac{2}{\lambda_{\alpha/0}} \frac{\Delta_\alpha(\hat{X}^{-p} \circ (X - \hat{X}))}{\Delta_\alpha^2(\hat{X}^{-p})} \tag{6.71}$$

Example 6.3. *By using* $\lambda_{\alpha/0} = |v_\alpha \cap \bar{v}_0| = |v_\alpha / v_0|$*, we extract* $\lambda_{\alpha/0}$ *for* $A^{i,p}$ *and* $G^{p,q,r}$ *(core tensor) of TUCKER3 as*

$$v_A \cap \bar{v}_0 = \{i,p\}/\{i,j,k\} = \{p\} \quad \Rightarrow \quad \lambda_{\alpha/0} = |p| \quad \textit{for } A^{i,p} \tag{6.72}$$

$$v_G \cap \bar{v}_0 = \{p,q,r\}/\{i,j,k\} = \{p,q,r\} \quad \Rightarrow \quad \lambda_{\alpha/0} = |p||q||r| \quad \textit{for } G^{p,q,r} \tag{6.73}$$

Proof. Following Equation 6.57 and taking $Z_\alpha = \mathbf{1}$ specially, i.e. $Z_\alpha(v_\alpha) = 1$ leads to

$$\vec{Z}_\alpha \leftarrow \vec{Z}_\alpha + \frac{2}{\nabla_\alpha^T D \nabla_\alpha \mathbf{1}} \circ \nabla_\alpha^T D(\vec{X} - \vec{\hat{X}}) \quad \text{since } \lambda_{max}(H) \leq \max_{v_\alpha} \big(H\mathbf{1}\big)(v_\alpha) \tag{6.74}$$

We may replace the matrix multiplication of $\nabla_\alpha^\top D \nabla_\alpha \mathbf{1}_\alpha$ by $(\nabla_\alpha^\top)^2 D \mathbf{1_0} \lambda_{\alpha/0}$ by using the bound

$$\|\nabla_\alpha^T D \nabla_\alpha \mathbf{1}\| \leq \|(\nabla_\alpha^T)^2 D \mathbf{1} \lambda_{\alpha/0}\| \tag{6.75}$$

where $\mathbf{1}_\alpha$ is a $|v_\alpha|$-dim column vector of all ones and $\mathbf{1_0}$ is $|v_0|$-dim column vector of all

ones. Noting that $(\nabla_\alpha^T)^2(D\mathbf{1_0}) = (\nabla_\alpha^T)^2(\vec{W} \circ \mathbf{1_0}) = (\nabla_\alpha^T)^2\vec{W}$ the update becomes

$$\vec{Z}_\alpha \leftarrow \vec{Z}_\alpha + \frac{2}{\lambda_{\alpha/0}(\nabla_\alpha^T)^2\vec{W}} \circ \nabla_\alpha^T\big(\vec{W} \circ (\vec{X} - \hat{\vec{X}})\big) \tag{6.76}$$

Finally after plugging $\nabla_\alpha^T = (I^T_{|v_0 \cap v_\alpha|} \otimes L_\alpha^T)$

$$\vec{Z}_\alpha \leftarrow \vec{Z}_\alpha + \frac{2}{\lambda_{\alpha/0}(I^T_{|v_0 \cap v_\alpha|} \otimes L_\alpha^T)^2\vec{W}} \circ (I^T_{|v_0 \cap v_\alpha|} \otimes L_\alpha^T)\big(\vec{W} \circ (\vec{X} - \hat{\vec{X}})\big) \tag{6.77}$$

and by use of Equation 6.62 we remove the vec operation

$$Z_\alpha \leftarrow Z_\alpha + \frac{2}{\lambda_{\alpha/0}} \frac{L_\alpha^T\big(W \circ (X - \hat{X})\big)}{(L_\alpha^T)^2 W} \qquad \text{with } (L_\alpha^T)^2 = L_\alpha^T \circ L_\alpha^T \tag{6.78}$$

and after reformulating (reshaping) by $\Delta_\alpha()$ we come up with Theorem 6.2. Here the multiplier 2 in the numerator comes from Equation 6.55. □

Remark 6.4. Here as part of the proof we propose the following norm bound in Equation 6.75

$$\|\nabla_\alpha^T D \nabla_\alpha \mathbf{1}\| \leq \|(\nabla_\alpha^T)^2 D\mathbf{1}\lambda_{\alpha/0}\| \tag{6.79}$$

Recall that ∇_α is a special type of sparse matrix as $\nabla_\alpha = I_{|v_0 \cap v_\alpha|} \otimes L_\alpha$. Here the scalar $\lambda_{\alpha/0} = |v_\alpha / v_0| = |\bar{v}_0 \cap v_\alpha|$ is the number of the non-zero columns in any rows of ∇_α since from left to right only one block in ∇_α is L_α and the rest is zero. L_α is an object of $|v_0 \cap \bar{v}_\alpha| \times |\bar{v}_0 \cap v_\alpha|$, and hence $\lambda_{\alpha/0}$ is the number of columns of L_α. Also note that the number of the non-zero columns in any rows of $\nabla_\alpha^T \nabla_\alpha$ is also $\lambda_{\alpha/0}$.

D in Equation 6.79 is a diagonal matrix as $D = diag(\vec{W})$ whose entries are non-zero. Ignoring it in the equation we have the general re-formulation of the bound in Equation 6.79. For a given $n \times m$ matrix A we have the following inequality

Lemma 6.1. *For the matrix $A \in \mathbb{R}^{n\times m}$ we have the following bound*

$$\frac{\|AA^T\mathbf{1}_n\|_2}{\|(A\circ A)\mathbf{1}_m\|_2} \leq n \tag{6.80}$$

where here $\|.\|_2$ is vector 2-norm while $\mathbf{1}_n$ and $\mathbf{1}_m$ are n-dim and m-dim ones vectors.

Proof. We use the following vector norm bounds for the vector $x \in \mathbb{R}^n$

$$\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2 \tag{6.81}$$

Then for the denominator

$$\sqrt{n}\|(A\circ A)\mathbf{1}_m\|_2 \geq \|(A\circ A)\mathbf{1}_m\|_1 = 1_m^T(A\circ A)\mathbf{1}_m = Tr(AA^T) \tag{6.82}$$

Here $A\circ A$ is a positive matrix. For the numerator, let $\lambda_{max}, \ldots, \lambda_{min}$ be the eigenvalues of AA^T where all λ_i are positive.

$$\|AA^T\mathbf{1}_n\|_2 \leq \|AA^T\|_2\|\mathbf{1}_n\|_2 = \lambda_{max}\sqrt{n} \tag{6.83}$$

$$\leq \left(\sum_i \lambda_i\right)\sqrt{n} = Tr(AA^T)\sqrt{n} \leq n\|(A\circ A)\mathbf{1}_m\|_2 \tag{6.84}$$

□

One view of the term $\lambda_{\alpha/0}$ during the iterative update of the tensor Z_α is that Z_α is composed of many cells updated as a block simultaneously. However while Z_α is being updated its certain elements are used to compute $\hat{X}$. On the other hand we need to compute the derivative $\partial\hat{X}/\partial Z_\alpha$ to get the new direction to move along with. The number of such elements (cells) of Z_α, i.e. the number of configurations (each configuration is a cell), that are also needed to compute $\hat{X}$ is exactly $\bar{v}_0 \cap v_\alpha$ which is $\lambda_{\alpha/0} = |v_\alpha/v_0| = |\bar{v}_0 \cap v_\alpha|$. Recall that we marginalize out $\bar{v}_0$ configurations to obtain $\hat{X}$. Then, dividing by $\lambda_{\alpha/0}$ effectively decreases the step size, in average, hence prevents

from escaping from the local maximum. If we update the cells in Z_α one by one, i.e. scalar update where after each update we compute $\hat{X}$, $\lambda_{\alpha/0}$ could be taken just by one.

6.3.4. Handling of the Missing Data

Missing data case can be handled easily by dropping the missing data terms from the likelihood [29]. The net effect of this is the addition of an indicator variable m_i to the gradient $\partial\mathcal{L}/\partial z$ of GLM as

$$\frac{\partial \mathcal{L}}{\partial z} = \varphi \sum_i (x_i - \hat{x}_i) m_i w_i g_{\hat{x}}(\hat{x}_i) l_i^\top \tag{6.85}$$

with $m_i = 1$ if x_i is observed otherwise $m_i = 0$. Hence we simply define a mask tensor M having the same order as the observation X, where the element $M(v_0)$ is 1 if $X(v_0)$ is observed and zero otherwise. In the update equations, we merely replace W with $W \circ M$.

6.4. Summary

This chapter first establishes a link between the theory of generalized linear models and tensor factorizations and second it provides a general solution for the computation of arbitrary tensor factorizations for the exponential dispersion models family.

The key observation for expressing a tensor factorization model as a GLM is to identify equivalence of vectorization operation and derivation of a multilinear structure. To illustrates this we apply the **vec** operator to both sides of $g(\hat{X}) = Z_1 Z_2$

$$\mathbf{vec}(g(\hat{X})) = (I_{|j|} \otimes Z_1)\, \mathbf{vec}(Z_2) = \frac{\partial (Z_1 Z_2)}{\partial Z_2}\, \mathbf{vec}(Z_2) \tag{6.86}$$

$$= \frac{\partial g(\hat{X})}{\partial Z_2}\, \mathbf{vec}(Z_2) \equiv \nabla_2 \vec{Z}_2 \tag{6.87}$$

Here we end up with a GLM such that here ∇_2 is the model matrix and $\vec{Z}_2$ is the parameter vector while in an *alternating optimization* procedure they switch the roles.

For general case, when updating Z_α, the derivative of the tensor $g(\hat{X})$ is defined as

$$\nabla_\alpha = \frac{\partial g(\hat{X})}{\partial Z_\alpha} = I_{|v_0 \cap v_\alpha|} \otimes L_\alpha \tag{6.88}$$

The importance of L_α is that it plays exactly the same role of $\prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'})$, and all the updates can be formulated by exactly the Δ function defined in Chapter 4 as

$$\Delta_\alpha^\varepsilon(Q) = \Big[\sum_{v_0 \cap \bar{v}_\alpha} Q(v_0)\ L_\alpha(o_\alpha)^\varepsilon \Big] \tag{6.89}$$

where here the configuration is $o_\alpha \equiv (v_0 \cup v_\alpha) \cap (\bar{v}_0 \cup \bar{v}_\alpha)$. After establishing a one to one relationship between GLM and GTF objects we can write the update as

$$\vec{Z}_\alpha \leftarrow \vec{Z}_\alpha + \left(\nabla_\alpha^\top D \nabla_\alpha \right)^{-1} \nabla_\alpha^\top D G (\vec{X} - \vec{\hat{X}}) \tag{6.90}$$

with $D = diag(\vec{W})$ is diagonal matrix whose diagonal is formed by $\vec{W}$. This update can further be simplified under two assumptions. The first is that we assume identity link function and the second we assume non-negative parameters, i.e. $Z_\alpha(v_\alpha) > 0$. Under these assumptions the update turns to be

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\Delta_\alpha(W \circ X)}{\Delta_\alpha(W \circ \hat{X})} \tag{6.91}$$

Next by relaxing the non-negativity requirement the update turns to be

$$Z_\alpha \leftarrow Z_\alpha + \frac{2}{\lambda_{\alpha/0}} \frac{\Delta_\alpha(W \circ (X - \hat{X}))}{\Delta_\alpha^2(W)} \tag{6.92}$$

with $\lambda_{\alpha/0}$ being the cardinality of the configuration $v_\alpha \cap \bar{v}_0$.

7. GENERALIZED COUPLED TENSOR FACTORIZATION

7.1. Introduction

The main motivation of this chapter is to construct a generalized coupled tensor factorization framework, by extending GTF approach developed in Chapter 6. As already pointed out tensors appear as a natural generalization of matrix factorization, when observed data and/or a latent representation have several semantically meaningful dimensions. Now consider the following motivating example

$$X_1^{i,j,k} \simeq \sum_r Z_1^{i,r} Z_2^{j,r} Z_3^{k,r} \qquad X_2^{j,p} \simeq \sum_r Z_2^{j,r} Z_4^{p,r} \qquad X_3^{j,q} \simeq \sum_r Z_2^{j,r} Z_5^{q,r} \tag{7.1}$$

where X_1 is an observed 3-way array and X_2, X_3 are 2-way arrays, while Z_α for $\alpha = 1 \dots 5$ are the latent 2-way arrays. The 2-way arrays are just matrices but this can be easily extended to objects having arbitrary number of indices. Here, Z_2 is a shared factor, coupling all models. As the first model is a CP while the second and the third ones are MF's, we call the combined factorization as CP/MF/MF model. Such models are of interest when one can obtain different views of the same piece of information (here Z_2) under different experimental conditions. Singh and Gordon [80] focused on a similar problem called as *collective matrix factorization* (CMF) or *multi-matrix factorization*, for relational learning but only for matrix factors and observations. For *coupled matrix and tensor factorization* (CMTF), recently [43] proposed a gradient-based all-at-once optimization method as an alternative to *alternating least square* (ALS) optimization and demonstrated their approach for a CP/MF coupled model. Similar models are used for protein-protein interactions (PPI) problems in gene regulation [84].

The outline of this chapter is as follows. In Section 7.2 we derive update equations for generalized coupled tensor factorization (GCTF) that handle the simultaneous TF where multiple observation tensors are available. In Section 7.3 we discuss modelling

each observation by a different cost function. Section 7.4 is about solution of our motivating problem and here finally we illustrate our approach on a musical audio restoration problem.

7.2. Coupled Tensor Factorization

In this section we address the problem when multiple observed tensors X_ν for $\nu = 1 \dots |\nu|$ are factorized simultaneously. Each observed tensor X_ν now has a corresponding index set $\mathcal{V}_{0,\nu}$ and a particular configuration will be denoted by $v_{0,\nu} \equiv u_\nu$

We define a $|\nu| \times |\alpha|$ *coupling matrix* R where

$$R^{\nu,\alpha} = \begin{cases} 1 & X_\nu \text{ and } Z_\alpha \text{ connected} \\ 0 & \text{otherwise} \end{cases} \qquad \hat{X}_\nu(u_\nu) = \sum_{\bar{u}_\nu} \prod_{\alpha} Z_\alpha(v_\alpha)^{R^{\nu,\alpha}} \tag{7.2}$$

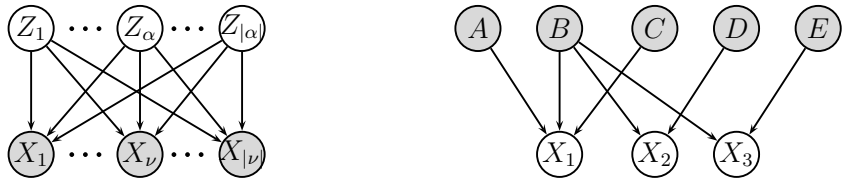


Figure 7.1. (Left) Coupled factorization structure where the arrow indicates the existence of the influence of latent tensor Z_α onto the observed tensor X_ν. (Right) The CP/MF/MF coupled factorization problem given in Equation 7.1.

For the coupled factorization, we get the following expression as the derivative of the log-likelihood by extending that of single observation case in Equation 6.7

$$\frac{\partial \mathcal{L}}{\partial Z_\alpha(v_\alpha)} = \sum_{\nu} R^{\nu,\alpha} \sum_{u_\nu \cap \bar{v}_\alpha} \Big(X_\nu(u_\nu) - \hat{X}_\nu(u_\nu) \Big) \varphi_\nu W_\nu(u_\nu) g_{\hat{X}}(\hat{X}(v_0)) \frac{\partial \hat{X}_\nu(u_\nu)}{\partial Z_\alpha(v_\alpha)} \tag{7.3}$$

where $W_\nu(u_\nu) \equiv W(\hat{X}_\nu(u_\nu))$ is the inverse variance function and φ_ν is the observation specific inverse dispersion parameter which is common for a given observation tensor. One remark is that as in the previous chapter we use the identity link function implying

that our models are linear, i.e. the estimations $\hat{X}$ and the parameters Z_α are linearly related, and hence we take

$$g_{\hat{X}}(\hat{X}(v_0)) = \frac{\partial g(\hat{X}(v_0))}{\partial \hat{X}(v_0)} = 1 \tag{7.4}$$

Then proceeding as in Section 6.3.1 (i.e. getting the Hessian and finding Fisher Information) we arrive at the update rule in vector form. That is, starting with the derivative $\frac{\partial \hat{X}_\nu(u_\nu)}{\partial Z_\alpha(v_\alpha)}$ as

$$\frac{\partial \hat{X}_\nu(u_\nu)}{\partial Z_\alpha(v_\alpha)} = \frac{\partial}{\partial Z_\alpha(v_\alpha)} \sum_{\bar{u}_\nu} \prod_{\alpha} Z_\alpha(v_\alpha)^{R^{\nu,\alpha}} = \sum_{\bar{u}_\nu \cap \bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'})^{R^{\nu,\alpha'}} \tag{7.5}$$

Then, the Fisher Information $-\Big\langle \frac{\partial^2 \mathcal{L}}{\partial Z_\alpha(v_\alpha)^2} \Big\rangle$ is

$$\begin{aligned} &= -\Big\langle \sum_\nu R^{\nu,\alpha} \sum_{v_0/v_\alpha} \varphi_\nu W_\nu(\hat{X}_\nu(u_\nu)) \frac{\partial \hat{X}_\nu(u_\nu)}{\partial Z_\alpha(v_\alpha)} \frac{\partial}{\partial Z_\alpha(v_\alpha)} \Big(X_\nu(u_\nu) - \hat{X}_\nu(u_\nu) \Big) \Big\rangle \\ &= \sum_\nu R^{\nu,\alpha} \sum_{v_0/v_\alpha} \varphi_\nu W_\nu(\hat{X}_\nu(u_\nu)) \Big\{ \frac{\partial \hat{X}_\nu(u_\nu)}{\partial Z_\alpha(v_\alpha)} \Big\}^2 \end{aligned} \tag{7.6}$$

since $\Big\langle X_\nu(u_\nu) - \hat{X}_\nu(u_\nu) \Big\rangle = 0$. Then the update in scalar form is

$$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha) + \frac{\sum_\nu R^{\nu,\alpha} \sum_{v_0/v_\alpha} \Big(X_\nu(u_\nu) - \hat{X}_\nu(u_\nu) \Big) \varphi_\nu W_\nu(\hat{X}_\nu(u_\nu)) \frac{\partial \hat{X}_\nu(u_\nu)}{\partial Z_\alpha(v_\alpha)}}{\sum_\nu R^{\nu,\alpha} \sum_{v_0/v_\alpha} \varphi_\nu W_\nu(\hat{X}_\nu(u_\nu)) \Big\{ \frac{\partial \hat{X}_\nu(u_\nu)}{\partial Z_\alpha(v_\alpha)} \Big\}^2} \tag{7.7}$$

and in vector form as

$$\vec{Z}_\alpha \leftarrow \vec{Z}_\alpha + \Big(\sum_\nu R^{\nu,\alpha} \varphi_\nu \nabla_{\alpha,\nu}^\top D_\nu \nabla_{\alpha,\nu} \Big)^{-1} \Big(\sum_\nu R^{\nu,\alpha} \varphi_\nu \nabla_{\alpha,\nu}^\top D_\nu \big(\vec{X}_\nu - \hat{\vec{X}}_\nu \big) \Big) \tag{7.8}$$

where $\nabla_{\alpha,\nu} = \partial g(\hat{X}_\nu)/\partial Z_\alpha$ and D_ν is the diagonal matrices as $D_\nu = diag(\text{vec } W_\nu)$. The update equations for the coupled case are quite intuitive; we calculate the $\Delta_{\alpha,\nu}$

functions defined as

$$\Delta_{\alpha,\nu}^{\varepsilon}(Q) = \left[\sum_{u_\nu \cap \bar{v}_\alpha} Q(u_\nu) \Big(\sum_{\bar{v}_0 \cap \bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'})^{R^{\nu,\alpha'}} \Big)^{\varepsilon} \right] \tag{7.9}$$

for each submodel and add the results.

Here we make an important assumption to get rid of the inverse dispersion parameters φ_ν. We assume that each observation tensor is modelled by the same type of distribution having the same dispersion parameter. In other words, all X_ν have the same cost function. Then, the inverse dispersion parameters φ_ν become independent of the sum $\sum_\nu$ and the φ_ν in the numerators and denominator cancel each other and disappear.

Lemma 7.1. *Update for non-negative GCTF*

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}(W_\nu \circ X_\nu)}{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}\Big(W_\nu \circ \hat{X}_\nu\Big)} \tag{7.10}$$

As a special case for the distributions whose inverse variance functions are as $W_\nu = \hat{X}_\nu^{-p}$, the update is

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}\Big(\hat{X}_\nu^{-p} \circ X_\nu\Big)}{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}\Big(\hat{X}_\nu^{1-p}\Big)} \tag{7.11}$$

Proof. The proof here is very much similar to the ones in Chapter 6. To simplify the update equation we make use of the generalized form of Equation 6.55 that bounds the step size. That is, the following is the generalized version of the bound of the maximal eigenvalue in Equation 6.55

$$\lambda_{max}(\sum_k Q_k) \leq \max_j \Big\{ \sum_k \frac{Q_k \beta_j}{\beta_j} \Big\} = \max_j \frac{\big\{ \sum_k Q_k \big\} \beta_j}{\beta_j} \tag{7.12}$$

where for $k = 1$ it is reduced to the standard formula in Equation 6.55. Then, for the

setting $H_\nu = \nabla_{\alpha,\nu}^T D_\nu \nabla_{\alpha,\nu}$

$$\lambda_{max}(\sum_\nu H_\nu) \leq \max_{v_\alpha} \{ \sum_\nu \frac{H_\nu Z_\alpha}{Z_\alpha} \} \qquad \text{s.t. } Z_\alpha(v_\alpha) > 0 \tag{7.13}$$

$$Z_\alpha \leftarrow Z_\alpha + \frac{\sum_\nu R^{\nu,\alpha} L_\alpha^T \{ W_\nu \circ (X_\nu - \hat{X}_\nu) \}}{\frac{\sum_\nu R^{\nu,\alpha} L_\alpha^T \{ W_\nu \circ \hat{X}_\nu \}}{Z_\alpha}} \tag{7.14}$$

$$\leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha} L_\alpha^T \{ W_\nu \circ X_\nu \}}{\sum_\nu R^{\nu,\alpha} L_\alpha^T \{ W_\nu \circ \hat{X}_\nu \}} \tag{7.15}$$

Finally we use $\Delta_{\alpha,\nu}()$ function abstraction to handle tedious reshape operations. □

7.2.1. General Update for GCTF

Lemma 7.2. *General update equation for GCTF*

$$Z_\alpha \leftarrow Z_\alpha + \frac{2}{\lambda_{\alpha/0}} \frac{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu} \Big(W_\nu \circ (X_\nu - \hat{X}_\nu) \Big)}{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}^2 (W_\nu)} \tag{7.16}$$

For the special case of the Tweedie family we plug $W_\nu = \hat{X}_\nu^{-p}$ and get the related formula as

$$Z_\alpha \leftarrow Z_\alpha + \frac{2}{\lambda_{\alpha/0}} \frac{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu} \Big(\hat{X}_\nu^{-p} \circ (X_\nu - \hat{X}_\nu) \Big)}{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}^2 \Big(\hat{X}_\nu^{-p} \Big)} \tag{7.17}$$

One important question in this update equation is that while α being constant, if all the scalars for different observations ν, i.e. $\lambda_{\alpha/\nu}$ where $\nu = 1, 2, \ldots$, are equal? Indeed, all the scalars are equal for specific factor α under the assumption that observed indices of a factor, if ever appears, are also observed for all observations tensors. hence we simply write them as $\lambda_{\alpha/0}$ with 0 for all observation tensors.

Proof. Bounding the step size

$$Z_\alpha \leftarrow Z_\alpha + \frac{\sum_\nu R^{\nu,\alpha} L_\alpha^T \{W_\nu \circ (X_\nu - \hat{X}_\nu)\}}{\frac{1}{2}\sum_\nu R^{\nu,\alpha} L_\alpha^T (W_\nu \circ L_\alpha)\mathbf{1}} \tag{7.18}$$

then afterwards effectively replacing $\nabla_{\alpha,\nu}^T \nabla_{\alpha,\nu}\mathbf{1}$ by $\lambda_{\alpha/0}(\nabla_{\alpha,\nu}^T)^2$ the update equation becomes

$$Z_\alpha \leftarrow Z_\alpha + \frac{2}{\lambda_{\alpha/0}} \frac{\sum_\nu R^{\nu,\alpha} L_\alpha^T \{W_\nu \circ (X_\nu - \hat{X}_\nu)\}}{\sum_\nu R^{\nu,\alpha} L_\alpha^{T^2} W_\nu} \tag{7.19}$$

□

7.3. Mixed Costs Functions

We derived update equations under the assumption of the same cost function for all the observed tensors. This enabled us to cancel out the dispersion parameters from the update equations. Now, we relax this assumption and leave the inverse dispersion parameters in the update equations. Then, update for non-negative GCTF for specific cost function for each observation tensor becomes

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}(\varphi_\nu W_\nu \circ X_\nu)}{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}\left(\varphi_\nu W_\nu \circ \hat{X}_\nu\right)} \tag{7.20}$$

where the scalar φ_ν denotes inverse dispersion parameter. This value, on the other hand, needs to estimated for each of observation ν. For two of the cost function types this value can easily be identified. The KL cost is modeled by the Poisson distribution where the dispersion parameter is 1. The Euclidean cost for the linear model where $\hat{X}(v_0) = \sum_{\bar{v}_0} \Lambda(v)$ and $\Lambda(v) = \prod_\alpha Z_\alpha(v_\alpha)$, assuming unite variance for $\Lambda(v)$ makes the dispersion parameter of X to be cardinality of the sum, i.e. $|\bar{v}_0]$.

Finally, as a special case for the distributions whose inverse variance functions

are as $W_\nu = \hat{X}_\nu^{-p_\nu}$, the update is

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}\Big(\varphi_\nu \hat{X}_\nu^{-p_\nu} \circ X_\nu\Big)}{\sum_\nu R^{\nu,\alpha} \Delta_{\alpha,\nu}\Big(\varphi_\nu \hat{X}_\nu^{1-p_\nu}\Big)} \tag{7.21}$$

Here we skipped the update for real data, but it can be similarly modified by replacing W_ν as $\varphi_\nu W_\nu$.

7.4. Experiments

Here we want to solve the CTF problem introduced in Equation 7.1, which constitutes to a coupled CP/MF/MF problem

$$\hat{X}_1^{i,j,k} = \sum_r A^{i,r} B^{j,r} C^{k,r} \qquad \hat{X}_2^{j,p} = \sum_r B^{j,r} D^{p,r} \qquad \hat{X}_3^{j,q} = \sum_r B^{j,r} E^{q,r} \tag{7.22}$$

where we employ the symbols $A : E$ for the latent tensors instead of Z_α. This factorization problem has the following R matrix with $|\alpha| = 5$, $|\nu| = 3$

$$R = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad \text{with} \quad \begin{aligned} \hat{X}_1 &= \textstyle\sum A^1 B^1 C^1 D^0 E^0 \\ \hat{X}_2 &= \textstyle\sum A^0 B^1 C^0 D^1 E^0 \\ \hat{X}_3 &= \textstyle\sum A^0 B^1 C^0 D^0 E^1 \end{aligned} \tag{7.23}$$

We want to use the general update in Equation 7.19. This requires derivation of $\Delta^\varepsilon_{\alpha,\nu}()$ for $\nu = 1$ (CP) and $\nu = 2$ (MF) but not for $\nu = 3$ since that $\Delta_{\alpha,3}()$ has the same shape as $\Delta_{\alpha,2}()$. Here we show the computation for B, i.e. for Z_2, which is the common factor

$$\Delta^\varepsilon_{B,1}(Q) = \left[\sum_{ik} Q^{i,j,k}\Big(A^{i,r}C^{k,r}\Big)^\varepsilon\right] = Q_{(1)}(C^\varepsilon \odot A^\varepsilon) \tag{7.24}$$

$$\Delta^\varepsilon_{B,2}(Q) = \left[\sum_{p} Q^{j,p}\big(D^{p,r}\big)^\varepsilon\right] = QD^\varepsilon \tag{7.25}$$

with $Q_{(n)}$ being *mode*-n unfolding operation that turns a tensor into matrix form [13]. In addition, for $\nu = 1$ the required scalar value $\lambda_{B/0}$ is $|r|$ here since $v_B \cap \bar{v}_0 = \{j, r\} \cap \{r\} = \{r\}$ noting that value $\lambda_{B/0}$ is the same for $\nu = 2, 3$. Finally the update for the factor B becomes

$$B \leftarrow B + \frac{2}{|r|} \frac{Q_{1(2)}(C \odot A) + Q_2 D + Q_3 E}{\hat{X}_{1(2)}^{-p}(C^2 \odot A^2) + \hat{X}_2^{-p} D^2 + \hat{X}_3^{-p} E^2} \tag{7.26}$$

where $Q_i = (X_i - \hat{X}_i) \circ \hat{X}_i^{-p}$ for $i = 1, 2, 3$. All tensor powers are element-wise. Note also that for the first non-coupled factor A the iterative update is

$$A \leftarrow A + \frac{2}{|r|} \frac{Q_{1(1)}(C \odot B)}{\hat{X}_{1(1)}^{-p}(C^2 \odot B^2)} \tag{7.27}$$

where $Q_1 = (X_1 - \hat{X}_1) \circ \hat{X}_1^{-p}$.

The simulated data size for observables is set to $|i| = |j| = |k| = |p| = |q| = 30$ while the latent dimension $|r|$ is set to 5. The number of alternating steps, i.e. iteration count is set to 1000. Euclidean cost is used as error measurement so the data generator is set to be Gaussian, although the experiment produced similar results for KL cost with Poisson data. The results are shown in Figure 7.2. The figure compares the original, the initial (start up) and the final (estimate) factors for A, B, C, D, E. Only the first row, i.e. $Z_\alpha(1, 1 : 10)$ is plotted. Note that the CP factorization is unique up to permutation and scaling [13] while MF is not unique, but when coupled with CP it recovers the original data as shown in the figure. For visualization, to find the correct permutation, for each of Z_α the matching permutation between the original and estimate are found by solving a *orthogonal Procrustes problem* [9] which finds the permutation matrix P by minimizing $|Z_\alpha^o - Z_\alpha^f P|$ with $P^T P = I$ where Z_α^o is the original factor.

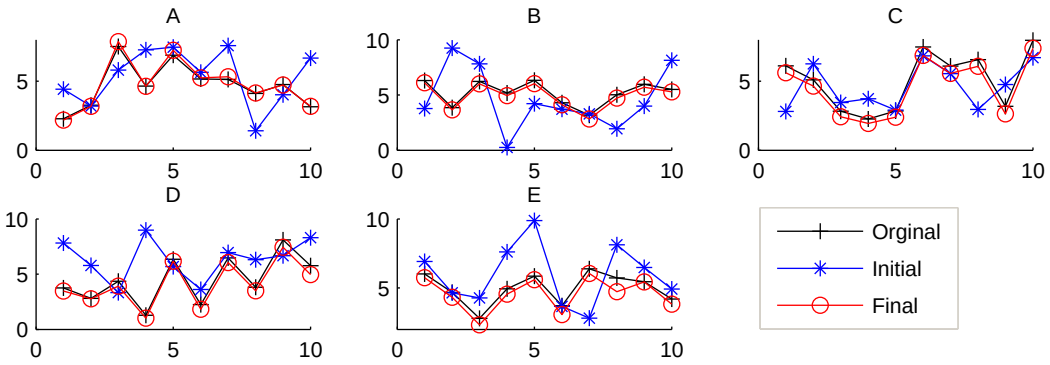


Figure 7.2. The figure compares the original, the initial (start up) and the final (estimate) factors for A, B, C, D, E.

7.4.1. Audio Experiments

In this section, we illustrate a real data application of our approach, where we reconstruct missing parts of an audio spectrogram $X(f,t)$, that represents the short time Fourier transform coefficient magnitude at frequency bin f and time frame t of a piano piece, see top left panel of Fig.7.4. This is a difficult matrix completion problem: as entire time frames (columns of X) are missing, low rank reconstruction techniques are likely to be ineffective. Yet such missing data patterns arise often in practice, e.g., when packets are dropped during digital communication. We will develop here a novel approach, expressed as a coupled tensor factorization model. In particular, the reconstruction will be aided by an approximate musical score, not necessarily belonging to the played piece, and spectra of isolated piano sounds. In this section we use the parenthesis notation $X(f,t)$ for a matrix scalar rather than the superscript notation $X^{f,t}$ that we use throughout this thesis since it is more familiar in the audio application literature.

Pioneering work of [11] has demonstrated that, when an audio spectrogram of music is decomposed using NMF as

$$X_1(f,t) \approx \hat{X}(f,t) = \sum_i D(f,i)E(i,t) \tag{7.28}$$

the computed factors D and E tend to be semantically meaningful and correlate well with the intuitive notion of spectral templates (harmonic profiles of musical notes) and a musical score (reminiscent of a piano roll representation such as a MIDI file). However, as time frames are modeled conditionally independently, it is impossible to reconstruct audio with this model when entire time frames are missing.

Restoration of the missing parts in audio spectrograms is a typical example where coupled factorization can be applied. Figure C.5 organizes various parts of the problem as a coupled factorization model. Here, the musical piece to be reconstructed shares B and D in common.

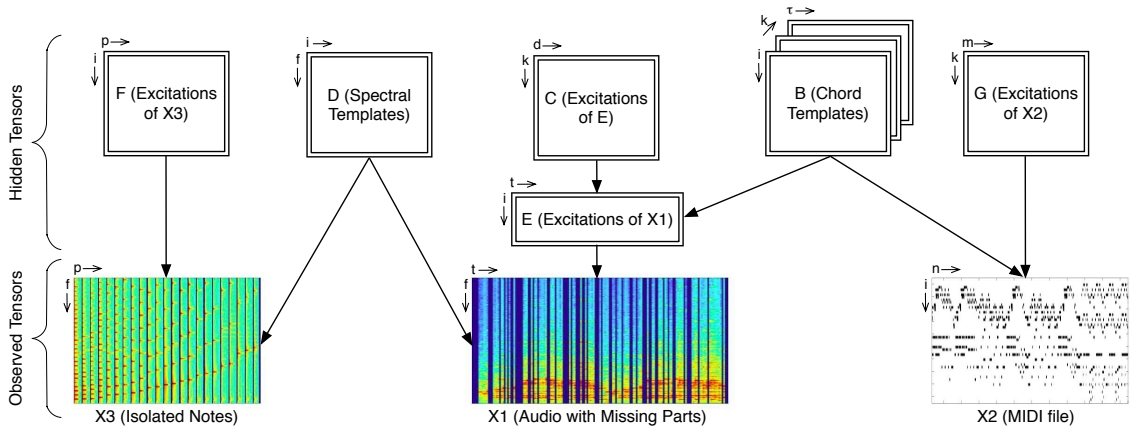


Figure 7.3. Restoration of the missing parts in audio spectrograms problem is organized as coupled tensor factorization.

In order to restore the missing parts in the audio, we form a model that can incorporate musical information of chords structures and how they evolve in time. In order to achieve this, we hierarchically decompose the excitation matrix E as a convolution of some basis matrices and their weights as

$$E(i,t) = \sum_{k,\tau} B(i,\tau,k)C(k,t-\tau) \tag{7.29}$$

Here the basis tensor B encapsulates both vertical and temporal information of the notes that are likely to be used in a musical piece; the musical piece to be reconstructed

will share B, possibly played at different times or tempi as modelled by G. After replacing E with the decomposed version, we get the following model (eq 7.30):

$$\hat{X}_1(f,t) = \sum_{i,\tau,k,d} D(f,i)B(i,\tau,k)C(k,d)Z(d,t,\tau) \qquad \text{Test file} \qquad (7.30)$$

$$\hat{X}_2(i,n) = \sum_{\tau,k,m} B(i,\tau,k)G(k,m)Y(m,n,\tau) \qquad \text{MIDI file} \qquad (7.31)$$

$$\hat{X}_3(f,p) = \sum_{i} D(f,i)F(i,p)T(i,p) \qquad \text{Merged training files} \qquad (7.32)$$

Here we have introduced new dummy indices d and m, and new (fixed) factors

$$Z(d,t,\tau) = \delta(d-t+\tau) \qquad (7.33)$$

$$Y(m,n,\tau) = \delta(m-n+\tau) \qquad (7.34)$$

to express this model in our framework. In eq 7.32, while forming X_3 we concatenate isolated recordings corresponding to different notes. Besides, T is a $0-1$ matrix, where $T(i,p) = 1(0)$ if the note i is played (not played) during the time frame p and F models the time varying amplitudes of the training data. R matrix for this model is defined as follows:

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad \text{with} \quad \begin{aligned} \hat{X}_1 &= \textstyle\sum D^1 B^1 C^1 Z^1 G^0 Y^0 F^0 T^0 \\ \hat{X}_2 &= \textstyle\sum D^0 B^1 C^0 Z^0 G^1 Y^1 F^0 T^0 \\ \hat{X}_3 &= \textstyle\sum D^1 B^0 C^0 Z^0 G^0 Y^0 F^1 T^1 \end{aligned} \qquad (7.35)$$

Figure 7.4 illustrates the performance the model, using KL cost ($W = \hat{X}^{-1}$) on a 30 second piano recording where the 70% of the data is missing; we get about 5dB SNR improvement, gracefully degrading from 10% to 80% missing data: the results are encouraging as quite long portions of audio are missing, see bottom right panel of Figure 7.4. In the figure top row, left to right: Observed matrices X_1: spectrum of the piano performance, darker colors imply higher magnitude (missing data (70%) are shown white), X_2, a piano roll obtained from a musical score of the piece, X_3, spectra

of 88 isolated notes from a piano. Bottom Row: Reconstructed X_1, the ground truth, and the SNR results with increasing missing data. Here, initial SNR is computed by substituting 0 as missing values.

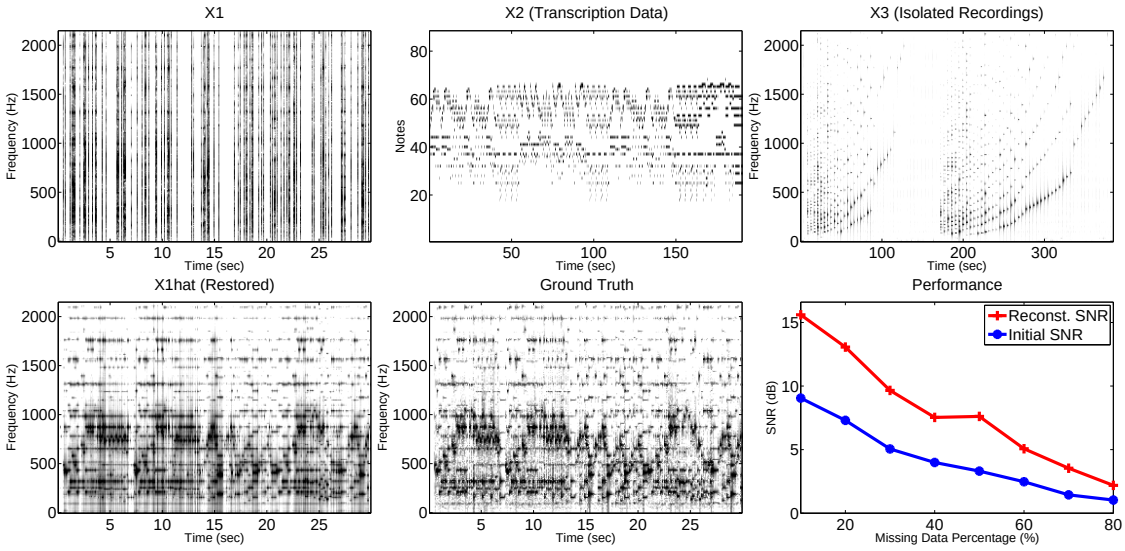


Figure 7.4. Top row, left to right: Observed matrices X_1: spectrum of the piano performance, darker colors imply higher magnitude, X_2, a piano roll obtained from a musical score of the piece, X_3, spectra of 88 isolated notes from a piano. Bottom Row: Reconstructed X_1, the ground truth, and the SNR results.

7.5. Summary

This chapter has provided a general solution for the computation of arbitrary coupled tensor factorizations. Here we introduce a new variable ν to index multiple observed tensors as X_ν for $\nu = 1 \dots |\nu|$ that are factorized simultaneously. We then define a $|\nu| \times |\alpha|$ *coupling matrix* R to encode the influence of the latent factor Z_α on the observed tensor X_ν such that

$$R^{\nu,\alpha} = \begin{cases} 1 & X_\nu \text{ and } Z_\alpha \text{ connected} \\ 0 & \text{otherwise} \end{cases} \qquad \hat{X}_\nu(u_\nu) = \sum_{\bar{u}_\nu} \prod_{\alpha} Z_\alpha(v_\alpha)^{R^{\nu,\alpha}} \tag{7.36}$$

Then the non-negative parameter update for CTF becomes

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha}\Delta_{\alpha,\nu}(W_\nu \circ X_\nu)}{\sum_\nu R^{\nu,\alpha}\Delta_{\alpha,\nu}\Big(W_\nu \circ \hat{X}_\nu\Big)} \tag{7.37}$$

and for general (real numbers) parameter update for CTF

$$Z_\alpha \leftarrow Z_\alpha + \frac{2}{\lambda_{\alpha/0}} \frac{\sum_\nu R^{\nu,\alpha}\Delta_{\alpha,\nu}\Big(W_\nu \circ \big(X_\nu - \hat{X}_\nu\big)\Big)}{\sum_\nu R^{\nu,\alpha}\Delta^2_{\alpha,\nu}(W_\nu)} \tag{7.38}$$

One note here is that, by assuming the same cost function for all the observations X_ν we could cancel out the dispersion parameters φ_ν from the update equations. Yet by leaving them as they are, i.e. replacing the inverse variance functions W_ν with $\varphi_\nu W_\nu$, we obtain more general update equations as

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_\nu R^{\nu,\alpha}\Delta_{\alpha,\nu}(\varphi_\nu W_\nu \circ X_\nu)}{\sum_\nu R^{\nu,\alpha}\Delta_{\alpha,\nu}\Big(\varphi_\nu W_\nu \circ \hat{X}_\nu\Big)} \tag{7.39}$$

This relaxes our assumption and allows use of the Gaussian, for example, for one observation while the Poisson for another observation.

7.6. Implementation

Here we present the implementations for CP/MF/MF problem described in the introduction. First implementation is for non-negative update for CTF, i.e. Multiplicative Update Rule (MUR) implementation, while the second is for general update for CTF.

Input: X, Z, MAXITER, N, M, p

N: number of latent tensor, here N=5

M: number of observation tensor, here M=3

p : distribution index, here p=0, i.e. Euclidean cost

Output: Z

for $e = 1$ to MAXITER **do**

  for $a = 1$ to 5 **do**

    $\hat{X}_1 = reshape(Z_1 * (Z_3 \odot Z_2)^T, i, j, k);$

    $\hat{X}_2 = Z_2 * Z_4^T;$

    $\hat{X}_3 = Z_2 * Z_5^T;$

    for $m = 1$ to M **do**

      $Q1_m = X_m \circ (\hat{X}_m^{-p});$

      $Q2_m = \hat{X}_m^{1-p};$

    end for

    if $(a == 1)$ **then**

      $UP = nshape(Q1_1, a) * (Z_3 \odot Z_2);$

      $DW = nshape(Q2_1, a) * (Z_3 \odot Z_2);$

    else if $(a == 2)$ **then**

      $UP = nshape(Q1_1, a) * (Z_3 \odot Z_1);$

      $DW = nshape(Q2_1, a) * (Z_3 \odot Z_1);$

      $UP = UP + Q1_2 * Z_4;$

      $DW = DW + Q2_2 * Z_4;$

      $UP = UP + Q1_3 * Z_5;$

      $DW = DW + Q2_3 * Z_5;$

    end if

    $Z_a = Z_a \circ (UP./DW);$

  end for

end for

Figure 7.5. Matlab-like implementation for the CP/MF/MF coupled problem in MUR algorithm. Khatri-Rao product and mode-n unfolding are implemented in N-way Toolbox [73] as $krb()$ and $nshape()$ functions respectively.

input: X, Z, MAXITER, N, M, p, LSIZ

N: number of latent tensor, here N=5

M: number of observation tensor, here M=3

p : distribution index, here p=0, i.e. Euclidean cost

LSIZ : Latent size, here LSIZ=5

Output: Z

for $e = 1$ to MAXITER **do**

  for $a = 1$ to N **do**

    $\hat{X}_1 = reshape(Z_1 * (Z_3 \odot Z_2)^T, i, j, k);$

    $\hat{X}_2 = Z_2 * Z_4^T;$

    $\hat{X}_3 = Z_2 * Z_5^T;$

    for $m = 1$ to M **do**

      $Q1_m = (X_m - \hat{X}_m) \circ (\hat{X}_m^{-p});$

      $Q2_m = (\hat{X}_m^{-p});$

    end for

    $C_a = Z_a^2;$

    if $(a == 1)$ **then**

      $UP = nshape(Q1_1, a) * (Z_3 \odot Z_2);$

      $DW = nshape(Q2_1, a) * (C_3 \odot C_2) * LSIZ/2;$

    else if $(a == 2)$ **then**

      $UP = nshape(Q1_1, a) * (Z_3 \odot Z_1);$

      $DW = nshape(Q2_1, a) * (C_3 \odot C_1) * LSIZ/2;$

      $UP = UP + Q1_2 * Z_4;$

      $DW = DW + Q2_2 * C_4 * LSIZ/2;$

      $UP = UP + Q1_3 * Z_5;$

      $DW = DW + Q2_3 * C_5 * LSIZ/2;$

    end if

  end for

  $Z_a = Z_a + (UP./DW);$

end for

Figure 7.6. Matlab-like implementation for the CP/MF/MF coupled problem in general update algorithm. Khatri-Rao product and mode-n unfolding are implemented in N-way Toolbox [73] as $krb()$ and $nshape()$ functions respectively.

8. CONCLUSION

In this thesis we developed a probabilistic framework for multiway analysis of high dimensional datasets. By exploiting the link between graphical models and tensor factorization models we cast any arbitrary tensor factorization problem, and many popular models such as CP or TUCKER3 as inference, where tensor factorization reduces to a parameter estimation problem.

Our algorithms are general in the sense that we can compute arbitrary factorizations in a message passing framework on a graph where vertices correspond to indices and cliques represent factors of the tensor factorization, derived for a broad class of exponential family distributions including special cases such as Tweedie's distributions corresponding to β-divergences.

We first illustrated our approach for the conditionally Poisson case and employed the EM algorithm for the optimization. We also extend the probability model to include the conjugate priors and obtain the update equations accordingly. One main saving in our framework appears in the computation of $\Delta_\alpha()$, that is computationally equivalent to computing expectations under probability distributions that factorise according to a given graph structure. As is the case with graphical models, this quantity can be computed via message passing: algebraically we distribute the summation over all $\bar{v}_\alpha$ and compute the sum in stages.

Consequently, by use of the exponential dispersion models and as a special case by the use of power variance functions where the variance functions is in the simple form as $v(\mu) = \mu^p$, we were able to derive the generic form of update equations for any β-divergence. The gradient ascent interpretation of the EM and dual formulation of likelihood maximization as divergence minimization are used to obtain the MUR and the ALS updates for Gaussian case. Conjugate priors are also introduced. In addition, the framework handles the missing data naturally.

We also describe a model selection framework for nonnegative tensor factorization with KL cost from a probabilistic perspective. The model selection includes both the task of model order determination and model structure learning. The model comparison is based on variational Bayes. Our main contribution is that our approach is not limited to a specific factorization structure and is able to do inference efficiently in any tensor factorization model.

We also establish a link between GLMs and tensor factorizations and provide a general solution for the computation of arbitrary coupled tensor factorizations where multiple observation tensors are factorized simultaneously. A powerful aspect of the GCTF framework is assigning different cost functions, i.e. distributions, to different observation tensors in a coupled factorization model. We illustrated our approach on a musical audio restoration problem. We believe that, as a whole, the GCTF framework covers a broad range of models that can be useful in many different application areas beyond audio processing, such as network analysis, bio-informatics or collaborative filtering.

Perhaps the most important contribution of the paper is the generalization of TF problem to a point that allows one to invent new factorization models appropriate to their applications. Pedagogically, the framework guides building new models as well as deriving update equations for β-divergence that unifies the popular cost functions. Indeed, results scattered in the literature can be derived in a straightforward manner.

8.1. Future Work

- *Non-linear tensor factorization models.* Tensor factorization models are linear models in the sense that the model estimate $\hat{X}$ and the parameters Z_α are linearly related. Just in the case of GLM we may use a non-identity link function to build non-linear tensor factorization models.

$$g\left(\hat{X}^{i,j,k}\right) = \sum_r A^{i,r} B^{j,r} C^{k,r}$$

- *Tensor factorization via MCMC inference.* Markov Chain Monte Carlo is also a viable alternative for inference [35].
- *Generalization of the model selection for arbitrary cost functions.* The main limitation of our model selection approach is that we use only the KL cost. Although we sketch the outline for the general cost functions, we did not extract the expectation of the posterior distribution for each cost function.
- *Conjugate priors for coupled tensor factorization.* For PLTF models we extended the framework to include prior knowledge and obtain fixed point update equations that include hyperparameters. For PLTF framework that we build in Chapter 4 we obtain both ML and MAP optimization. For GTF and GCTF developed in Chapter 6 and 7, the frameworks consider only likelihood maximization and does not take into consideration the prior belief. The frameworks GTF and GCTF can be extended for also MAP optimization to include hyperparameters in the update equations. Recall that the conjugate priors for the EDM family are given as

$$p(s|\theta,\varphi_s) = g(s,\varphi_s)\exp\varphi_s\left(s^T\theta - \psi(\theta)\right) \tag{8.1}$$

$$p(\theta|n_0,s_0) = h(n_0,s_0)\exp\{n_0\left(s_0\theta - \psi(\theta)\right)\} \tag{8.2}$$

  with n_0, s_0 being the hyperparameters corresponding to belief about a prior sample size and prior expectation of the mean parameter [68]. Hence one can start using product of the two equations to compute the likelihood and come up with MAP optimized version of GTF and GCTF.
- *Arbitrary latent structure.* In this thesis we generalized probabilistic tensor factorization in two perspectives; generalization for major class of cost functions and generalization for arbitrary latent structure. To illustrate the second generalization perspective we show the results mainly for CP and TUCKER3. However, it would be interesting and useful to experiment the results on an arbitrary (possibly a real-world application) latent structure.
- *Graphical models foundation and message passing.* This thesis is inspired from the close link between tensor factorization and probabilistic graphical models.

Even further, the abstraction Δ function that we use in the update equations is identical to the message passing of graphical models that consist of contraction and marginalization operations. In addition to this our matricization procedure is a form of junction tree. However, yet a theoretical foundation, i.e. deriving tensor factorization from the graphical model theory, is incomplete. In addition to this, Δ function is not fully optimized for performance perspective where one could use factor graphs rather than DAG.

- *Performance of tensor factorization.* There is certainly a need for a specification for tensor factorization performance benchmarks assuming there is already none. The specification should specify datasets, factorization models, hardware capability such as maximum memory, software capability such as sequential or parallel processing, classification of algorithms such as gradient base iterative algorithms or alternating but direct algorithm ALS. The interest here is practical rather than theoretical, and one consequence of such a work can answer a query like what the fastest way to factorize $10^3 \times 10^3 \times 10^3$ cube for CP or TUCKER3 is. Clearly one should consider memory consumption and may load data partially from the disk.
- *Constraints on optimization.* One very important point; constraints on optimization methods for TF make the factorization the most useful. The domain of the latent factors is almost unlimited, while the constraints such as non-negativity, uniqueness, orthogonality and sparsity re-shape the factors for the desired property. In this work we could not make room for a technical discussion and development on constraints except that the non-negativity is already handled by using KL cost due to the non-negative nature of the Poisson distribution. In addition, columns (or fibers) in factors can always be orthogonalized by Gram-Schmidt method.

APPENDIX A: PLTF FOR GAUSSIAN GENERATIVE MODELS

In Chapter 4 we, first, developed fixed point update equation for PLTF models for the KL cost. Then, we generalized the development for the unified case where we obtain update equations for EU, KL, IS costs in a singe equations. Here we obtain update equations for the EU as modelling unknown mean, known variance of the Gaussian and for the IS cost as known mean, unknown variance of the Gaussian. We skip the conjugate prior analysis for EU and IS cost that we did for the KL cost previously but the procedure is similar. Also notationally we denote the PLTF for EU and IS costs as $PLTF_{EU}$ and $PLTF_{IS}$.

A.1. Gaussian Modelling of the EU Cost

For Euclidean $PLTF$, we write the following generative model for the element of the latent tensor S

$$S(v) \sim \mathcal{N}(S; \Lambda(v), 1) \tag{A.1}$$

Note that due to reproductivity property of the Gaussian distributions [71] the observation $X(v_0) = \sum_{\bar{v}_0} S(v)$ has the same type of distribution as $S(v)$. The variance here is constant and is equal to 1 but taking it as any constant C does not make any difference since it will canceled to 1 in the posterior expectation equation $\langle S(v)|X(v_0)\rangle$. The derivation of PLTF_{EU} fixed point update equation follows closely Chapter 4 where we merely replace the Poisson likelihood with that of a Gaussian. The complete data log-likelihood becomes

$$\mathcal{L}_{EU} \;=\; \sum_{v} M(v_0) \left(-\frac{1}{2}\log(2\pi) - \frac{1}{2}\left(S(v) - \Lambda(v)\right)^2 \right) \tag{A.2}$$

subject to the constraint $X(v_0) = \sum_{\bar{v}_0} S(v)$ for $M(v_0) = 1$. The sufficient statistics of the Gaussian posterior $p(S|Z, X)$ are available in closed form as

$$\langle S(v)|X(v_0)\rangle = \Lambda(v) - \frac{1}{K}(X(v_0) - \hat{X}(v_0)) \tag{A.3}$$

where K is the cardinality of unobserved configurations, i.e. the number of all possible configurations of $\bar{v}_0$ and hence $K = |\bar{v}_0|$. Then, the solution of the M step after plugging Equation A.3 in $\frac{\partial \mathcal{L}_{EU}}{\partial Z_\alpha(v_\alpha)}$ and by setting it to zero

$$\begin{aligned} \frac{\partial \mathcal{L}_{EU}}{\partial Z_\alpha(v_\alpha)} &= \sum_{\bar{v}_\alpha} M(v_0) \left(\left(X(v_0) - \hat{X}(v_0) \right) \partial_\alpha \Lambda(v) \right) \\ &= \Delta_\alpha(M \circ X) - \Delta_\alpha(M \circ \hat{X}) = 0 \end{aligned} \tag{A.4}$$

The solution of this fixed point equation leads to two related but different iterative schemata: multiplicative updates (MUR) and alternating least squares (ALS).

- *$PLTF_{EU}$ Multiplicative Update Rules (MUR).* This method is indeed gradient ascent similar to [5] by setting

  $$\eta(v_\alpha) = Z_\alpha(v_\alpha)/\Delta_\alpha(M \circ \hat{X})(v_\alpha) \tag{A.5}$$

  in the following

  $$Z_\alpha(v_\alpha) \leftarrow Z_\alpha(v_\alpha) + \eta(v_\alpha) \frac{\partial \mathcal{L}_{EU}}{\partial Z_\alpha(v_\alpha)} \tag{A.6}$$

  Then the update rule becomes simply

  $$Z_\alpha \leftarrow Z_\alpha \circ \Delta_\alpha(M \circ X)/\Delta_\alpha(M \circ \hat{X}) \tag{A.7}$$

- *$PLTF_{EU}$ Alternating Least Squares (ALS).* The idea behind ALS is to obtain a closed form solution for Z_α directly from Equation A.4

$$\Delta_\alpha(M \circ X) = \Delta_\alpha(M \circ \hat{X}) \tag{A.8}$$

Note that $\hat{X}$ depends on Z_α, see Equation 4.2. This equation can be solved for Z_α by least squares, as it is linear in Z_α. If there is no missing data ($M(v_0) = 1$ for all v_0), the result is available in closed form. To see this, we write all the tensors in matrix form and write the solution explicitly using standard matrix algebra.

A.2. Gaussian Modelling of the IS Cost

IS cost can also be modeled as zero mean unknown variance of a Gaussian [31] as $S(v) \sim \mathcal{N}(S(v); 0, \Lambda(v))$, $X(v_0) \sim \mathcal{N}(X(v_0); 0, \hat{X}(v_0))$, where $X(v_0) = \sum_{\bar{v}_0} S(v)$. Taking the derivative of the log likelihood of the Gaussian w.r.t. variance parameter

$$\frac{\partial \log p(S(v); 0, \Lambda(v))}{\partial_\alpha \Lambda(v)} = -\frac{1}{2\Lambda(v)} + \frac{1}{2\Lambda(v)^2}(S(v) - 0)^2 \tag{A.9}$$

and after applying the chain rule the optimization problem turns to solution of the following

$$\frac{\partial \mathcal{L}}{\partial Z_\alpha(v_\alpha)} = \sum_{v_0/v_\alpha} \left[\left(-\frac{1}{2\Lambda(v)} + \frac{1}{2\Lambda(v)^2} \langle S(v)^2 | X(v_0) \rangle \right) \partial_\alpha \Lambda(v) \right] = 0 \tag{A.10}$$

$$= \sum_{v_0/v_\alpha} \left(\frac{\langle S(v)^2 | X(v_0) \rangle}{Z_\alpha(v_\alpha) \partial_\alpha \Lambda(v)} - 1 \right) = 0 \tag{A.11}$$

where we solve for $Z_\alpha(v_\alpha)$ as

$$Z_\alpha(v_\alpha) = \frac{\sum_{v_0/v_\alpha} \frac{\langle S(v)^2 | X(v_0) \rangle}{\partial_\alpha \Lambda(v)}}{\sum_{v_0/v_\alpha} 1} \tag{A.12}$$

Computation of $\langle S(v)^2|X(v_0)\rangle$ requires posterior mean and variance [1] as

$$\langle S(v)|X(v_0)\rangle = \frac{X(v_0)}{\hat{X}(v_0)}\Lambda(v) \qquad Var\left(S(v)|X(v_0)\right) = \Lambda(v) - \frac{\Lambda(v)^2}{\hat{X}(v_0)} \tag{A.13}$$

$$\left\langle S(v)^2|X(v_0)\right\rangle = \langle S(v)|X(v_0)\rangle^2 + Var\left(S(v)|X(v_0)\right) \tag{A.14}$$

$$= \frac{X(v_0)^2}{\hat{X}(v_0)^2}\Lambda(v)^2 + \Lambda(v) - \frac{\Lambda(v)^2}{\hat{X}(v_0)} \tag{A.15}$$

where this result is given in [19, 31].

APPENDIX B: EXPONENTIAL FAMILY DISTRIBUTIONS

B.1. Exponential Family Distributions

In this section we list and derive main properties of the exponential family distributions that we referred in this thesis. The derivations for the Gaussian distribution has already been made in Chapter 2.

B.1.1. Gamma Distributions

In the following we find sufficient statistics, canonical parameters, cumulant function and entropy for the gamma distribution. First we write the density:

$$p(x; a, b) = \exp\left((a-1)\log x - bx + a\log b - \log\Gamma(a)\right) \tag{B.1}$$

Sufficient Statistics are easily identify as the functions x that go into multiplication with the canonical parameters

$$\begin{bmatrix} t_1(x) \\ t_2(x) \end{bmatrix} = \begin{bmatrix} x \\ \log x \end{bmatrix} \tag{B.2}$$

Canonical Parameters are identified as the multipliers of the sufficient statistics coming from the definition

$$\begin{bmatrix} t_1(x) \\ t_2(x) \end{bmatrix}^T \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x \\ \log x \end{bmatrix}^T \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x \\ \log x \end{bmatrix}^T \begin{bmatrix} -b \\ a-1 \end{bmatrix} \tag{B.3}$$

Thus

$$\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} = \begin{bmatrix}-b\\ a-1\end{bmatrix} \qquad \text{or} \qquad \begin{bmatrix}b\\ a\end{bmatrix} = \begin{bmatrix}-\theta_1\\ \theta_2+1\end{bmatrix} \tag{B.4}$$

Cumulant Function is computed as

$$p(x;\mu,\theta) = \exp\left(\begin{bmatrix}x\\ \log x\end{bmatrix}^T \begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} + a\log b - \log\Gamma(a)\right) \tag{B.5}$$

$$= \exp\left(\begin{bmatrix}x\\ \log x\end{bmatrix}^T \begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} + (\theta_2+1)\log(-\theta_1) - \log\Gamma(\theta_2+1)\right) \tag{B.6}$$

where $\psi(\theta_1,\theta_2)$ is left as the remaining terms depending only on canonical parameters θ_1,θ_2 as

$$\psi(\theta_1,\theta_2) = -(\theta_2+1)\log(-\theta_1) + \log\Gamma(\theta_2+1) \tag{B.7}$$

h(x) is identified as the terms that do not contain any canonical parameters

$$h(x) = 1 \tag{B.8}$$

The *Entropy* is

$$\left\langle \begin{bmatrix}x\\ \log x\end{bmatrix} \right\rangle = \begin{bmatrix}\frac{a}{b}\\ \Psi(a) - \log b\end{bmatrix} \tag{B.9}$$

where $\Psi()$ is called as digamma function and it is $\Psi(a) = \frac{\partial \log\Gamma(a)}{\partial a}$. Then

$$H[X] = -\left(\begin{bmatrix}\frac{a}{b}\\ \Psi(a) - \log b\end{bmatrix}^T \begin{bmatrix}-b\\ a-1\end{bmatrix} + a\log b - \log\Gamma(a)\right) \tag{B.10}$$

$$= a + (1-a)\Psi(a) - \log b + \log\Gamma(a) \tag{B.11}$$

B.1.2. Poisson Distribution

In the following we find sufficient statistics, canonical parameters, cumulant function and entropy for the Poisson distribution. The density for the Poisson distribution

$$\mathcal{PO}(x;a) = \exp(-a + x\log a - \log x!) \tag{B.12}$$

The only *sufficient statistics* is

$$t(x) = x \tag{B.13}$$

Canonical Parameters

$$t(x)\theta = x\log a \qquad \Rightarrow \qquad \theta = \log a \qquad \Rightarrow \qquad a = \exp(\theta) \tag{B.14}$$

Cumulant Function

$$p(x;a) = \exp\left(x\theta - a\right)(x!)^{-1} \tag{B.15}$$

$$= \exp\left(x\theta - \exp(\theta)\right)(x!)^{-1} \tag{B.16}$$

where $\psi(\theta)$ is left as the remaining terms depending only on canonical parameters θ as

$$\psi(\theta) = \exp(\theta) \tag{B.17}$$

h(x) is identified as the terms that do not contain any canonical parameters

$$h(x) = (x!)^{-1} \tag{B.18}$$

Entropy . Note that the expected sufficient statistics is

$$\langle x\rangle = a \tag{B.19}$$

Then

$$H[X] = -\left(a \log a - a - \langle \log x! \rangle\right) \tag{B.20}$$

$$= a(1 - \log a) + \sum_{k}^{\infty} \exp(-a + k \log a - \log k!) \log(k!) \tag{B.21}$$

note that how we replace x by a in the $\langle \log x! \rangle$.

B.1.3. Relation of the Poisson and Multinomial distributions

Let $X_i \quad i = 1 \ldots k$ k independent random variables having Poisson distribution with parameters $\lambda_i \quad i = 1 \ldots k$. Let N be a random variable obtained as sum of the variables $N = \sum_i^k X_i$. Then the joint probability distribution conditioned on $N = n$ has the multinomial distribution. That is,

$$p(X_1 = x_1, \ldots, X_k = x_k | N = n) = \frac{n!}{x_1! x_2! \ldots x_k!} \lambda_1^{x_1} \lambda_2^{x_2} \ldots \lambda_k^{x_k} \tag{B.22}$$

Proof. Note that given $x_1, \ldots, x_k$, N has no longer uncertainty, and hence dropped out

$$p(x_1, \ldots, x_k | n) = \frac{p(x_1, \ldots, x_k, n)}{p(n)} = \frac{p(x_1, \ldots, x_k)}{p(n)} = \frac{\prod_i^k p(x_i)}{p(n)} \tag{B.23}$$

Since all the variables have the Poisson distribution

$$= \frac{\prod_i^k \exp(-\lambda_i - \log x_i! + x_i \log \lambda_i)}{\exp(-\sum_i^k \lambda_i - \log n! + n \log \sum_i^k \lambda_i)} \tag{B.24}$$

$$= \frac{\exp \sum_i^k (-\lambda_i - \log x_i! + x_i \log \lambda_i)}{\exp(-\sum_i^k \lambda_i - \log n! + n \log \sum_i^k \lambda_i)} \tag{B.25}$$

$$= \exp\left(\sum_i^k -\lambda_i - \sum_i^k \log x_i! + \sum_i^k x_i \log \lambda_i + \sum_i^k \lambda_i + \log n! - n \log \sum_i^k \lambda_i \right) \tag{B.26}$$

$$= \exp\left(\log n! - \sum_i^k \log x_i! + \sum_i^k x_i \log \lambda_i - n \log \sum_i^k \lambda_i \right) \tag{B.27}$$

Finally, note that $n = \sum_i^k x_i$ and $p_i = \frac{\lambda_i}{\sum_i^k \lambda_i}$ we come up with the definition of the

multinomial distribution

$$= \exp\left(\log n! - \sum_i^k \log x_i! + \sum_i^k x_i \log \lambda_i - \sum_i^k x_i \log \sum_i^k \lambda_i\right) \tag{B.28}$$

$$= \exp\left(\log n! - \sum_i^k \log x_i! + \sum_i^k x_i \log \frac{\lambda_i}{\sum_i^k \lambda_i}\right) \tag{B.29}$$

□

Table B.1. Characteristics of some common distributions from exponential family. In general a is mean (location) and b is shape parameter. For the gamma distribution we have an important equality $\langle \log x \rangle_{\mathcal{G}} = \psi(a) - \log b$ for convention 1, $\langle \log(x) \rangle_{\mathcal{G}} = \psi(a) + \log b$ for convention 2. For multinomial distribution $\sum_i^k x_i = n$, $x = [x_1, \ldots, x_k]$, $p = [p_1, \ldots, p_k]$, and $\sum_i^k p_i = 1$. For multivariate Gaussian distribution, X is m-dimensional row vector as $X = (X_1, X_2, \ldots, X_m)$, A is m-dimensional row vector and B is $m \times m$ covariance matrix.

Distribution	**Density**	**Mean**	**Variance**	**Support**
Gaussian	$\mathcal{N}(x; a, b) = \exp\left(-\frac{(x-a)^2}{2b} - \frac{1}{2}\log(2\pi b)\right)$	a	b	$x \in (-\infty, \infty)$
Poisson	$\mathcal{PO}(x; a) = \exp\left(-a + x \log a - \log x!\right)$	a	a	$x \in \{0, 1, 2, \ldots\}$
Gamma$_1$	$\mathcal{G}_1(x; a, b) = \exp\left((a-1)\log x - bx + a \log b - \log \Gamma(a)\right)$	$\frac{a}{b}$	$\frac{a}{b^2}$	$x \in (0, \infty)$
Gamma$_2$	$\mathcal{G}_2(x; a, b) = \exp\left((a-1)\log x - \frac{1}{b}x - a \log b - \log \Gamma(a)\right)$	ab	ab^2	$x \in (0, \infty)$
Inv. Gaussian	$\mathcal{IG}(x; a, b) = \exp\left(\frac{1}{2}\log\left(\frac{b}{2\pi x^3}\right) + \frac{-b(x-a)^2}{2a^2 x}\right)$	a	$\frac{a^3}{b}$	$x \in (0, \infty)$
Multinomial	$\mathcal{M}(x; n; p) = n! \prod_i^k \frac{p_i^{x_i}}{x_i!} = \exp\left(\log n! - \sum_i^k \log x_i! + \sum_i^k x_i \log p_i\right)$	np_i	$np_i(1-p_i)$	$x_i \in \{0, 1, \ldots, n\}$ $\sum_i x_i = n$
Multivariate Gaussian	$\mathcal{N}(X; A, B) = \exp\left(-\frac{1}{2}(X-A)B^{-1}(X-A)^T - \frac{1}{2}\log\left((2\pi)^m \lvert B\rvert\right)\right)$	A	B	$X \in (-\infty, \infty)$

B.1.4. KL divergence of Distributions

In this section we list the entropy and KL divergence (between two) of the Gaussian, the Poisson and the gamma distributions. We recall that the (Shannon) entropy and KL divergence are defined as

$$H[p] = -\sum_z p(z) \log p(z) \tag{B.30}$$

$$D_{KL}(q||p) = \sum_z q(z) \log \frac{q(z)}{p(z)} \tag{B.31}$$

Table B.2. Entropy of the Gaussian, the Poisson, the gamma and the multivariate Gaussian [38] distributions.

Distribution	**Notation**	**Entropy**
Gaussian	$p(x) = \mathcal{N}(x; a, b)$	$H[X] = \frac{1}{2}(1 + \log 2\pi + \log b)$
Poisson	$p(x) = \mathcal{PO}(x; a)$	$H[X] = a(1 - \log a) + \sum_k^\infty \exp(-a + k \log a - \log k!) \log k!$
Gamma	$p(x) = \mathcal{G}(x; a, b)$	$H[X] = a + (1-a)\Psi(a) - \log b + \log \Gamma(a)$
Multivariate Gaussian	$p(X) = \mathcal{N}(X; A, B)$	$H[X] = H[X_1, X_2, \ldots, X_m] = \frac{1}{2} \log((2\pi e)^n \lvert K \rvert)$

KL divergence of two Gaussian Distributions

KL divergence between two Gaussian distributions $q(x) = \mathcal{G}(x|\alpha, \beta)$ and $p(x) = \mathcal{G}(x|a, b)$

$$D_{KL}(q||p) = \frac{(\alpha - a)^2}{2b} + \frac{1}{2}\left(\frac{\beta}{b} - 1 - \log \frac{\beta}{b}\right) \tag{B.32}$$

KL divergence of two Gamma Distributions

KL divergence between two gamma distributions $q(x) = \mathcal{G}(x|\alpha, \beta)$ and $p(x) = \mathcal{G}(x|a, b)$

$$D_{KL}(q||p) = (\alpha - 1)\psi(\alpha) - \log\beta - \alpha - \log\Gamma(\alpha) \tag{B.33}$$
$$+ \log(\Gamma(a)) + a\log b - (a-1)(\psi(\alpha) + \log(\beta)) + \frac{\alpha\beta}{b}$$
$$= \log\frac{\Gamma(a)\beta^{\alpha}}{\Gamma(\alpha)b^{a}} + (\alpha - a)(\psi(\alpha) - \log\beta) + \alpha(\frac{b-\beta}{\beta}) \tag{B.34}$$

KL divergence of two Poisson Distributions

KL divergence between two Poisson distributions $q(x) = \mathcal{PO}(x|\alpha)$ and $p(x) = \mathcal{PO}(x|a)$

$$D_{KL}(q||p) = \alpha\log\frac{\alpha}{a} + a - \alpha \tag{B.35}$$

where as expected it meets the definition of the KL divergence.

APPENDIX C: MISCELLANEOUS

C.1. Miscellaneous

C.1.1. Matrix Operations

Determinant of a diagonal matrix

$$X = diag(\lambda_1, \lambda_2, \ldots \lambda_n) \tag{C.1}$$

$$|X| = \prod_i^n \lambda_i \tag{C.2}$$

Eigenvalues of a diagonal matrix

$$X = diag(\lambda_1, \lambda_2, \ldots \lambda_n) \tag{C.3}$$

$$eig(X) = \lambda_1, \lambda_2, \ldots \lambda_n \tag{C.4}$$

Determinant of a square matrix

$$eig(X) = \lambda_1, \lambda_2, \ldots \lambda_n \tag{C.5}$$

$$|X| = \prod_i^n \lambda_i \tag{C.6}$$

C.1.2. Gamma function and Stirling Approximation

Gamma function extends the factorial function to real and complex numbers as

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt \qquad z \text{ real} \tag{C.7}$$

$$\Gamma(n) = (n-1)! \qquad n \text{ integer} \tag{C.8}$$

For large (integer) n the factorial function is approximated as

$$n! \approx \sqrt{2\pi n}(\frac{n}{e})^n \quad \Rightarrow \quad \log n! \approx \frac{1}{2}\log(2\pi n) + n\log n - n \tag{C.9}$$

Dividing by n (to get $\Gamma(n) = (n-1)!$) and ignoring the constant term $\frac{1}{2}\log(2\pi)$ we may write the following approximation

$$\log\Gamma(n) \approx -\frac{1}{2}\log n + n\log n - n \qquad n \to \infty \tag{C.10}$$

REFERENCES

1. Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

2. Comon, P., "Independent Component Analysis: A New Concept?", *Signal Processing, Elsevier*, Vol. 36, No. 3, pp. 287–314, 1994.

3. Hyvärinen, A., J. Karhunen and E. Oja, *Independent Component Analysis*, Wiley, New York, 2001.

4. Paatero, P. and U. Tapper, "Positive Matrix Factorization: A Non-Negative Factor Model with Optimal Utilization of Error Estimates of Data Values", *Environmetrics*, Vol. 5, 1994.

5. Lee, D. D. and H. S. Seung, "Algorithms for Non-negative Matrix Factorization", *Neural Information Processing Systems*, Vol. 13, pp. 556–562, 2001.

6. Deerwester, S., S. Dumais, T. Landauer, G. Furnas and L. Beck, "Improving Information Retrieval with Latent Semantic Indexing", *Proceedings of the 51st Annual Meeting of the ASIS*, Vol. 25, pp. 36–40, 1988.

7. Koren, Y., R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems", *IEEE Computer*, Vol. 42, pp. 30–37, 2009.

8. Wainwright, M. and M. I. Jordan, "Graphical Models, Exponential Families, and Variational Inference", *Foundations and Trends in Machine Learning*, Vol. 1, pp. 1–305, 2008.

9. Golub, G. H. and C. F. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, The Johns Hopkins University, 3rd edn., 1996.

10. Hoyer, P., *Non-negative Matrix Factorization with Sparseness Constraints*, 2007, `http://www.hiit.fi/node/70/`, accessed at December 2011.

11. Smaragdis, P. and J. C. Brown, "Non-Negative Matrix Factorization for Polyphonic Music Transcription", *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 177–180, 2003.

12. Kiers, H. A. L., "Towards a Standardized Notation and Terminology in Multiway Analysis", *Journal of Chemometrics*, Vol. 14, pp. 105–122, 2000.

13. Kolda, T. G. and B. W. Bader, "Tensor Decompositions and Applications", *SIAM Review*, Vol. 51, No. 3, pp. 455–500, 2009.

14. Acar, E. and B. Yener, "Unsupervised Multiway Data Analysis: A Literature Survey", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, No. 1, pp. 6–20, 2009.

15. Hitchcock, F. L., "The Expression of a Tensor or a Polyadic as a Sum of Products", *Journal of Mathematical Physics*, Vol. 6, pp. 164–189, 1927.

16. Tucker, L. R., "Implications of Factor Analysis of Three-way Matrices for Measurement of Change", C. W. Harris (Editor), *Problems in Measuring Change*, pp. 122–137, University of Wisconsin Press, Madison WI, 1963.

17. Carroll, J. D. and J. J. Chang, "Analysis of Individual Differences in Multidimensional Scaling via an N-way Generalization of Eckart-Young Decomposition", *Psychometrika*, Vol. 35, pp. 283–319, 1970.

18. Harshman, R. A., *Foundations of the PARAFAC Procedure: Models and Conditions for an Explanatory Multi-modal Factor Analysis*, Tech. Rep. 16, UCLA Working Papers in Phonetics, 1970.

19. Fevotte, C., N. Bertin and J. L. Durrieu, "Nonnegative Matrix Factorization with

the Itakura-Saito Divergence. With application to music analysis", *Neural Computation*, Vol. 21, 2009.

20. Wikipedia, *Scale Invariance*, 2011, `http://en.wikipedia.org/wiki/Scale_invariance`, accessed at December 2011.

21. Itakura, F. and S. Saito, "Analysis Synthesis Telephony Based on the Maximum Likelihood Method", *Proceedings of IEEE 6th International Congress on Acoustics*, pp. C–17–C–20, 1968.

22. Banerjee, A., S. Merugu, I. S. Dhillon and J. Ghosh, "Clustering with Bregman Divergences", *Journal of Machine Learning Research*, Vol. 6, pp. 1705–1749, 2005.

23. Jorgensen, B., "Exponential Dispersion Models", *J. of the Royal Statistical Society. Series B*, Vol. 49, pp. 127–162, 1987.

24. Tweedie, M. C. K., "An Index Which Distinguishes Between Some Important Exponential Families", *Statistics: applications and new directions, Indian Statist. Inst., Calcutta*, pp. 579–604, 1984.

25. Bar-Lev, S. K. and P. Enis, "Reproducibility and Natural Exponential Families with Power Variance Functions", *Annals of Statistics*, Vol. 14, pp. 1507–1522, 1986.

26. Basu, A., I. R. Harris, N. L. Hjort and M. C. Jones, "Robust and Efficient Estimation by Minimising a Density Power Divergence", *Biometrika*, Vol. 85, No. 3, pp. 549–559, 1998.

27. Eguchi, S. and Y. Kano, *Robustifing Maximum Likelihood Estimation*, Tech. rep., Institute of Statistical Mathematics in Tokyo, 2001.

28. Minami, M. and S. Eguchi, "Robust Blind Source Separation by Beta Divergence", *Neural Computation*, Vol. 14, pp. 1859–1886, 2002.

29. Salakhutdinov, R. and A. Mnih, "Probabilistic Matrix Factorization", *Advances in Neural Information Processing Systems*, Vol. 20, 2008.

30. Cemgil, A. T., "Bayesian Inference for Nonnegative Matrix Factorisation Models", *Computational Intelligence and Neuroscience*, Vol. 2009, pp. 1–17, 2009.

31. Fevotte, C. and A. T. Cemgil, "Nonnegative Matrix Factorisations as Probabilistic Inference in Composite Models", *Proceddings of 17th European Signal Processing Conference*, 2009.

32. Cichocki, A., R. Zdunek, A. H. Phan and S. Amari, *Nonnegative Matrix and Tensor Factorization*, Wiley, 2009.

33. Mørup, M., L. K. Hansen and S. M. Arnfred, "Algorithms for Sparse Non-negative TUCKER", *Neural Computation*, Vol. 20, No. 8, pp. 2112–2131, 2008.

34. Porteous, I., E. Bart and M. Welling, "Multi-HDP: A Non Parametric Bayesian Model for Tensor Factorization", *in Proceedings of the 23rd national conference on Artificial intelligence*, Vol. 3, pp. 1487–1490, 2008.

35. Schmidt, M. and S. Mohamed, "Probabilistic Non-negative Tensor Factorisation using Markov Chain Monte Carlo", *17th European Signal Processing Conference*, 2009.

36. Févotte, C. and A. Ozerov, "Notes on Nonnegative Tensor Factorization of the Spectrogram for Audio Source Separation : Statistical Insights and Towards Self-clustering of the Spatial Cues", *7th International Symposium on Computer Music Modeling and Retrieval*, 2010.

37. Bernardo, J. M. and A. F. M. Smith, *Bayesian Theory*, John Wiley & Sons, Inc., New York, 1994.

38. Cover, T. M. and J. A. Thomas, *Elements of Information Theory*, Wiley-

Interscience, New York, NY, USA, 1991.

39. Collins, M., S. Dasgupta and R. E. Schapire, "A Generalization of Principal Component Analysis to the Exponential Family", *Advances in Neural Information Processing Systems*, MIT Press, 2001.

40. Ghahramani, Z. and M. Beal, "Propagation Algorithms for Variational Bayesian Learning", *Neural Information Processing Systems*, Vol. 13, 2000.

41. Kruskal, J. B., "Three-way Arrays: Rank and Uniqueness of Trilinear Decompositions, with Application to Arithmetic Complexity and Statistics", *Linear Algebra Applications*, Vol. 18, pp. 95–138, 1977.

42. Kruskal, J. B., *Rank, Decomposition, and Uniqueness for 3-way and N-way Arrays. Multiway Data Analysis*, North-Holland, Amsterdam, 1989.

43. Acar, E., T. G. Kolda and D. M. Dunlavy, *All-at-once Optimization for Coupled Matrix and Tensor Factorizations*, 2011, `http://arxiv.org/abs/1105.3422`, accessed at December 2011.

44. Şimşekli, U., Y. K. Yılmaz and A. T. Cemgil, "Score Guided Audio Restoration Via Generalised Coupled Tensor Factorization", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2012.

45. Yılmaz, Y. K. and A. T. Cemgil, "Probabilistic Latent Tensor Factorization", *Proceedings of the 9th international conference on Latent variable analysis and signal separation*, LVA/ICA'10, pp. 346–353, Springer-Verlag, 2010.

46. Jordan, M. I. (Editor), *Learning in Graphical Models*, MIT Press, 1999.

47. Tweedie, M. C. K., "Functions of a Statistical Variate with Given Means, with Special Reference to Laplacian Distributions", *Proceedings of the Cambridge Philosophical Society*, Vol. 49, pp. 41–49, 1947.

48. Yılmaz, Y. K. and A. T. Cemgil, *Algorithms for Probabilistic Latent Tensor Factorization*, 2011, http://www.sciencedirect.com/science/article/pii/S0165168411003537, accessed at October 2011.

49. Fevotte, C. and J. Idier, "Algorithms for Nonnegative Matrix Factorization with the Beta Divergence", *Neural Computation*, Vol. 23, No. 9, pp. 2421–2456, 2011.

50. Nelder, J. A. and R. W. M. Wedderburn, "Generalized Linear Models", *Journal of the Royal Statistical Society, Series A*, Vol. 135, pp. 370–384, 1972.

51. Yılmaz, Y. K., A. T. Cemgil and U. Şimşekli, "Generalised Coupled Tensor Factorisation", *Neural Information Processing Systems*, 2011.

52. Menendez, M. L., "Shannon's Entropy in Exponential Families: Statistical Applications", *Applied Mathematics Letters*, Vol. 13, No. 1, pp. 37–42, 2000.

53. Winn, J. M., *Variational Message Passing and its Applications*, Ph.D. Thesis, University of Cambridge, 2003.

54. Agarwal, A. and H. III Daumé, "A Geometric View of Conjugate Priors", *Machine Learning*, Vol. 81, pp. 99–113, 2010.

55. Dempster, N. M. L., A. P. and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society*, Vol. B 39, No. 1, pp. 1–38, 1977.

56. Neal, R. and G. E. Hinton, "A View Of The EM Algorithm That Justifies Incremental, Sparse, And Other Variants", *Learning in Graphical Models*, pp. 355–368, Kluwer Academic Publishers, 1998.

57. Minka, T. P., *Expectation Maximization as Lower Lound Maximization*, 1998, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.8562, accessed at December 2011.

58. Heckerman, D., C. Meek and G. Cooper, *A Bayesian Approach to Causal Discovery*, Tech. rep., 1997.

59. Bishop, C. M., *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.

60. Cooper, G. F., "The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks", *Artificial Intelligence*, Vol. 42, No. 2-3, pp. 393–405, 1990.

61. Huang, C. and A. Darwiche, "Inference in Belief Networks: A Procedural Guide", *International Journal of Approximate Reasoning*, Vol. 15, pp. 225–263, 1994.

62. Morris, C. N., "Natural Exponential Families with Quadratic Variance Functions", *Annals of Statistics*, Vol. 10, pp. 65–80, 1982.

63. Cichocki, A., S. Cruces and S. Amari, "Generalized Alpha-Beta Divergences and Their Application to Robust Nonnegative Matrix Factorization", *Entropy*, Vol. 13, No. 1, pp. 134–170, 2011.

64. Lafferty, J., "Additive Models, Boosting, and Inference for Generalized Divergences", *Proceedings of Annual Conference on Computational Learning Theory*, pp. 125–133, ACM Press, 1999.

65. Kus, V., D. Morales and I. Vajda, "Extensions of the Parametric Families of Divergences Used in Statistical Inference", *Kybernetika*, Vol. 44, No. 1, pp. 95–112, 2008.

66. Amari, A., "Information Geometry in Optimization, Machine Learning and Statistical Inference", *Frontiers of Electrical and Electronic Engineering in China*, Vol. 5, No. 3, pp. 241–260, 2010.

67. McCulloch, C. E. and J. A. Nelder, *Generalized Linear Models*, Chapman and Hall,

2nd edn., 1989.

68. Diaconis, P. and D. Ylvisaker, "Conjugate Priors for Exponential Families", *Annals of Statistics*, Vol. 7, pp. 269–281, 1979.

69. Gordon, G. J., "Generalized^2 Linear^2 Models", *Neural Information Processing Systems*, 2002.

70. Cemgil, A. T., U. Simsekli and Y. C. Subakan, "Probabilistic Latent Tensor Factorization Framework for Audio Modeling", *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2011.

71. Meyer, P. L., *Introductory Probability and Statistical Applications*, Reading MA, Addison-Wesley, 2nd edn., 1970.

72. Kaas, R., D. Dannenburg and M. Goovaerts, "Exact Credibility for Weighted Observations", *Actuarial Studies In Non-life insurance Bulletin*, Vol. 27, pp. 287–295, 1997.

73. Andersson, C. A. and R. Bro, "The N-way Toolbox for MATLAB", *Chemometrics and Intelligent Laboratory Systems*, Vol. 52, p. pp. 14, 2000.

74. de Almeida, A. L. F., G. Favier and J. C. M. Mota, "Space-time Spreading-multiplexing for MIMO Wireless Communication Systems Using the PARATUCK-2 Tensor Model", *Signal Processing*, Vol. 89, No. 11, pp. 2103–2116, 2009.

75. Beal, M. J., *Variational Algorithms for Approximate Bayesian Inference*, Ph.D. Thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

76. McCulloch, C. E. and S. R. Searle, *Generalized, Linear, and Mixed Models*, Wiley, 2001.

77. Kaas, R., *Compound Poisson Distributions And GLM's Tweedie's Distribution*,

Tech. rep., Lecture, Royal Flemish Academy of Belgium for Science and the Arts, 2005.

78. Magnus, J. R. and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, Wiley, 3rd edn., 2007.

79. Csiszar, I. and G. Tusndy, "Information Geometry and Alternating Minimization Procedures", *Statistics and Decisions, Supplement Issue*, Vol. 1, pp. 205–237, 1984.

80. Singh, A. P. and G. J. Gordon, "A Unified View of Matrix Factorization Models", *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Part II*, 5212, pp. 358–373, Springer, 2008.

81. Chong, E. and S. Żak, *An Introduction to Optimization*, John Wiley & Sons, New York, 1996.

82. Marcus, M. and H. Minc, *A Survey of Matrix Theory and Matrix Inequalities*, Dover, 1992.

83. Harville, D. A., *Matrix Albegra From A Statistician's Perspective*, Springer, 1997.

84. Xu, Q., E. W. Xiang and Q. Yang, "Protein-protein Interaction Prediction via Collective Matrix Factorization", *Proceedings of the IEEE International Conference on Bioinformatics Biomedicine*, pp. 62–67, 2010.