UNIVERSITY OF CALIFORNIA, SAN DIEGO

Generalized Statistical Methods for Mixed Exponential Families

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in
Electrical and Computer Engineering
(Signal and Image Processing)

by

Cécile Levasseur

Committee in charge:

Kenneth Kreutz-Delgado, Chair
Uwe F. Mayer, Co-Chair
Ian Abramson
Sanjoy Dasgupta
Gert Lanckriet
Bhaskar D. Rao

2009

The dissertation of Cécile Levasseur is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2009

DEDICATION

To my mother, for her infinite love and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I am grateful to my advisor, Professor Kenneth Kreutz-Delgado, for his continual guidance, encouragement, and enthusiasm throughout the duration of my PhD study. His unfailing support and patience were precious gifts, especially during the difficult mourning of my father. I am also grateful to Uwe Mayer who, over the years, has become my second advisor. Under his constant and thorough guidance I learned to be both creative and attentive to details to successfully solve real-world problems. I would also like to thank my other committee members, Professors Ian Abramson, Sanjoy Dasgupta, Gert Lanckriet and Bhaskar Rao for their time and interest in my thesis topic.

I would also like to thank the University of California Microelectronics Innovation and Computer Research Opportunities (MICRO) Program and the affiliated company Fair Isaac Corporation for supporting me during four consecutive years of my PhD program. The sustained collaboration with Fair Isaac Corporation would not have been possible without the enthusiasm and support of Gregory Gancarz, Mike Lazarus and Uwe Mayer. This work was supported in part by MICRO Projects ID 03-042, 04-046, 05-011 and 06-173. The funds for the last quarters of my PhD program were provided by the National Science Foundation grant No. CCF-0830612. I am also thankful for the support of Professor Bhaskar Rao and the resources of the Digital Signal Processing Lab in the UCSD ECE department.

Chapters 3 and 4, in part, are a reprint of the material as it appears in "Data-pattern discovery methods for detection in nongaussian high-dimensional data sets," C. Levasseur, K. Kreutz-Delgado, U. Mayer and G. Gancarz, in *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, pp. 545–549, Nov. 2005 and "Generalized statistical methods for unsupervised minority class detection in mixed data sets," C. Levasseur, U. F. Mayer, B. Burdge and K. Kreutz-Delgado, in *Proceedings of the First IAPR Workshop on Cognitive Information Processing (CIP)*, pp. 126–131, June 2008. Chapter 5, in

part, is a reprint of the material as it appears in "A unifying viewpoint of some clustering techniques using Bregman divergences and extensions to mixed data sets," C. Levasseur, B. Burdge, K. Kreutz-Delgado and U. F. Mayer, in *Proceedings of the First IEEE International Workshop on Data Mining and Artificial Intelligence (DMAI)*, pp. 56–63, Dec. 2008. Chapters 3, 4 and 5, in part, are a reprint of the material as it appears in "Generalized statistical methods for mixed exponential families, part I: theoretical foundations," C. Levasseur, K. Kreutz-Delgado and U. F. Mayer, submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Sept. 2009. Appendix D, in part, is a reprint of the material as it will appear in "Classifying non-Gaussian and mixed data sets in their natural parameter space," C. Levasseur, U. F. Mayer and K. Kreutz-Delgado, in *Proceedings of the Nineteenth IEEE International Workshop in Machine Learning for Signal Processing (MLSP)*, Sept. 2009. Chapter 4, Chapter 5 and Appendix D, in part, are a reprint of the material as it appears in "Generalized statistical methods for mixed exponential families, part II: applications," C. Levasseur, U. F. Mayer and K. Kreutz-Delgado, submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Sept. 2009. The dissertation author was the primary researcher and author, and the co-authors listed in these publications contributed to or supervised the research which forms the basis for this dissertation.

I would like to thank my fellow graduate students in Professors Kenneth Kreutz-Delgado and Bhaskar Rao's Digital Signal Processing Lab. In particular, I am grateful to Brandon Burdge and Joseph Murray for invaluable technical discussions. I also thank Yogananda Isukapalli for inspiring conversations and debates.

I would like to thank my pilates/yoga instructor Alexia Cervantes for helping me keep the stress level under control and transform most Mondays and Wednesdays into more peaceful days.

I also would like to thank my support network and friends at UCSD and elsewhere. Thanks to Azadeh Bozorgzadeh, Elizabeth Gire, Laddan Hashemian, Mohamed Jalloh, Periklis Liaskovitis, Maziar Nezhad and Kostas Stamatiou for

countless lively lunch-breaks. Thanks to my favorite roommate Danka Vuga for her infinite energy and enthusiastic support. Thanks to my friend Jittra Jootar for her continual support. A special thank you to my friend Yiannis Spyropoulos, my favorite lunch partner, for sharing our ups and downs during these emotional last years as a PhD student. Thanks to my international support group, Fanny Després (France), Valérie Duay (Switzerland) and Richard Hurni (Switzerland) for their unfailing friendship, continual support and faith in my abilities. Thanks to Aurélie Lozano and her husband H.-C. Huang for their generous friendship, for the many unforgettable dinners and concerts in New York City. Thanks to my dearest friend Chandra Murthy for a beautiful friendship and an unforgettable first journey to India.

I would like to express my sincere gratitude to my family for their unquestioning love and support. From the deepest of my heart, I thank my mother, my father who most certainly watches over me, and my two brothers Marc and Arnaud for allowing me freedom to endlessly experience and explore. I thank Christopher and his family for adopting me, loving me and supporting me totally, fully and unconditionally.

VITA

1975	Born, Ivry-sur-Seine, France
2002	B.S., Communication Systems Swiss Federal Institute of Technology Lausanne, Switzerland
2004	M.S., Electrical and Computer Engineering University of California, San Diego, U.S.A.
2009	Ph.D., Electrical and Computer Engineering University of California, San Diego, U.S.A.

PUBLICATIONS

C. Levasseur, K. Kreutz-Delgado, U. Mayer and G. Gancarz, "Data-pattern discovery methods for detection in nongaussian high-dimensional data sets," *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, pp. 545–549, Nov. 2005.

C. Levasseur, U. F. Mayer, B. Burdge and K. Kreutz-Delgado, "Generalized statistical methods for unsupervised minority class detection in mixed data sets," *Proceedings of the First IAPR Workshop on Cognitive Information Processing (CIP)*, pp. 126–131, June 2008.

C. Levasseur, B. Burdge, K. Kreutz-Delgado and U. F. Mayer, "A unifying viewpoint of some clustering techniques using Bregman divergences and extensions to mixed data sets," *Proceedings of the First IEEE Int'l Workshop on Data Mining and Artificial Intelligence (DMAI)* held in conjunction with *the Eleventh IEEE Int'l Conference on Computer and Information Technology*, pp. 56–63, Dec. 2008.

C. Levasseur, U. F. Mayer and K. Kreutz-Delgado, "Classifying non-Gaussian and mixed data sets in their natural parameter space," to appear in the *Proceedings of the Nineteenth IEEE Int'l Workshop on Machine Learning for Signal Processing (MLSP)*, Sept. 2009.

C. Levasseur, K. Kreutz-Delgado and U. F. Mayer, "Generalized statistical methods for mixed exponential families, part I: theoretical foundations," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, submitted, Sept. 2009.

C. Levasseur, U. F. Mayer and K. Kreutz-Delgado, "Generalized statistical methods for mixed exponential families, part II: applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, submitted, Sept. 2009.

ABSTRACT OF THE DISSERTATION

Generalized Statistical Methods for Mixed Exponential Families

by

Cécile Levasseur

Doctor of Philosophy in Electrical and Computer Engineering
(Signal and Image Processing)

University of California, San Diego, 2009

Kenneth Kreutz-Delgado, Chair
Uwe F. Mayer, Co-Chair

This dissertation considers the problem of learning the underlying statistical structure of complex data sets for fitting a generative model, and for both supervised and unsupervised data-driven decision making purposes. Using properties of exponential family distributions, a new unified theoretical model called Generalized Linear Statistics is established.

The complexity of data is generally a consequence of the existence of a large number of components and the fact that the components are often of mixed data types (i.e., some components might be continuous, with different underlying distributions, while other components might be discrete, such as categorical, count or Boolean). Such complex data sets are typical in drug discovery, health care, or fraud detection.

The proposed statistical modeling approach is a generalization and amalgamation of techniques from classical linear statistics placed into a unified framework referred to as Generalized Linear Statistics (GLS). This framework includes techniques drawn from latent variable analysis as well as from the theory of Generalized Linear Models (GLMs), and is based on the use of exponential family

distributions to model the various mixed types (continuous and discrete) of complex data sets. The methodology exploits the connection between data space and parameter space present in exponential family distributions and solves a nonlinear problem by using classical linear statistical tools applied to data that have been mapped into parameter space.

One key aspect of the GLS framework is that often the natural parameter of the exponential family distributions is assumed to be constrained to a lower dimensional latent variable subspace, modeling the belief that the intrinsic dimensionality of the data is smaller than the dimensionality of the observation space.

The framework is equivalent to a computationally tractable, mixed data-type hierarchical Bayes graphical model assumption with latent variables constrained to a low-dimensional parameter subspace. We demonstrate that exponential family Principal Component Analysis, Semi-Parametric exponential family Principal Component Analysis, and Bregman soft clustering are not separate unrelated algorithms, but different manifestations of model assumptions and parameter choices taken within this common GLS framework. Because of this insight, these algorithms are readily extended to deal with the important mixed data-type case. This framework has the critical advantage of allowing one to transfer high-dimensional mixed-type data components to low-dimensional common-type latent variables, which are then, in turn, used to perform regression or classification in a much simpler manner using well-known continuous-parameter classical linear techniques.

Classification results on synthetic data and data sets from the University of California, Irvine machine learning repository are presented.

1 Introduction

Many important risk assessment system applications depend on the ability to accurately detect the occurrence of key events and/or predict their probabilities given a large data set of observations. For example, this problem frequently arises in drug discovery ("Do the molecular descriptors associated with known drugs suggest that a new candidate drug will have low toxicity and high effectiveness?"), medicine ("Do the epidemiological data suggest that the trace elements in the local water supply cause cancer?"), health care ("Do the descriptors associated with the professional behavior of a medical health-care worker suggest that he/she is an outlier in the efficacy category he/she was assigned to?") and failure prediction ("Based on the measurements of monitored disk-drive reliability and performance metrics, what is the probability that the hard drive containing my dissertation will fail in the next 72 hours?"). As another important example, the financial industry is concerned with the problem of fraud detection, such as credit card fraud detection ("Given the data for a large set of credit card users, does the usage pattern of this particular card indicate that it might have been stolen?"). Often in such domains, little or no *a priori* knowledge exists regarding the true sources of any causal relationships that may occur between variables of interest. In these situations, meaningful information regarding the key events must be extracted from the data itself.

The problem of *supervised* data-driven detection or prediction is one of relating descriptors of a large, *labeled* database of "objects" (e.g., credit card transactions) to measured properties of these objects, then using these empirically de-

termined relationships to categorize, or infer the properties of, new objects. For the risk assessment problems suggested earlier, the measured object properties are often non-Gaussian (and comprised of categorical, count, and continuous data), possibly very noisy, and highly non-linearly related. As a consequence, the resulting categorization problem is very difficult. In many cases, the difficulties are further compounded because the descriptor space of objects is of very high dimension. Even worse, the database of training examples may be *unlabeled*, in which case *unsupervised* training methods must be used. For example, it is quite common in the financial industry to have large existing databases containing a mixture of both non-fraudulent and fraudulent events which have never been tagged as such, yet which provide the only data available for training purposes. Lack of fully labeled data often occurs in other important domains as well. To ameliorate these difficulties, we developed appropriate probability models for classes of objects (such as the credit card transaction classes "fraudulent" and "non-fraudulent" for example) which have low-dimensional parameterizations and associated low-dimensional approximate sufficient statistics ("features"). These models can then be used for supervised and unsupervised classification.

The approach proposed and utilized here is a generalization and amalgamation of techniques from classical linear statistics, logistic regression, Principal Component Analysis (PCA), and Generalized Linear Models (GLMs) into a framework referred to, analogously to GLMs theory, as *Generalized Linear Statistics (GLS)*. As defined in this dissertation, Generalized Linear Statistics includes techniques drawn from latent variable analysis [1, 2] as well as from the theory of Generalized Linear Models (GLMs) and Generalized Linear Mixed Models (GLMMs) [3–6]. It is based on the use of exponential family distributions to model the various mixed types (continuous or discrete) of measured object properties. Despite the name, this is a *nonlinear* methodology which exploits the separation in exponential family distributions between the *data space* (also known as the *expected value space*) and the *parameter space* as soon as one leaves the domain of purely

Gaussian random variables. The point is that although the problem at hand may be nonlinear, it can be attacked using classical linear and other standard statistical tools applied to data that have been *mapped into the parameter space*, which is assumed to have a natural, flat Euclidean space structure. For example, in the parameter space one can perform regression (resulting in the technique of logistic regression and other GLMs methods [3–8]), PCA (resulting in a variety of "generalized PCA" methods [2, 9–13]), or clustering [14–16]. This approach provides an effective way to exploit tractably parameterized latent-variable exponential-family probability models to address the problem of data-driven learning of model parameters and features useful for the development of effective classification and regression algorithms.

To reiterate, the Generalized Linear Statistics framework draws inspiration primarily from Generalized Linear Models and latent variable modeling. The additional use of the Non-Parametric Maximum Likelihood (NPML) estimation technique brings added flexibility to the model and enhanced generalization characteristics. Laird's classic 1978 paper [17] appears to be generally acknowledged as the first paper to propose the Expectation-Maximization (EM) algorithm for Non-Parametric Maximum Likelihood estimation in the mixture density context, cf. Figure 1.1. A few years earlier, Simar, in his 1976 paper [18], studied maximum likelihood estimation in the case of mixtures of Poisson distributions, but the EM algorithm was not available at the time. A few years later, Jewell, in his 1982 paper [19], followed Simar's approach and drew similar results focusing on the particular case of mixtures of Exponential distributions, but suggested the EM algorithm for the maximum likelihood estimation of the mixing distribution, as proposed previously by Laird. After Lindsay's classic 1983 papers [20, 21] placed the NPML approach on a more rigorous footing by exploiting the convex geometry properties of the likelihood function, Mallet's paper appeared in 1986 [22] and further explored some of the fundamental issues raised by Lindsay using optimal design theory. Laird, Lindsay and Mallet, to a greater or lesser extent, all point

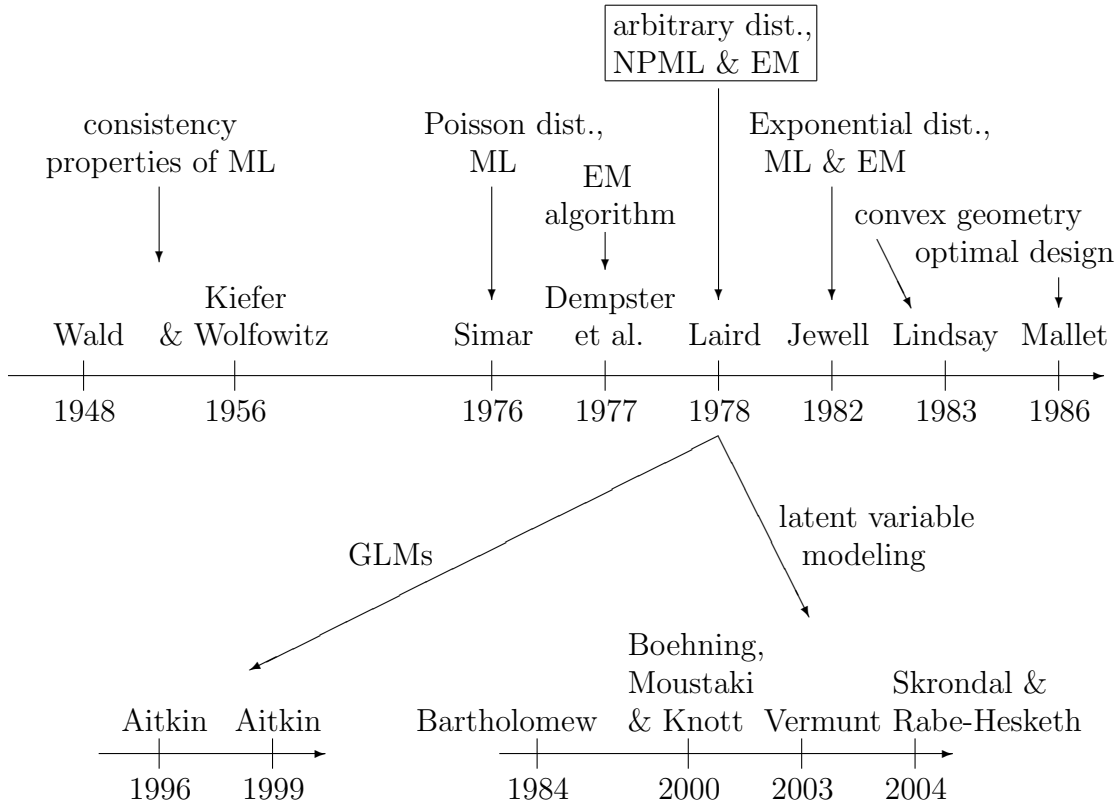

Figure 1.1 Chronology of ideas and influence of Non-Parametric Maximum Likelihood (NPML) in Generalized Linear Models (GLMs) and latent variable modeling.

out that the presence of nuisance (incidental) parameters has to be addressed in order to assess the consistency properties of the maximum likelihood estimator and that the early work by Wald [23, 24] and by Kiefer and Wolfowitz [25] is relevant in this regard. Also relevant is the paper on EM-based non-parametric density estimation in pharmacokinetic models by Schumitzky (1991) [26].

It was only natural that the GLMs community would begin to explore the NPML approach as a vehicle to deal with the important and difficult problem of parameter estimation for random- and mixed-effects GLMs and, indeed, one can see the fruitful outcome of such inquiry in the 1996 and 1999 papers by Aitkin [27–30]. It was also natural for the latent variable research community to

follow up on the NPML thread. After all, in the classical linear Gaussian situation, regression and latent variable (i.e., factor analysis) problems are different levels of the problem of learning a linear model, the simpler problem of regression being the "non-blind" case and the more difficult problem of factor analysis being the "blind" case (using the terminology common to communications engineers). The latent variable research community was well aware of the (now classic) 1984 paper by Bartholomew [31] (whose ideas are now fully exploited in the 1999 text by Bartholomew and Knott [1]), which placed latent variable analysis in a unifying framework and made the points of connection between that theory and the mixed data-type GLMs framework quite evident. Not surprisingly, then, other researchers (Knott, Skrondal, Rabe-Hesketh and Moustaki) explored the problem of non-parametric latent variable analysis, including the use of the NPML approach as suggested by Aitkin. An interesting paper along this line of inquiry was written by Moustaki and Knott (2000) [32], while a more recent one was written by Vermunt (2003) [33]. A good review of the literature through the end of the 1990's and exposition of EM-based NPML in latent variable and GLMs analysis is given in a book written by Boehning in 2000 [34]. A more up-to-date and thorough presentation can found in the book by Skrondal and Rabe-Hesketh written in 2004 [2]. In addition to the ideas described in the literature cited above, such as Aitkin [27–30] and Laird [17], the analysis to construct the framework outlined in this dissertation draws from Collins et al. [10].

The specific Generalized Linear Statistics (GLS) framework developed here is equivalent to a mixed data-type hierarchical Bayes graphical model assumption with latent variables constrained to a low-dimensional parameter subspace. Although a variety of techniques exist for performing inference on graphical models, it is in general very difficult to learn the parameters which constitute the model even if it is assumed that the graph structure is known [35, 36]. A novel and important aspect of this work is that the GLS graphical model can be learned, which provides important insight into the underlying statistical structure of a complex

data set and allows the development of a variety of inference techniques, either by using well-established Bayesian Network techniques [37], or by developing new techniques. In addition to providing a better understanding of the data, learning the GLS model provides a generative model of the data, making it possible to generate synthetic data with the same statistical structure as the original data. This is particularly useful in cases where data are very difficult or expensive to obtain and when the original data are proprietary and cannot be used for publication purposes (as encountered in the financial services industry).

1.1 Outline of this thesis

Chapter 2 presents a review of several well-known statistical modeling techniques that will be referred to and exploited in subsequent sections, such as Generalized Linear Models (GLMs) and latent variable modeling. The relationships between these techniques and others like Principal Component Analysis (PCA) and factor analysis are particularly emphasized.

Chapter 3 presents the proposed statistical modeling approach as a generalization of techniques drawn from classical linear statistics, logistic regression, Principal Component Analysis (PCA), and Generalized Linear Models (GLMs), all amalgamated into a new framework referred to as *Generalized Linear Statistics* (GLS). It is presented in a mixed data-type hierarchical Bayes graphical model framework.

Chapter 4 exposes the convex optimization problem related to fitting one extreme of the GLS model to a set of data. This extreme case of the GLS model is similar to exponential family Principal Component Analysis, proposed in [10], and is characterized by the fact that each data point is mapped to one (generally different) parameter point in parameter space, whereas the general GLS case considers a set of parameter points shared by all the data points. In light of the significant numerical difficulties associated with the cyclic-coordinate descent-like

algorithm based on Bregman distance properties proposed in [10], especially in the mixed data-type case, this dissertation focuses on an algorithm based on Iterative Reweighted Least Squares (IRLS), an approach commonly used in the GLMs literature [4, 38, 39]. Using an IRLS-based learning algorithm makes it possible to tractably attack the more general problem of prediction in a mixed data-type environment. Since the optimal model parameter values in this optimization problem may be non-finite [10], a penalty function is introduced that defines and places a set of constraints onto the loss function via a penalty parameter in a way so that any divergence to infinity is avoided. Additionally, for several exponential family distributions with natural restrictions on their parameter, a positivity constraint on the natural parameter values has to be introduced. Synthetic data examples for several exponential family distributions in both mixed and non-mixed data-type cases are presented and generative models are fit to the data. Furthermore, an unsupervised minority class detection technique to be performed in parameter space is proposed. The synthetic data example demonstrates that there are domains for which classical linear techniques used in the data space, such as PCA, perform significantly worse than the new proposed parameter space technique.

Chapter 5 presents a general point of view that relates the exponential family Principal Component Analysis (exponential PCA) technique of [10] to the Semi-Parametric exponential family Principal Component Analysis (SP-PCA) technique of [15] and to the Bregman soft clustering method presented in [16]. The proposed viewpoint is then illustrated with a clustering problem in mixed data sets. The three techniques considered here all utilize Bregman distances and can all be explained within a single hierarchical Bayes graphical model framework. They are not separate unrelated algorithms, but different manifestations of model assumptions and parameter choices taken within a common framework. Selecting a Bayesian or a classical approach as well as various parametric choices in the proposed model are demonstrated to determine the three algorithms. Because of this insight, the algorithms are readily extended to deal with the important mixed

data-type case. Considering synthetic data examples of mixed types, exponential PCA, with the addition of a non-parametric estimation tool, is demonstrated to rival SP-PCA and Bregman soft clustering in terms of clustering performance for some data sets.

Appendix A summarizes properties of exponential family distributions that are essential to the framework presented here. In particular, it introduces Bregman distances, which encompass a large class of distance/divergence functions. Bregman distances are a generalization of the log-likelihood function of any member of the exponential family of distributions and as such, the convexity properties of Bregman distances are important to the optimization problem attacked in Chapter 4. Recently, research has shown that many important algorithms can be generalized from Euclidean metrics to distances defined by a Bregman distance [10, 15, 16, 40], i.e., the algorithms can be generalized from Gaussian distributed data components to data components distributed according to an exponential family, such as binary- or integer-valued.

Appendix B reviews the Newton-Raphson IRLS minimization technique used to learn the Generalized Linear Statistics model.

Appendix C presents the theory supporting non-parametric mixture models within the Generalized Linear Statistics (GLS) framework, including the Non-Parametric Maximum Likelihood (NPML) estimation technique used in Section 3 and Section 5. Then, the Expectation-Maximization (EM) algorithm is developed for the NPML estimation technique with a special focus on exponential family distributions.

Appendix D presents details on the work with University of California, Irvine, machine learning repository data sets [41] and emphasizes the benefits of classifying non-Gaussian and mixed data sets in their natural parameter space.

2 Generalized Linear Models and latent variable modeling

The goal of this section is to review and reveal the relationships between several well-known statistical modeling techniques that will be referred to and exploited in subsequent sections.

There are two general approaches to modeling response processes. In statistics and biostatistics, the most common approach involves Generalized Linear Models (GLMs) techniques, whereas a latent variable modeling, or latent response formulation, is popular in econometrics and psychometrics [2]. Although different in appearance, these approaches can generate equivalent models for many response types. This presentation emphasizes the similarity of the two approaches and relates them to well-known techniques such as logistic regression, factor analysis, Principal Component Analysis (PCA) and discriminant analysis. Figure 2.4 at the end of the chapter helps to summarize and compare these techniques.

2.1 Generalized Linear Models (GLMs)

Generalized Linear Models [3,4,6,38] are a unified class of regression models for discrete and continuous response variables, and have been used routinely in dealing with observational studies [42]. Many statistical developments in terms of modeling and methodology in the past twenty years may be viewed as special cases of GLMs. Examples include logistic regression for binary responses, linear regres-

sion for continuous responses and loglinear models for counts [43]. Applications of the logistic regression model provide a basic tool for epidemiologic investigation of chronic diseases. Similar methods have been extensively used in econometrics. Probit and logistic models play a key role in all forms of assay experiments. The loglinear model is the cornerstone of modern approaches to the analysis of contingency table data, and has been found particularly useful for medical and social sciences. Poisson regression models are widely employed to study rates of events such as disease outcomes. The complementary loglog model arises in the study of infectious diseases and more generally, in the analysis of survival data associated with clinical and longitudinal follow-up studies.

2.1.1 The standard Gaussian linear model, or linear regression model

Consider the conditional probability distribution $p(\mathbf{x}|\boldsymbol{\theta})$ of the outcome or response variable $\mathbf{x}$ given the parameter vector $\boldsymbol{\theta}$ to be a Gaussian distribution with mean $\boldsymbol{\mu}$ and constant covariance matrix. Note that, in this special case of a Gaussian distribution assumption with known covariance matrix, the parameter $\boldsymbol{\theta}$ is equal to the mean vector $\boldsymbol{\mu}$. The standard Gaussian linear model, or *linear regression model*, expresses the conditional expectation of the response variable, given the parameter, in a linear structure as follows:

$$\boldsymbol{\mu} \triangleq E[\mathbf{x}|\boldsymbol{\theta}] = \boldsymbol{\theta} = \mathbf{aV}, \tag{2.1}$$

where $\mathbf{x}$ is a $(1 \times d)$ row vector containing the outcome variable, $\mathbf{a}$ is a $(1 \times q)$ row vector of estimated coefficients and $\mathbf{V}$ is a $(q \times d)$ matrix of explanatory variables. The linear regression model can alternatively be specified by

$$\mathbf{x} = \boldsymbol{\theta} + \boldsymbol{\epsilon},$$

where the residuals $\boldsymbol{\epsilon}$, also called disturbances or error components, are independently Gaussian distributed with zero mean and constant diagonal covariance matrix Ψ, i.e., $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \Psi)$. The $(1 \times d)$ linear predictor vector $\boldsymbol{\theta}$ is called the

systematic component of the linear model while the $(1 \times d)$ vector $\boldsymbol{\epsilon}$ is the *stochastic component*. The linear regression literature often uses the notation $\mathbf{y} = \boldsymbol{\beta}\mathbf{X} + \boldsymbol{\epsilon}$; the reader should be aware of this difference in notation, which we follow to keep the rest of this dissertation consistent (so their $\mathbf{y}$ is our $\mathbf{x}$, their $\mathbf{X}$ is our $\mathbf{V}$, and their $\boldsymbol{\beta}$ is our $\mathbf{a}$).

The matrix $\mathbf{V}$ containing the explanatory variables is assumed to be *deterministic* and *known*. The coefficients vector $\mathbf{a}$ is assumed to be *deterministic* and *unknown*. The parameter vector $\boldsymbol{\theta}$ is often called the linear predictor. The response variables are sometimes called "dependent" variables whereas the explanatory variables are called "independent" variables. In linear regression models, and more generally in Generalized Linear Models, the explanatory variables affect the response only through the linear predictor, and the response process is fully described by specifying the conditional probability of $\mathbf{x}$ given the linear predictor $\boldsymbol{\theta}$. Furthermore, the components $\{x_i\}_{i=1}^{d}$ of the response vector $\mathbf{x}$, conditioned in the parameter $\boldsymbol{\theta}$, are independently distributed,

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{i=1}^{d} p_i(x_i|\boldsymbol{\theta}) = \prod_{i=1}^{d} p_i(x_i|\theta_i).$$

The parameters are estimated according to the principle of least squares and are optimal according to minimum dispersion theory, or, in the case of a Gaussian distribution, are optimal to the Maximum Likelihood (ML) theory [7].

2.1.2 Generalized Linear Models (GLMs)

The basic principal behind Generalized Linear Models (GLMs) is that the systematic component of the linear model can be transformed to create an analytical framework that closely resembles the standard linear model but accommodates a wide variety of non-Gaussian outcome variables. Hence, GLMs are a generalization of the standard Gaussian linear model to the exponential family of distributions. It consists of the following components: first, the conditional probability distribution $p(\mathbf{x}|\boldsymbol{\theta})$ is now assumed to be a member of the exponen-

tial family of distributions with parameter vector $\boldsymbol{\theta}$, cf. Appendix A. Since the Gaussian distribution belongs to the exponential family, allowing $p(\mathbf{x}|\boldsymbol{\theta})$ to be any member of the family means generalizing the standard Gaussian linear model. The probability distribution of the response variable is often referred to as the *random component* of the GLMs [7]. Second, the functional relationship between the conditional expectation of the response variable and the linear predictor is:

$$\boldsymbol{\mu} \triangleq E[\mathbf{x}|\boldsymbol{\theta}] = g(\boldsymbol{\theta}) \quad \text{or} \quad \boldsymbol{\theta} = g^{-1}(\boldsymbol{\mu}) = f(\boldsymbol{\mu}), \tag{2.2}$$

where $g(\cdot)$ is the *link function* associated with the exponential family distribution $p(\mathbf{x}|\boldsymbol{\theta})$. Then, as previously for the standard Gaussian linear model, the parameter $\boldsymbol{\theta}$ is expressed in a linear structure as follows:

$$f(\boldsymbol{\mu}) = \boldsymbol{\theta} = \mathbf{aV}. \tag{2.3}$$

As for the Gaussian linear model, the linear function of the explanatory variables shown in equation (2.3) is called the *systematic component* of the GLMs. Consequently, the link function describes a functional relationship between the systematic component and the expectation of the random component. It provides a bijective relationship between the parameter space and the data space (the conditional expectation of the response variable belongs to the same space as the response variable itself, i.e., to the space referred to as data space), cf. Appendix A.

As before, the matrix $\mathbf{V}$ is assumed to be *deterministic* and *known*, and the vector $\mathbf{a}$ *deterministic* and *unknown*.

For example, for dichotomous or binary responses taking on the values 0 or 1, the response variables $\mathbf{x} = [x_1, \ldots, x_d]$, conditioned on $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_d]$, are independently Bernoulli distributed, cf. Section 2.1.3. The model is then called a *logistic regression*, with conditional expectation $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_d]$ defined by

$$\mu_i = \frac{\exp\{\theta_i\}}{1 + \exp\{\theta_i\}} \quad \text{or} \quad \theta_i = \log\left(\frac{\mu_i}{1 - \mu_i}\right) = f(\mu_i) \tag{2.4}$$

for $i = 1, \ldots, d$, where $f(\cdot)$ is often referred to as the *logit* function.

Counts are discrete nonnegative integer valued responses $\{0, 1, \dots\}$. The standard model for counts is a Poisson regression with conditional expectation

$$\mu_i = \exp\{\theta_i\} \quad \text{or} \quad \theta_i = \log(\mu_i) = f(\mu_i)$$

for $i = 1, \dots, d$. Counts have a Poisson distribution if the events being counted for a unit of time occur at a constant rate in continuous time and are mutually independent. If a count corresponds to the number of events in a given number N of trials, the count has a Binomial distribution if the events for a unit are independent and equally probable. Then, the standard model is a regression with the following conditional expectation for $i = 1, \dots, d$:

$$\mu_i = N\frac{\exp\{\theta_i\}}{1 + \exp\{\theta_i\}} \quad \text{or} \quad \theta_i = \log\left(\frac{\mu_i}{N - \mu_i}\right) = f(\mu_i).$$

2.1.3 An example: the logistic regression model

As shown in Section 2.1.2, logistic regression is a special case of GLMs, cf. equation (2.4).

"Given the data for a large set of credit card users does the usage pattern of this particular card indicate that it might have been stolen?". This research question is characterized by a response variable that is not continuous, but rather dichotomous, i.e., has only two values (0 and 1). Logistic regression analysis is specifically designed for use in such situations [44], and therefore more appropriate than linear regression. Although it is used primarily for dichotomous response variables, the technique can be extended to situations involving response variables with three or more categories (polytomous, or multinomial, response variables). It is then referred to as polytomous logistic regression.

Like linear regression, the logistic model relates explanatory, or independent, variables to a response, or dependent, variable, and the logistic model yields regression coefficients, a conditional expectation of the response variable (also referred to as the predicted value) and residuals. However, unlike linear regression,

as a special case of GLMs, the logistic model uses a functional relationship between the predictor and the conditional expectation of the response variable which is nonlinear. The logistic curve is sigmoidal, as shown in Figure 2.1. Moreover, the curve never falls below 0, or reaches above 1. Thus, the predicted values obtained using the logistic regression can always be interpreted as probabilities. In other words, in logistic regression analysis for dichotomous response variables, one attempts to predict the probability that an observation belongs to one of two groups (such as, in our credit card transaction example, the "fraudulent" and "non-fraudulent" groups). Thus, logistic regression is frequently used as a statistical classification methodology. For the vector-valued case, the conditional expectation $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_d]$, or probability that for each component the predicted value is 1, for the logistic regression model is given by

$$\mu_i = \frac{\exp\{\theta_i\}}{1 + \exp\{\theta_i\}} = \frac{1}{1 + \exp\{-\theta_i\}}, \tag{2.5}$$

with $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_d] = \mathbf{aV}$. Equation (2.5) exactly corresponds to GLMs with a Bernoulli distribution assumption expressed in equation (2.4).

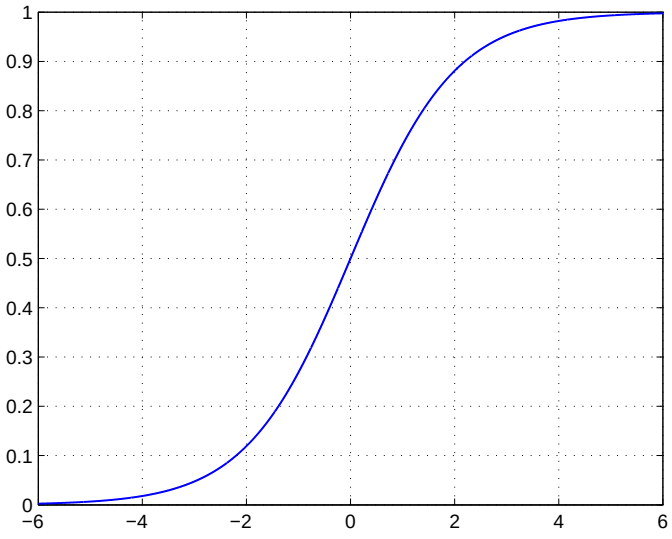

Figure 2.1 The logistic function or curve, with θ_i on the horizontal axis and $\exp\{\theta_i\}/(1 + \exp\{\theta_i\})$ on the vertical axis.

Whereas in linear regression analysis the model parameters are chosen according to the least squares criterion, in logistic regression and more generally in Generalized Linear Models, the Maximum Likelihood (ML) criterion is generally used for selecting parameter estimates.

2.1.4 Random Effect Generalized Linear Models (RE-GLMs)

The models are the same as that described for the Generalized Linear Models, i.e.,

$$f(\boldsymbol{\mu}) = \boldsymbol{\theta} = \mathbf{aV}, \tag{2.6}$$

except that now the vector $\mathbf{a}$ is assumed to be *random* and *unknown*.

2.1.5 Blind Random Effect Generalized Linear Models (BRE-GLMs)

The *Blind* Random Effect Generalized Linear Models (BRE-GLMs) differ from the RE-GLMs in that the matrix $\mathbf{V}$ is additionally assumed to be *deterministic* and *unknown*. The exponential family Principal Component Analysis method described in [10] and often referred to in Section 3 and Section 4 belongs to the large class of BRE-GLMs.

2.2 Latent variable modeling

Latent variables are widely used in different disciplines such as medicine, economics, engineering, psychology, geography, marketing and biology.

A *latent variable* is defined as a random variable whose realizations are hidden [2]. This is in contrast to response or *manifest variables* where the realizations are observed. Hence, a statistical model, i.e., simply one specifying the joint distribution of a set of random variables, becomes a latent variable model when some of these variables, i.e., the latent variables, are unobservable [1]. Variables which can be directly observed, referred to as response variables, are denoted by

$\mathbf{x} = [x_1, \ldots, x_i, \ldots, x_d]$. Latent variables are denoted by $\mathbf{a} = [a_1, \ldots, a_j, \ldots, a_q]$, where in practice q is much smaller than d. Examples in social sciences of entities which are handled as if they were measurable quantities but for which no measuring instruments exist, are business confidence, quality of life, conservatism or general intelligence.

Adopting a twofold classification of measurement levels of variables as in [1], *numerical* variables are being distinguished from *categorical* variables. Numerical variables have realized values in the set of real numbers and may be discrete or continuous. Categorical variables assign individuals to one of a set of categories and may be ordered or unordered. Then, *factor analysis* corresponds to a latent variable method with numerical response and latent variables. A latent variable method with numerical response variables and categorical latent variables is usually referred to as *latent trait analysis* whereas a latent variable method with categorical response and latent variables is referred to as *latent class analysis*, cf. Table 2.1.

Table 2.1 Types of latent variable models grouped according to whether the response and latent variables are categorical or continuous.

Response \ Latent variables	Continuous	Categorical
Continuous	Factor analysis	Latent profile analysis
Categorical	Latent trait analysis	Latent class analysis

As only the response variable $\mathbf{x}$ can be observed, any inference must be based on the joint distribution $p(\mathbf{x}, \mathbf{a})$ as follows:

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{a}) \mathrm{d}\mathbf{a} = \int p(\mathbf{x}|\mathbf{a}) \pi(\mathbf{a}) \mathrm{d}\mathbf{a}, \tag{2.7}$$

where $\pi(\mathbf{a})$ is the prior distribution of $\mathbf{a}$, $p(\mathbf{x}|\mathbf{a})$ is the conditional distribution of $\mathbf{x}$ given $\mathbf{a}$. The main interest is in what can be known about $\mathbf{a}$ after $\mathbf{x}$ has been

observed. This information is conveyed by the conditional density

$$p(\mathbf{a}|\mathbf{x}) \quad = \quad \pi(\mathbf{a})p(\mathbf{x}|\mathbf{a})/p(\mathbf{x}). \tag{2.8}$$

The *assumption of conditional independence* is an important axiom in the latent variable modeling and states the following: if the correlations among the x_i's are induced by a set of latent variables $\mathbf{a}$ then, when all a_j's are accounted for, the x_i's will be uncorrelated if all the a_j's are held fixed. In other words,

$$p(\mathbf{x}|\mathbf{a}) \quad = \quad \prod_{i=1}^{d} p_i(x_i|\mathbf{a}). \tag{2.9}$$

It is regarded as a definition of what it means to say that the latent variables $\mathbf{a}$ are *complete*. Then, equation (2.7) becomes

$$p(\mathbf{x}) \quad = \quad \int \prod_{i=1}^{d} p_i(x_i|\mathbf{a})\pi(\mathbf{a})\mathrm{d}\mathbf{a}. \tag{2.10}$$

2.2.1 Factor analysis

Factor analysis aims at identifying statistical dependencies of the response variable components in a low-dimensional representation when the response and latent variables are both continuous. It was primarily concerned with hypotheses about the organization of mental ability suggested by the examination of correlation or covariance matrices for sets of cognitive test variables [45]. For analyzing the structure of covariance or correlation matrices two methods, namely Principal Component Analysis (PCA) [46,47] and factor analysis, exist. The two approaches formally resemble each other but have rather different aims so that it is important to distinguish between them. The following Section 2.3 discusses PCA.

The aim of factor analysis is to account for the covariances of a response variable $\mathbf{x} = [x_1, \ldots, x_d]$ in terms of a small number of hypothetical variables, or factors. Put simply in correlation terms, the first question that arises is whether any correlation exists, that is whether the correlation matrix differs from the unit matrix. If there is correlation, the next question is whether there is a random variable a_1 such that all partial correlation coefficients between the response variables

after eliminating the effect of a_1 are zero. If not, then two random variables a_1 and a_2 are postulated and the partial correlation coefficients after eliminating a_1 and a_2 are examined. The process continues until all partial correlations between the response variables are zero.

In factor analysis, the basic assumption is that

$$x_i = \sum_{j=1}^{q} a_j v_i[j] + \epsilon_i \quad \text{for } i = 1, \ldots, d, \tag{2.11}$$

where a_j is the jth common factor, the number q of such factors being specified, and where ϵ_i is a residual representing sources of variation affecting only the response variable component x_i [45]. The d residual variables $\{\epsilon_i\}_{i=1}^{d}$ are assumed to be independent of one another and of the q factors $\{a_j\}_{j=1}^{q}$.

In matrix form, using a row vector notation, equations (2.11) become

$$\mathbf{x} \;=\; \mathbf{aV} + \boldsymbol{\epsilon}, \tag{2.12}$$

where $\mathbf{V}$ is a $(q \times d)$ matrix of loadings. The covariance matrix of the response variable $\mathbf{x}$ is denoted by $\boldsymbol{\Sigma}$. Assuming the factors to be uncorrelated with unit variances and the residuals diagonal covariance matrix to be $\boldsymbol{\Psi}$ yields

$$\boldsymbol{\Sigma} \;=\; \mathbf{V}'\mathbf{V} + \boldsymbol{\Psi}. \tag{2.13}$$

In practice the elements of $\mathbf{V}$ and $\boldsymbol{\Psi}$ are unknown parameters that have to be estimated from the observed data. Conventionally, the factors and the residuals follow independent Gaussian distributions with zero mean vectors, $\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$, resulting in a Gaussian distributed response variable with mean zero, $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. In order to permit the model to have a nonzero mean, a parameter vector $\mathbf{b}$ can be added, yielding

$$\mathbf{x} \;=\; \mathbf{aV} + \mathbf{b} + \boldsymbol{\epsilon}. \tag{2.14}$$

The maximum likelihood approach is used to estimate the various parameters, although because there is no closed-form solution for $\mathbf{V}$ and $\boldsymbol{\Psi}$, their values must be obtained via an iterative procedure.

It is important to note that there is an indeterminacy issue in that, with more than one factor, equations (2.11) do not by themselves determine either the factors or the parameters uniquely. For if the factors a_j are uncorrelated, they may be replaced by an orthogonal transformation of themselves, with a corresponding transformation of the loadings, while with correlated factors any nonsingular linear transformation may be made.

2.3 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a widely used technique for dimensionality reduction, lossy data compression, feature extraction and data visualization that seeks a projection, i.e., a linear method that projects the high-dimensional data onto a lower dimensional space, that best represents the data in a least-squares sense [37, 47–49]. It is of importance to notice that PCA, unlike factor analysis which focuses on correlation, searches for directions of maximum variance, i.e., directions of maximum uncertainty [46]. It is also known as the Karhunen-Loève transform.

In Principal Component Analysis, the components of an observed or outcome variable $\mathbf{x} = [x_1, \ldots, x_d]$ are transformed linearly and orthogonally into a new variable $\mathbf{a} = [a_1, \ldots, a_d]$ with an equal number of components that have the property of being uncorrelated [45]. These are chosen such that a_1 has maximum variance, a_2 has maximum variance subject to being uncorrelated with a_1, and so on. The transformation is obtained by finding the eigenvalues and eigenvectors (also called latent roots and vectors) of either the covariance or the correlation matrix of the observed variable $\mathbf{x}$. The eigenvalues, arranged in descending order of magnitude, are equal to the variances of the corresponding components of the latent variable $\mathbf{a}$, which are the unstandardized principal components. Often, the first few components account for a large proportion of the total variance of the variable $\mathbf{x}$ and may then, for certain purposes, be used to summarize the original data.

In other words, used as a dimensionality reduction technique, PCA would only retain the q first principal components, where $q < d$, i.e., the q components that account for the maximum variance in the observed variable. In general, all components are, however, needed to reproduce accurately the correlation coefficients between the observed variable components. Hence the method is not appropriate for investigating their correlation structure. When it is employed, no hypothesis needs to be made about the components of $\mathbf{x}$; they do not even need to be random variables.

It is clear from this that PCA is variance-oriented, whereas factor analysis is covariance- or correlation-oriented. The aims of the two methods can also be contrasted by considering the nature of the relationships involved. In PCA, the components a_i, $i = 1, \ldots, d$, are by definition linear functions of the observed variable components x_i, $i = 1, \ldots, d$. In factor analysis, the basic assumption is given by equation (2.11).

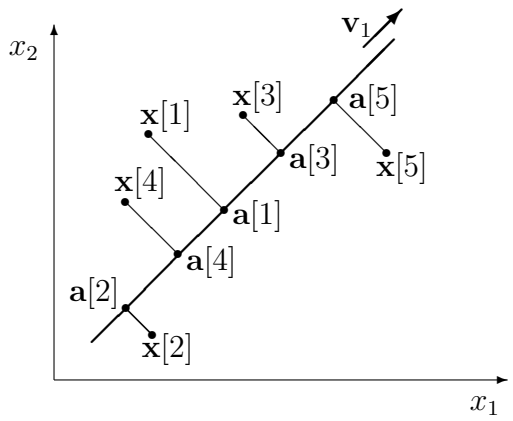

Figure 2.2 Principal Component Analysis seeks an orthogonal projection onto a subspace of lower dimensionality than that of the original data.

From a practical point of view, given a data set of observations $\{\mathbf{x}[k]\}_{k=1}^{n}$ where $\mathbf{x}[k] = \big[x_1[k], \ldots, x_d[k]\big] \in \mathbb{R}^d$, an estimate of the covariance matrix, or

scatter matrix $\mathbf{S}$, is obtained as follows:

$$\mathbf{S} \triangleq \sum_{k=1}^{n} (\mathbf{x}[k] - \overline{\mathbf{x}})^T (\mathbf{x}[k] - \overline{\mathbf{x}}), \tag{2.15}$$

where $\overline{\mathbf{x}}$ is the sample mean of the observations. An eigenvalue decomposition of the scatter matrix yields

$$\mathbf{S} = \mathbf{\Gamma} \mathbf{\Delta} \mathbf{\Gamma}^T, \tag{2.16}$$

where the diagonal matrix $\mathbf{\Delta}$ contains the eigenvalues arranged in decreasing order of magnitude, and where the ith column of $\mathbf{\Gamma}$ is the normalized eigenvector corresponding to the ith eigenvalue. The new variables are then defined by

$$\mathbf{a}[k] \triangleq \mathbf{x}[k]\mathbf{\Gamma}, \tag{2.17}$$

for $k = 1, \ldots, n$. For dimensionality reduction purposes, i.e., for a projection from an original space of dimension d into a space of lower dimension q, only the q first eigenvectors, usually referred to as the q principal axes or principal components, are used to create a $(q \times d)$ projection matrix $\mathbf{V}$. As a result, for a data set of observations $\{\mathbf{x}[k]\}_{k=1}^{n}$ where $\mathbf{x}[k] = \big[x_1[k], \ldots, x_d[k]\big] \in \mathbb{R}^d$, a new data set $\{\mathbf{a}[k]\}_{k=1}^{n}$ where $\mathbf{a}[k] = \big[a_1[k], \ldots, a_q[k]\big] \in \mathbb{R}^q$ is generated. This yields for $k = 1, \ldots, n$

$$\mathbf{x}[k] = \mathbf{a}[k]\mathbf{V}, \tag{2.18}$$

where $\mathbf{V} = [\mathbf{v}_1^T, \ldots, \mathbf{v}_q^T]^T$ is a $(q \times d)$ matrix. The process of orthogonal projection is illustrated in Figure 2.2, where the data set $\big\{\mathbf{x}[1], \ldots, \mathbf{x}[5]\big\}$ produces the set $\big\{\mathbf{a}[1], \ldots, \mathbf{a}[5]\big\}$ with a projection onto the direction given by the first principal component $\mathbf{v}_1$.

2.3.1 Probabilistic PCA

As previously acknowledged, PCA is an ubiquitous technique for data analysis and processing that is not based upon a probability model. However, there

exists a probabilistic formulation of PCA, known as *probabilistic* PCA: it can be demonstrated that the principal components of a set of observed data vectors can be determined through maximum likelihood estimation of parameters in a latent variable model closely related to factor analysis [37, 45, 50–52]. Even though the focus of factor analysis differs from the focus of PCA, i.e., a focus on covariance versus a focus on variance as demonstrated earlier, the two methods yield similar results in the special case of *isotropic* error model, where the residual variances are constrained to be equal, i.e., $\mathbf{\Psi} = \sigma^2 \mathbf{I}$ in equation (2.13).

Probabilistic PCA can be formulated by first introducing a latent variable $\mathbf{a}$ corresponding to the principal-component or low-dimensional subspace. The prior distribution over $\mathbf{a}$ is assumed to be a zero-mean unit-covariance Gaussian distribution

$$p(\mathbf{a}) = \mathcal{N}(\mathbf{a}|\mathbf{0}, \mathbf{I}).$$

Next, a Gaussian conditional distribution $p(\mathbf{x}|\mathbf{a})$ is defined over the observed variable $\mathbf{x}$ conditioned on the value of the latent variable

$$p(\mathbf{x}|\mathbf{a}) = \mathcal{N}(\mathbf{x}|\mathbf{a}\mathbf{V} + \mathbf{b}, \sigma^2 \mathbf{I}),$$

in which the mean of $\mathbf{x}$ is a general linear function of $\mathbf{a}$ governed by a $(q \times d)$ matrix $\mathbf{V}$ and the d-dimensional vector $\mathbf{b}$. The rows of $\mathbf{V}$ span a linear subspace within the data space that corresponds to the principal subspace. The other parameter in this model is the scalar σ^2 governing the variance of the conditional distribution.

Alternatively, the model can be specified by

$$\mathbf{x} = \mathbf{a}\mathbf{V} + \mathbf{b} + \boldsymbol{\epsilon},$$

where $\mathbf{a}$ is a q-dimensional Gaussian latent variable, and $\boldsymbol{\epsilon}$ is a d-dimensional zero-mean Gaussian-distributed noise variable with variance $\sigma^2 \mathbf{I}$.

The values of the parameters $\mathbf{V}$, $\mathbf{a}$ and σ^2 can be determined by using Maximum Likelihood (ML) estimation.

2.4 Discriminant analysis

Discriminant analysis as well as Principal Component Analysis (PCA) is a widely used technique for dimensionality reduction. However, whereas PCA seeks a projection that best represents the data in a least squares sense, discriminant analysis seeks a projection that best separates the data in a least squares sense [49]. There are actually two major purposes of discriminant analysis: description and prediction. *Descriptive* discriminant analysis is a statistical technique that allows one to identify variables that best discriminate members of two or more groups from one another. *Predictive* discriminant analysis allows one to predict the group membership status of observations of which the group status is unknown [44]. Hence, discriminant analysis fulfills a similar role as the one performed by logistic regression. However, discriminant analysis requires assumptions about the data that are more restrictive than those for logistic regression. For example, it requires that within each group, the components of the response variable follow a multivariate Gaussian distribution. Besides, the variance-covariance structures of the response variable components should be equal across all groups [44]. This implies that, in particular, discriminant analysis should not be used with categorical response variables.

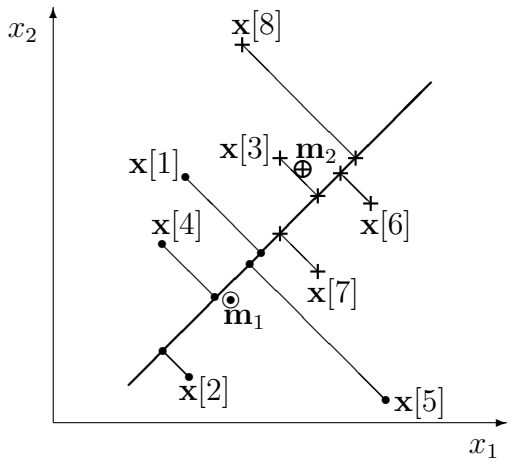

Figure 2.3 Discriminant analysis.

Discriminant analysis is performed as follows in order to separate class $\mathcal{D}_1$ from class $\mathcal{D}_2$. First, the within-class scatter matrix $\mathbf{S}_W$ is defined as follows:

$$\begin{aligned} \mathbf{S}_W &\triangleq \mathbf{S}_1 + \mathbf{S}_2, \quad \text{where} \\ \mathbf{S}_1 &\triangleq \sum_{k:\ \mathbf{x}[k]\in\mathcal{D}_1} (\mathbf{x}[k]-\mathbf{m}_1)^T(\mathbf{x}[k]-\mathbf{m}_1) \quad \text{and} \\ \mathbf{S}_2 &\triangleq \sum_{k:\ \mathbf{x}[k]\in\mathcal{D}_2} (\mathbf{x}[k]-\mathbf{m}_2)^T(\mathbf{x}[k]-\mathbf{m}_2). \end{aligned} \tag{2.19}$$

Then, the between-class scatter matrix $\mathbf{S}_B$ is defined as follows:

$$\mathbf{S}_B \triangleq (\mathbf{m}_1-\mathbf{m}_2)^T(\mathbf{m}_1-\mathbf{m}_2), \tag{2.20}$$

where $\mathbf{m}_1$, respectively $\mathbf{m}_2$, is the sample mean for data belonging to class $\mathcal{D}_1$, respectively class $\mathcal{D}_2$. Discriminant analysis aims at finding a direction of projection that maximizes the between-class scatter while minimizes the within-class scatter. The process of orthogonal projection is illustrated in Figure 2.3, where the data set $\{\mathbf{x}[1],\mathbf{x}[2],\mathbf{x}[4],\mathbf{x}[5]\}$ becomes easily separable from the data set $\{\mathbf{x}[3],\mathbf{x}[6],\mathbf{x}[7],\mathbf{x}[8]\}$. Note that for this example, PCA would have decided for a direction of projection orthogonal to the one chosen by discriminant analysis, making it impossible to separate the two data sets.

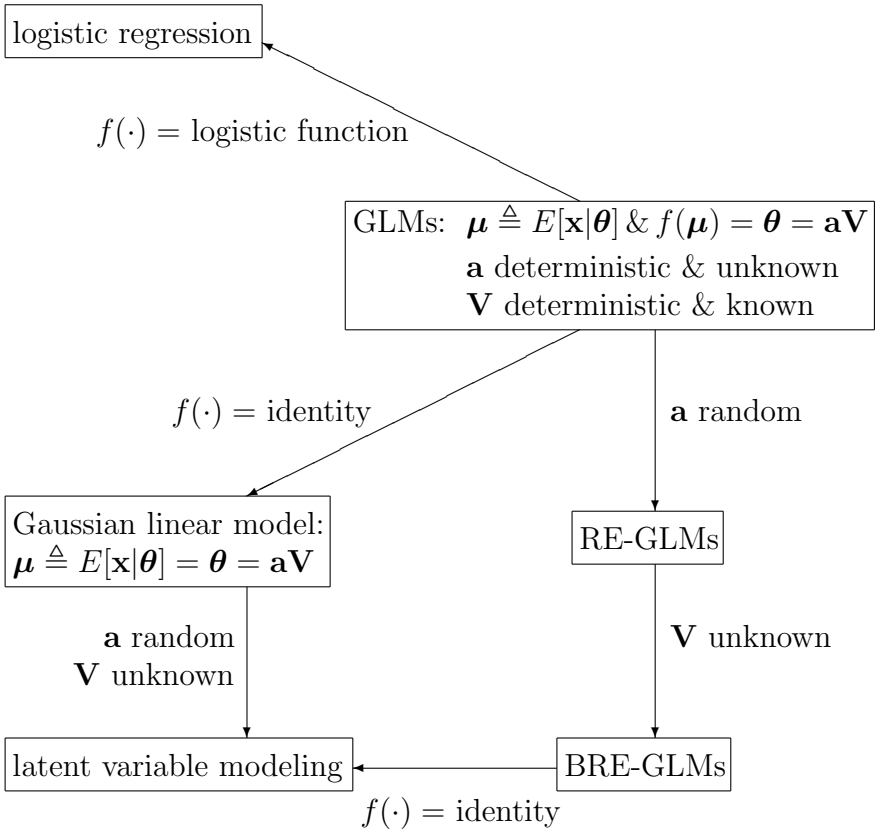

Figure 2.4 Hierarchy and relationship of ideas between well-known statistical techniques such as the standard Gaussian linear model, Generalized Linear Models (GLMs) and latent variable modeling.

3 Generalized Linear Statistics (GLS)

The proposed statistical modeling approach is a generalization and amalgamation of techniques from classical linear statistics, logistic regression, Principal Component Analysis (PCA), and Generalized Linear Models (GLMs) into a framework we refer to as *Generalized Linear Statistics* (GLS), analogously to the GLMs theory. This is actually a *nonlinear* methodology which exploits the split that occurs for exponential family distributions between the *data space* (also known as the *expected value space*) and the *parameter space* as soon as one leaves the domain of purely Gaussian random variables, cf. Appendix A. The point is that although the problem may be nonlinear in data space, it can be attacked using classical linear and other standard statistical tools applied to data that have been *mapped into the parameter space*, which is assumed to have a natural, flat Euclidean space structure. For example, one can perform regression (resulting in the technique of logistic regression and other GLMs methods [3–8]), PCA (resulting in a variety of "generalized PCA" methods [2, 9–13]), or clustering [14–16] in parameter space.

This framework is used to develop algorithms capable of classification techniques in domains involving highly heterogenous (mixed) data types, and involving labeled as well as unlabeled data sets. Specifically, this work considers mixed data-type records which have both continuous (e.g., Exponential and Gaussian) and discrete (e.g., count and binary) components. It focuses on the development of both *supervised* and *unsupervised* classification algorithms which

can be trained using labeled and unlabeled training data sets, respectively. The unsupervised case is very difficult and takes one out of the domain of the standard supervised approaches, such as neural networks and Support Vector Machines (SVMs).

3.1 Theoretical framework

The problem is abstractly stated as follows. A particular "object" of interest can be associated with a variety of descriptor random variables. Practitioners choose measurable descriptor variables that they believe are likely to be informative about interesting properties "attached to the object". These descriptors can be viewed as comprising the components of a random vector $\mathbf{x} = [x_1, \ldots, x_d] \in \mathbb{R}^d$, where the dimension d is equal to the number of descriptors. Thus the vector $\mathbf{x}$ is a point in a d-dimensional descriptor space.

Following the probabilistic Generalized Latent Variable (GLV) formalism described in [1, 2, 31], it is assumed that training descriptor space points can be drawn from populations having factorable class-conditional probability density functions of the form:

$$p(\mathbf{x}|\boldsymbol{\theta}) = p_1(x_1|\boldsymbol{\theta}) \cdot \ldots \cdot p_d(x_d|\boldsymbol{\theta}) = p_1(x_1|\theta_1) \cdot \ldots \cdot p_d(x_d|\theta_d) = \prod_{i=1}^{d} p_i(x_i|\theta_i). \quad (3.1)$$

This is referred to as the *latent variable assumption* throughout this dissertation. Delta-functions are admitted so that densities are well-defined for discrete, continuous, and mixed random variables. Note the critical assumption that the components of $\mathbf{x}$ are independent, when conditioned on the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$. It is further assumed that $\boldsymbol{\theta}$ can be written as

$$\boldsymbol{\theta} = \mathbf{aV} + \mathbf{b} \quad (3.2)$$

with $\mathbf{V} \in \mathbb{R}^{q \times d}$ and $\mathbf{b} \in \mathbb{R}^d$ deterministic, and $\mathbf{a} \in \mathbb{R}^q$. While one generally assumes $q < d$ for dimensionality reduction (and ideally $q << d$), this is strictly speaking not required. This work both considers a Bayesian approach for which

a is treated as a random vector and a classical approach where the vector **a** is deterministic. We first assume that **a** is a random vector. The randomness of **a** causes **a** to be called the *random effect*. The notation used here is motivated by the discussions in [10] and [29]. Figure 3.1 shows the geometry of the lower

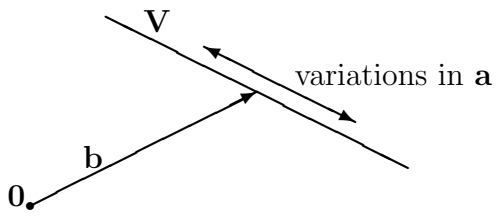

Figure 3.1 Random effect on the parameter space lower dimensional subspace.

dimensional subspace and the variations that occur due to the randomness of **a**. The matrix **V** is assumed to have full row-rank so that the relationship between **a** and $\boldsymbol{\theta}$ is one-to-one. Then, conditioning on the random vector $\boldsymbol{\theta}$ is equivalent to conditioning on the low-dimensional random vector **a**, so that

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(\mathbf{x}|\mathbf{a}) = p_1(x_1|\mathbf{a}) \cdot \ldots \cdot p_d(x_d|\mathbf{a}). \tag{3.3}$$

This is precisely the condition under which **a** is a *complete* latent variable [1]. In a probabilistic sense, all of the information that is *mutually* contained in the data vector **x** must be contained in the latent variable **a**. As noted in [1,2,53], equations (3.1) and (3.2) generalize the classical factor analysis model (as described, e.g., in [47] and [1]) to the case when the marginal densities $p_i(x_i|\theta_i)$ are non-Gaussian. Indeed, the subscript "i" on $p_i(\cdot|\cdot)$ serves to indicate that the marginal densities can all be different, allowing for the possibility of **x** containing categorical, discrete, and continuous valued components. As described below, it is further assumed that the marginal densities are each one-parameter exponential family densities, allowing the rich and powerful theory of such densities to be fruitfully exploited

[54–57], and it is commonly the case that θ_i is taken to be the so-called *natural parameter* (or some bijective function of the natural parameter) of the exponential family density $p_i(\cdot|\cdot)$. Because both the Generalized Linear Models (GLMs) and the Generalized Latent Variable (GLV) methodologies exploit the linear structure (3.2), they can be viewed as special cases of a Generalized Linear Statistics (GLS) approach to data analysis.

In the Generalized Linear Models (GLMs) theory, $\mathbf{V}$ and $\mathbf{b}$ are known and $\mathbf{a}$ is deterministic and unknown [3,4]. In both the Generalized Latent Variable (GLV) theory described in [1,2,53] and the random- and Mixed-effects Generalized Linear Models (MGLMs) literature [4,5,8,27–30,38,58–61], $\mathbf{V}$ and $\mathbf{b}$ are deterministic while $\mathbf{a}$ (and hence $\boldsymbol{\theta}$) is treated as a random vector. The difference between GLV and MGLMs is that in GLV, *all* of the quantities $\mathbf{V}$, $\mathbf{b}$, and $\mathbf{a}$ are unknown, and hence need to be identified, resulting in a so-called "blind" estimation problem, whereas in MGLMs, $\mathbf{V}$ is a known matrix of regressor variables and only the deterministic vector $\mathbf{b}$ and the unknown realizations of the *random effect* vector $\mathbf{a}$ (also known as *latent variable*) must be estimated. In both GLV and MGLMs, it is assumed that the linear relationship (3.2) holds in parameter space, and that the tools of linear and statistical inverse theory are applicable or insightful, at least conceptually. The MGLMs theory is a generalization of the classical theory of linear regression, while the GLV theory is a generalization of the classical theory of statistical factor analysis and PCA. In both cases, the generalization is based on a move from the data/description space containing the measurement vector $\mathbf{x}$ to the parameter space containing $\boldsymbol{\theta}$ via a generally nonlinear transformation known as a link function [3–6] (cf. Appendix A). It is in the latter space that the linear relationship (3.2) is assumed to hold.

Graphical models, also referred to as probabilistic graphical models, bring together graph theory and probability theory in a powerful formalism for multivariate statistical learning. Many of the classical multivariate probabilistic systems studied in statistics, information theory and pattern recognition (e.g., PCA, Inde-

pendent Component Analysis (ICA) and factor analysis) are special cases of the general graphical model formalism [37, 62–65]. As an amalgamation of such techniques, the Generalized Linear Statistics approach also is a subclass of graphical model techniques. More precisely, a graphical model consists of a collection of probability distributions that factor according to the structure of an underlying graph. A graph is formed by a collection of *nodes* (also called *vertices*) connected by a collection of *edges*. An edge consists of a pair of vertices and may either be directed or undirected. Associated with each vertex is a random variable (or a group of random variables) taking values in some set which may either be continuous or discrete. The graph then captures the way in which the joint distribution over all of the random variables can be decomposed into a product of factors each depending only on a subset of the variables. In the directed case, more popular in the machine learning community than the undirected one, the graphical model is often referred to as a Bayesian Network, and each edge is directed from parent to child with an arrow. The directed graphs that are usually considered are subject to an important restriction namely that there must be no directed cycles. In other words, there should be no closed paths within the graph such that one can move from node to node along links following the direction of the arrows and end up back at the starting node. Such graphs are also called directed acyclic graphs [37]. A directed graphical model with S nodes consists of a collection of probability distributions that factor as follows:

$$p(\mathbf{x}) = \prod_{s=1}^{S} p\big(x_s | x_{\pi(s)}\big), \tag{3.4}$$

where $\mathbf{x} = [x_1, \ldots, x_S]$, the node s represents the random variable x_s and $\pi(s)$ denotes the set of all parents of a given node s [37, 64]. Each directed graph represents a specific decomposition of a joint probability distribution into a product of conditional probabilities. Figure 3.2 presents the directed graphical model corresponding to the Generalized Linear Statistics (GLS) approach. Equation (3.3) corresponds exactly to equation (3.4). The focus of graphical models being to

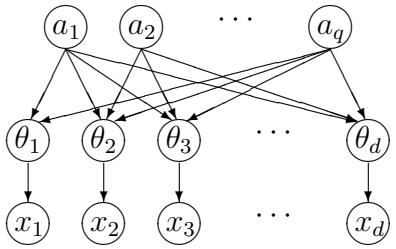

Figure 3.2 Graphical model for the Generalized Linear Statistics approach.

solve *inference* problems (e.g., how can the hidden states of a system be efficiently inferred, given partial and possibly noisy observations?) to perform *learning* tasks (e.g., how can the parameters and structure of the model be estimated?) and to construct *decision* theories, learning a GLS graphical model perfectly fits our approach.

Since $\mathbf{a}$ (and hence $\boldsymbol{\theta}$) is treated as a random vector (Bayesian approach), the (non-conditional) probability density function $p(\mathbf{x})$ requires a generally intractable integration over the parameters,

$$p(\mathbf{x}) = \int p(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} = \int \prod_{i=1}^{d} p_i(x_i|\theta_i)\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}, \tag{3.5}$$

where $\pi(\boldsymbol{\theta})$ is the probability density function of $\boldsymbol{\theta} = \mathbf{a}\mathbf{V}+\mathbf{b}$. Given the observation matrix $\mathbf{X} = \big[\mathbf{x}[1]^T, \ldots, \mathbf{x}[n]^T\big]^T \in \mathbb{R}^{n\times d}$ composed of n independent and identically distributed statistical data samples, each assumed to be stochastically equivalent to the random row vector $\mathbf{x}$, $\mathbf{x}[k] = \big[x_1[k], \ldots, x_d[k]\big] \sim \mathbf{x}$, the data likelihood function is defined as

$$p(\mathbf{X}) = \prod_{k=1}^{n} p\big(\mathbf{x}[k]\big) = \prod_{k=1}^{n} \int p\big(\mathbf{x}[k]|\boldsymbol{\theta}\big)\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \tag{3.6}$$

using the *independent and identically distributed statistical samples assumption*, and then becomes

$$p(\mathbf{X}) = \prod_{k=1}^{n} \int \prod_{i=1}^{d} p_i\big(x_i[k]|\theta_i\big)\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \tag{3.7}$$

using the *latent variable assumption*, with $\boldsymbol{\theta} = \mathbf{aV} + \mathbf{b}$. For specified exponential family densities $p_i(\cdot|\cdot), i = 1, \ldots, d$, maximum likelihood identification of the model (3.5) corresponds to identifying $\pi(\boldsymbol{\theta})$, which, under the condition $\boldsymbol{\theta} = \mathbf{aV} + \mathbf{b}$, corresponds to identifying the matrix $\mathbf{V}$, the vector $\mathbf{b}$, and a density function, $\mu(\mathbf{a})$, on the random effect $\mathbf{a}$ via a maximization of the likelihood function $p(\mathbf{X})$ with respect to $\mathbf{V}$, $\mathbf{b}$, and $\mu(\mathbf{a})$. This is generally a quite difficult problem [4, 5, 38] and it is usually attacked using approximation methods which correspond to replacing the integrals in (3.5), (3.6) and (3.7) by sums [29]:

$$p(\mathbf{x}) = \sum_{l=1}^{m} p\big(\mathbf{x}|\underline{\boldsymbol{\theta}}[l]\big)\pi_l = \sum_{l=1}^{m}\prod_{i=1}^{d} p_i\big(x_i|\underline{\theta}_i[l]\big)\pi_l, \tag{3.8}$$

$$p(\mathbf{X}) = \prod_{k=1}^{n}\sum_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_{l,k} \tag{3.9}$$

$$= \prod_{k=1}^{n}\sum_{l=1}^{m}\prod_{i=1}^{d} p_i\big(x_i[k]|\underline{\theta}_i[l]\big)\pi_{l,k} \tag{3.10}$$

over a finite number of discrete support points ("atoms") $\underline{\boldsymbol{\theta}}[l]$ (equivalently, $\underline{\mathbf{a}}[l]$) for $l = 1, \ldots, m$, $1 \le m \le n$, with point-mass probabilities

$$\pi_l \triangleq \pi\big(\boldsymbol{\theta} = \underline{\boldsymbol{\theta}}[l]\big) = \pi\big(\mathbf{a} = \underline{\mathbf{a}}[l]\big),$$
$$\pi_{l,k} \triangleq \pi\big(\boldsymbol{\theta}[k] = \underline{\boldsymbol{\theta}}[l]\big) = \pi\big(\mathbf{a}[k] = \underline{\mathbf{a}}[l]\big) = \pi_l,$$

the last equality resulting from the *independent and identically distributed statistical samples assumption*. Note that $\boldsymbol{\theta}$ and $\mathbf{a}$ are (discrete) *random variables* while $\underline{\boldsymbol{\theta}}[l]$ and $\underline{\mathbf{a}}[l], l = 1, \ldots, m$, are the m *nonrandom* support point *values* (i.e., the values of the random variables having nonzero probabilities). These m support points are shared by all the data points $\mathbf{x}[k]$, $k = 1, \ldots, n$. Also recall that taking $\pi\big(\underline{\boldsymbol{\theta}}[l]\big) = \pi\big(\underline{\mathbf{a}}[l]\big)$ for $\underline{\boldsymbol{\theta}}[l] = \underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}$ with the matrix $\mathbf{V}$ having full row-rank results in the assumption that the relationship between the discrete values $\underline{\boldsymbol{\theta}}[l]$ and $\underline{\mathbf{a}}[l]$ is one-to-one. As clearly described in [29], this approximation is justified either as a Gaussian quadrature approximation to the integral in (3.7) in the case of a Gaussian assumption for the probability density function $\pi(\boldsymbol{\theta})$ [4, 5, 38], or

by appealing to the fact that the Non-Parametric Maximum Likelihood (NPML) estimate [17, 53, 66] of the mixture density $\pi(\boldsymbol{\theta})$ yields a solution which takes a finite number of points of support [17, 20–22, 25, 26, 66], cf. Appendix C.

With $\boldsymbol{\theta} = \mathbf{a}\mathbf{V} + \mathbf{b}$, with $\mathbf{V}$, $\mathbf{b}$ fixed and $\mathbf{a}$ random, the single-sample likelihood (3.8) is equal to

$$p(\mathbf{x}) = \sum_{l=1}^{m} p\big(\mathbf{x}|\underline{\boldsymbol{\theta}}[l]\big)\pi_l = \sum_{l=1}^{m} p\big(\mathbf{x}|\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big)\pi_l, \tag{3.11}$$

and the data likelihood (3.9) is equal to

$$p(\mathbf{X}) = \prod_{k=1}^{n}\sum_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l = \prod_{k=1}^{n}\sum_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big)\pi_l. \tag{3.12}$$

The data likelihood is thus (approximately) the likelihood of a finite mixture of exponential family densities with unknown mixture proportions or point-mass probability estimates π_l and unknown point-mass support points $\underline{\mathbf{a}}[l]$, with the linear predictor $\underline{\boldsymbol{\theta}}[l] = \underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}$ in the lth mixture component [29]. In the mixture models literature, the point-mass probabilities π_l are called mixing proportions or weights, the densities $p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)$ are called the component densities of the mixture and equation (3.11) is referred to as the m-component finite mixture density [53]. The combined problem of Maximum Likelihood Estimation (MLE) of the parameters $\mathbf{V}$, $\mathbf{b}$, the point-mass support points (atoms) $\underline{\mathbf{a}}[l]$ and the point-mass probability estimates $\pi_l, l = 1, \ldots, m$, (as approximations to the unknown, and possibly continuous density $\mu(\mathbf{a})$) is known as the Semiparametric Maximum Likelihood mixture density Estimation (SMLE) problem [53, 66, 67].

This problem can be attacked by using the Expectation-Maximization (EM) algorithm [1, 15, 17, 20–22, 29, 34, 53, 68–74]. Then, the number m of distinct support point values is often strictly smaller than the number of data points n, i.e., $m < n$. Note that, historically, Laird's classic 1978 paper [17] appears to be generally acknowledged as the first paper that proposed the EM algorithm for NPML estimation in the mixture density context; then, Lindsay's classic 1983 papers [20, 21] improved upon the theoretical foundations of the NPML estimation

approach and later Mallet's 1986 paper [22] further explored some of the fundamental issues raised by Lindsay. As noted above, this problem (i.e., simultaneously identifying $\mathbf{b}$, $\underline{\mathbf{a}}[l]$, π_l for all l, and $\mathbf{V}$) is the subject matter of Generalized Latent Variable (GLV) analysis [1, 2, 53, 75]. The commonly encountered alternative problem of estimating $\mathbf{b}$, $\underline{\mathbf{a}}[l]$ and $\pi_l, l = 1, \ldots, m$, for *known* $\mathbf{V}$, where the elements of $\mathbf{V}$ are comprised of measured regressor variables, is a generalization of classical linear regression and is the subject matter of the theory of random- and Mixed-effects Generalized Linear Models (MGLMs) [3–5, 8, 27–30, 38, 58, 60, 61].

However, a classical approach to the GLS estimation problem can also be considered and the vector $\mathbf{a}$ (and hence $\boldsymbol{\theta}$) is treated as a deterministic vector. Then, to each data point $\mathbf{x}[k]$, $k = 1, \ldots, n$, corresponds a (generally different) parameter point, yielding a total of n points $\underline{\boldsymbol{\theta}}[k]$, $k = 1, \ldots, n$, in parameter space (and hence n points $\underline{\mathbf{a}}[k]$, $k = 1, \ldots, n$, in the parameter space low-dimensional subspace) as presented in the exponential family Principal Component Analysis technique [10]. The data likelihood is simply equal to

$$p(\mathbf{X}) = \prod_{k=1}^{n} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[k]\big) = \prod_{k=1}^{n} p\big(\mathbf{x}[k]|\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}\big). \tag{3.13}$$

Contrary to the Bayesian approach, no point-mass probabilities have to be estimated. For consistency of vocabulary throughout this dissertation, the points $\underline{\mathbf{a}}[k]$, $k = 1, \ldots, n$, in the parameter space low-dimensional subspace are called latent variables for both Bayesian and classical approaches. Similarly, the parameter points $\underline{\boldsymbol{\theta}}[k]$, $k = 1, \ldots, n$, are called atoms in both approaches. The classical approach can also be seen as an extreme case of the Bayesian approach for which the probability density function $\pi(\boldsymbol{\theta})$ is a delta function (one per data point) and the total number of atoms m equals the number of data points n, i.e., $m = n$. Note that while the $m < n$ parameter points of the Bayesian approach are shared by all the data points, the classical approach assigns one parameter point to each data point (hence $m = n$). This extreme case is the approach followed in Section 4. Section 5 presents a general point of view and considers and compares both

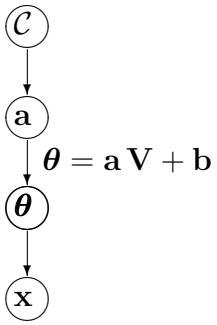

Figure 3.3 The GLS model as a Markov chain.

approaches ($m < n$ and $m = n$). Interestingly, in this classical approach, it can be shown that the latent variables $\underline{\mathbf{a}}$ are approximate sufficient statistics and provide all the information needed to make decisions on future data.

Proof. We consider the problem of classifying data point $\mathbf{x}$. The Maximum A Posteriori (MAP) estimator for the class $\mathcal{C}$ is defined as follows:

$$\widehat{\mathcal{C}}_{MAP} \triangleq \arg\max_{\mathcal{C}} p(\mathcal{C}|\mathbf{x}) = \arg\max_{\mathcal{C}} p(\mathcal{C}, \mathbf{x}). \tag{3.14}$$

Acknowledging that the GLS graphical model is similar to the Markov chain presented in Figure 3.3, we have:

$$\begin{aligned} p(\mathcal{C}|\mathbf{x})p(\mathbf{x}) = p(\mathcal{C}, \mathbf{x}) &= \int p(\mathcal{C}, \mathbf{x}, \mathbf{a})\mathrm{d}\mathbf{a} \\ &= \int p(\mathcal{C}|\mathbf{x}, \mathbf{a})p(\mathbf{x}, \mathbf{a})\mathrm{d}\mathbf{a} \end{aligned}$$

and, because of the Markov chain structure,

$$\approx \int p(\mathcal{C}|\mathbf{a})p(\mathbf{a}|\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{a}$$

and, because each data point $\mathbf{x}$ exactly corresponds to one support point $\underline{\mathbf{a}} = \underline{\mathbf{a}}(\mathbf{x})$,

$$\begin{aligned} &= \int p(\mathcal{C}|\mathbf{a})\delta(\mathbf{a} - \underline{\mathbf{a}})p(\mathbf{x})\mathrm{d}\mathbf{a} \\ &= p(\mathcal{C}|\underline{\mathbf{a}})p(\mathbf{x}). \end{aligned}$$

Hence, $p(\mathcal{C}|\underline{\mathbf{a}}) \approx p(\mathcal{C}|\mathbf{x})$ and $\underline{\mathbf{a}}$ is an approximate sufficient statistics. □

The goal of the work proposed and analyzed in this thesis is to fit an adequately faithful, class-conditional probability model of the form (3.12) for a Bayesian approach or (3.13) for a classical approach to labeled (when available) or unlabeled data to develop algorithms for making decisions on new measurements or future data. The family of models provided by (3.12) and (3.13), where the component densities are exponential family densities as described in Appendix A, is very flexible and can be used to model both labeled and unlabeled cases. For example, fraud detection considers the problem of labeling a new measurement as fraudulent or non-fraudulent. If an adequate fit of a parameterized probability distribution has only been found to the single, labeled class of non-fraudulent points, an interesting question is whether the new data point fits well with this distribution or whether it should be flagged as an outlier worthy of further scrutiny and indicating possible fraud. Alternatively, if class-conditional distributions can be fitted to both fraudulent and non-fraudulent labeled data, a Bayes-optimal likelihood ratio test can be computed. Indeed, it is well known that knowledge of class-conditional probability density functions $p(\mathbf{x}|\mathcal{C}_c)$ and the *a priori* class probabilities $p(\mathcal{C}_c)$ for classes $\mathcal{C}_c, c = 1, \ldots, K$, is sufficient for the development of Bayes-optimal classifiers that can then be applied to future data [49]. Class-conditional density-based tests can be equivalently posed as discriminant functions that are functions of sufficient statistics of the densities (when they exist) and which, in turn, define decision surfaces in feature space. Of course, the most difficult situation arises when the training samples are unlabeled. However, even in the unlabel-data case, sometimes the single-class model can still be effective for detection. For example, if the ratio of fraudulent data (say class $\mathcal{C}_2$, as measured by $p(\mathcal{C}_2)$) to non-fraudulent data (class $\mathcal{C}_1$, as measured by $p(\mathcal{C}_1)$) is very small, $p(\mathcal{C}_2)/p(\mathcal{C}_1) \ll 1$, then the unlabeled data points are approximately distributed like the non-fraudulent data, and the simpler single-class model might be effectively assumed and utilized. This condition can be satisfied in practice; for example researchers working on credit card fraud detection indicate that fraudulent credit card transactions are typically ap-

proximately one tenth of one percent of all transactions. This type of applications is called minority class detection and is presented in Section 4.4.

3.2 Component probability density models

As mentioned above, parameterized density functions of the form (3.1) whose parameter vectors $\boldsymbol{\theta} = \mathbf{aV} + \mathbf{b}$ have to be learned in a data-driven manner are being used. The approach proposed here is motivated by the exponential family Principal Component Analysis technique developed in [10]. Following the MGLMs and GLV literature cited above, it is assumed that each component density $p_i(x_i|\theta_i)$ in (3.1) for $x_i \in \mathcal{X}_i, i = 1, \ldots, d$, is a one-parameter standard exponential family density of the form:

$$p(x_i|\theta_i) = \exp\big(x_i\theta_i - G(\theta_i)\big)h(x_i), \tag{3.15}$$

cf. Appendix A. The function $G(\cdot)$ is the cumulant generating function and is defined as

$$G(\theta_i) = \log \int_{\mathcal{X}_i} \mathrm{e}^{\theta_i x_i} h(x_i)\nu(\mathrm{d}x_i),$$

where $\nu(\cdot)$ is either the Lebesgue measure or the counting measure [55, 76]. Continuous exponential family probability densities are defined with respect to the Lebesgue measure whereas discrete probability densities are defined with respect to the counting measure (further discussed in Appendix A). It can be shown, using Fubini's theorem [76], that the cumulant generating function of a parameter vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_d]$ is $G(\boldsymbol{\theta}) = \sum_{i=1}^{d} G(\theta_i)$ (further discussed in Section 4). Note, again, that the possibility of each scalar component x_i of $\mathbf{x}$ having a different exponential family distribution is allowed. This is referred to as the *mixed or hybrid exponential family distributions assumption*. Hence, it is possible to work with descriptor spaces made up of mixed types of data, such as categorical data, discrete/count data, and continuous data. Exponential family densities have many useful and important mathematical properties which can be fruitfully exploited to

obtain Maximum Likelihood Estimates (MLEs) of the parameters in the model (3.12) or (3.13) [54, 55, 57], cf. Appendix A.

3.3 Automatic feature extraction

Ideally, the fitting of parameterized probability models makes it possible to generate features, or data-patterns, which belong to a significantly lower dimensional space than the original descriptor space, but which retain all the statistical information contained in the descriptor space representation. This is a nontrivial step which can greatly improve the performance of classification algorithms and enhance the ability to interpret the results of such algorithms [77–79]. The general problem of feature selection is known to be a difficult, combinatorially-hard problem and the problem of efficient generation of informative features is of great interest [78]. This work investigates the generalization to the proposal presented in [10] that one should perform maximum likelihood estimation of the parameters of the model distribution (3.15) subject to the requirement that the parameters be constrained to a low-dimensional linear subspace in parameter space. This generalization corresponds to investigating the utility of the more general models provided by (3.12) and (3.13). Note that the dimension, q, of the low-dimensional constrained-parameter subspace is a design parameter that must be determined from empirical and model-fitting considerations.

Once the model (3.12) or (3.13) has been identified, one can then estimate $\underline{\mathbf{a}} \in \mathbb{R}^q$ for a new data measurement $\mathbf{x} \in \mathbb{R}^d$ as a vehicle for obtaining a low-dimensional feature (where ideally, $q \ll d$) which captures all of the relevant statistical information in $\mathbf{x}$ in the sense that when conditioned on $\mathbf{a}$ the probability density of $\mathbf{x}$ has the factorial form (3.3). Recall that this is precisely the requirement for the latent variable $\underline{\mathbf{a}}$ to be complete [1]. These features can be used to develop effective low-dimensional algorithms and help to gain insight into the statistical nature of the addressed classification problems.

3.4 Synthetic data generating model

In practice, producing synthetic observations from a generative model can prove informative in understanding the form of the probability distribution represented by that model [37]. Also, a side benefit of successfully fitting a faithful probability model (3.12) or (3.13) to the data is the ability to generate synthetic data in sufficient quantity that meaningful Monte-Carlo simulations and statistical tests on any proposed algorithm can be performed. This is especially important when data are scarce or expensive to get.

Acknowledgement

Chapter 3, in part, is a reprint of the material as it appears in "Data-pattern discovery methods for detection in nongaussian high-dimensional data sets," C. Levasseur, K. Kreutz-Delgado, U. Mayer and G. Gancarz, in *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, pp. 545–549, Nov. 2005, "Generalized statistical methods for unsupervised minority class detection in mixed data sets," C. Levasseur, U. F. Mayer, B. Burdge and K. Kreutz-Delgado, in *Proceedings of the First IAPR Workshop on Cognitive Information Processing (CIP)*, pp. 126–131, June 2008 and "Generalized statistical methods for mixed exponential families, part I: theoretical foundations," C. Levasseur, K. Kreutz-Delgado and U. F. Mayer, submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Sept. 2009. The dissertation author was the primary author of these papers.

4 Learning at one extreme of the GLS framework

This section focuses on the estimation procedure of Generalized Linear Statistics (GLS) framework components for the extreme case where the number of parameter points equals the number of data points $m = n$. This extreme case is similar to the exponential family Principal Component Analysis technique proposed in [10] (further discussed in Section 5). We interpret it as a form of Principal Component Analysis (PCA) performed in parameter space instead of data space as in classical PCA. Therefore, we often compare the consequences of decision making in parameter space based on GLS modeling with the consequences of decision making in data space based on information from classical PCA and other classical linear techniques.

This section is organized as follows. The estimation procedure is considered first for a single exponential family, and subsequently for data of mixed types where several exponential families are involved. The loss function is defined and its convexity properties studied; an iterative minimization algorithm is derived in detail. The angle between subspaces is defined as a measure of the estimation performance. A set of constraints and penalty functions are then designed and added to the loss function in order to address the following concerns: divergence at infinity, positivity of the natural parameter, matrix identifiability problems. As noted in [10], it is possible for the atoms to diverge since the optimum may be at infinity. To avoid such a behavior a penalty function is introduced; this defines

and places a set of constraints into the loss function via a penalty parameter in a way that penalizes any divergence to infinity. Also, for several exponential family distributions with natural restrictions on their parameter, an additional positivity constraint on the natural parameter values has to be taken into account. For example, the natural parameter of the Gamma distribution is the negative of the inverse of the scale parameter, which is assumed to be strictly positive, cf. Appendix A. As a result, the natural parameter must be constrained to be strictly negative. Several approaches that impose such constraints are studied. A key assumption of the GLS framework is that the parameters are restricted to a low-dimensional subspace. However, an orthonormality constraint is needed for the matrix that defines this low-dimensional parameter subspace. It can be shown that otherwise the matrix is not unique and that other equivalent representations can be derived by orthogonal transformations of it [45]. The orthonormality constraint reduces the impact of the identifiability problem. For both the single exponential family case and the mixed data-type case, synthetic data examples provide insights into the relationship between the low-dimensional parameter subspace originally used to create the synthetic data sets and the subspace estimated within the GLS framework. Finally, insights about the underlying statistical structure of complex data sets gained by GLS modeling are utilized in a problem of unsupervised minority class detection. Minority class detection aims to differentiate rare key events belonging to a "minority class" from the remainder of the data belonging to a "majority class". An unsupervised minority class detection algorithm performed in parameter space rather than in data space as in more classical approaches is proposed and tested on a synthetic data example.

4.1 Problem description

Following the Generalized Linear Statistics (GLS) framework presentation in Section 3, the special case where the number of parameter points equals

the number of data points, i.e., $m = n$, is solely considered. Hence, the point-mass probabilities do not need to be estimated and the Expectation-Maximization (EM) algorithm is unnecessary. To each vector $\mathbf{x}$ corresponds a single vector $\underline{\mathbf{a}}$, i.e., a single vector $\underline{\boldsymbol{\theta}}$, and they all share a common index $k = 1, \ldots, n$.

Let $\mathbf{X}$ be the $(n \times d)$ matrix of observations. The dimension of the data space is referred to as d and the number of points in the data set is referred to as n. The k'th row of the matrix $\mathbf{X}$ is the data row vector $\mathbf{x}[k]$. The observations can also be referred to as the data set $\big\{\mathbf{x}[k]\big\}_{k=1}^{n}$, where $\mathbf{x}[k] = \big[x_1[k], x_2[k], \ldots, x_d[k]\big]$. Consequently, the observation matrix is denoted as follows:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}[1] \\ \mathbf{x}[2] \\ \vdots \\ \mathbf{x}[n] \end{pmatrix} = \begin{pmatrix} x_1[1] & \ldots & x_d[1] \\ x_1[2] & \ldots & x_d[2] \\ \vdots & \ddots & \vdots \\ x_1[n] & \ldots & x_d[n] \end{pmatrix}.$$

The proposed algorithm aims to identify the set of parameters $\big\{\underline{\boldsymbol{\theta}}[k]\big\}_{k=1}^{n}$, where each $\underline{\boldsymbol{\theta}}[k]$ is the "projection" of a corresponding $\mathbf{x}[k]$ onto a lower dimensional subspace of the parameter space. The dimension of this lower dimensional subspace is referred to as q, where $q < d$, ideally $q \ll d$, and its basis is defined as $\{\mathbf{v}_j\}_{j=1}^{q}$ where $\mathbf{v}_j = [v_{j1}, v_{j2}, \ldots, v_{jd}]$. Hence, the matrix $\mathbf{V}$ defined by

$$\mathbf{V} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_q \end{pmatrix} = \begin{pmatrix} v_{11} & \ldots & v_{1d} \\ v_{21} & \ldots & v_{2d} \\ \vdots & \ddots & \vdots \\ v_{q1} & \ldots & v_{qd} \end{pmatrix}$$

is $(q \times d)$ and identifies the lower dimensional subspace of the parameter space. The latent variable matrix $\underline{\mathbf{A}}$ is $(n \times q)$ and represents the coordinates of each $\underline{\boldsymbol{\theta}}[k]$, $k = 1, \ldots, n$, in this lower dimensional subspace:

$$\underline{\mathbf{A}} = \begin{pmatrix} \underline{\mathbf{a}}[1] \\ \underline{\mathbf{a}}[2] \\ \vdots \\ \underline{\mathbf{a}}[n] \end{pmatrix} = \begin{pmatrix} \underline{a}_1[1] & \ldots & \underline{a}_q[1] \\ \underline{a}_1[2] & \ldots & \underline{a}_q[2] \\ \vdots & \ddots & \vdots \\ \underline{a}_1[n] & \ldots & \underline{a}_q[n] \end{pmatrix}.$$

Therefore, each $\underline{\boldsymbol{\theta}}[k]$, $k = 1, \ldots, n$, can be represented as a linear combination of the basis vectors plus a d-dimensional offset or displacement vector $\mathbf{b}$ as follows:

$$\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b} = \sum_{j=1}^{q} \underline{a}_j[k]\mathbf{v}_j + \mathbf{b}, \tag{4.1}$$

with $\mathbf{b} = [b_1, \ldots, b_d]$. In other words, the proposed algorithm aims to find the $\underline{\boldsymbol{\theta}}[k]$ that is "best" in parameter space for its corresponding $\mathbf{x}[k]$, for all $k = 1, \ldots, n$.

The matrix $\underline{\boldsymbol{\Theta}}$ is of the same dimensions as the observation matrix, namely $(n \times d)$:

$$\underline{\boldsymbol{\Theta}} = \underline{\mathbf{A}}\mathbf{V} + \mathbf{B} = \begin{pmatrix} \underline{\boldsymbol{\theta}}[1] \\ \underline{\boldsymbol{\theta}}[2] \\ \vdots \\ \underline{\boldsymbol{\theta}}[n] \end{pmatrix} = \begin{pmatrix} \underline{\theta}_1[1] & \ldots & \underline{\theta}_d[1] \\ \underline{\theta}_1[2] & \ldots & \underline{\theta}_d[2] \\ \vdots & \ddots & \vdots \\ \underline{\theta}_1[n] & \ldots & \underline{\theta}_d[n] \end{pmatrix},$$

where the offset matrix $\mathbf{B}$ is $(n \times d)$ and simply composed of n identical rows $\mathbf{b}$:

$$\mathbf{B} = \begin{pmatrix} \mathbf{b} \\ \mathbf{b} \\ \vdots \\ \mathbf{b} \end{pmatrix} = \begin{pmatrix} b_1 & \ldots & b_d \\ b_1 & \ldots & b_d \\ \vdots & \ddots & \vdots \\ b_1 & \ldots & b_d \end{pmatrix}.$$

Assuming a Maximum Likelihood (ML) estimation path as traditionally used in the GLMs literature [6], the loss function is the negative log-likelihood function and is given by:

$$L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) = -\log p(\mathbf{X}|\underline{\boldsymbol{\Theta}}),$$

subject to the constraint $\underline{\boldsymbol{\Theta}} = \underline{\mathbf{A}}\mathbf{V} + \mathbf{B}$. Without loss of generality, the offset term $\mathbf{b}$ could be absorbed in the standard manner into the matrix $\mathbf{V}$ using homogenous coordinates as in Section 2. However, for the sake of completeness, its presence remains throughout the remainder of this dissertation.

In accordance with the Generalized Linear Statistics framework, the following assumptions are made:

(i) *the independent and identically distributed statistical samples assumption*: the samples $\mathbf{x}[k]$, $k = 1, \ldots, n$, are drawn independently and identically;

(ii) *the latent variable assumption*: the components $x_i[k]$, $i = 1, \ldots, d$, are independent when conditioned on the parameter vector $\underline{\boldsymbol{\theta}}[k]$, i.e., $p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[k]\big) = p_1\big(x_1[k]|\underline{\theta}_1[k]\big) \cdot \ldots \cdot p_d\big(x_d[k]|\underline{\theta}_d[k]\big)$ for all k, $k = 1, \ldots, n$;

(iii) *the mixed or hybrid exponential family distributions assumption*: each density function $p_i\big(x_i[k]|\underline{\theta}_i[k]\big)$ is any one-parameter exponential family distribution with $\underline{\theta}_i[k]$ taken to be the natural parameter of the exponential family density or some simple function of it. This assumption allows the rich and powerful theory of exponential family distributions to be fruitfully utilized.

The marginal densities $p_i(\cdot|\cdot)$ can all be different, allowing for the possibility of $\mathbf{x}[k]$ containing continuous and discrete valued components. Consequently, the loss function becomes:

$$L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) = -\log p(\mathbf{X}|\underline{\boldsymbol{\Theta}}) \qquad \textit{matrix distribution}, \tag{4.2}$$

$$= -\sum_{k=1}^{n} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[k]\big) \qquad \textit{vector distribution}, \tag{4.3}$$

$$= -\sum_{k=1}^{n}\sum_{i=1}^{d} \log p_i\big(x_i[k]|\underline{\theta}_i[k]\big) \qquad \textit{scalar distribution}. \tag{4.4}$$

A distribution is said to be a member of the exponential family if it has a density function of the form

$$p\big(x_i[k]|\underline{\theta}_i[k]\big) = \exp\Big(x_i[k]\underline{\theta}_i[k] - G\big(\underline{\theta}_i[k]\big)\Big),$$

where the function $G(\cdot)$ is the cumulant generating function, defined as

$$G(\underline{\theta}_i) = \log \int_{\mathcal{X}_i} \mathrm{e}^{\theta_i x_i} \nu(\mathrm{d}x_i),$$

and where $\nu(\cdot)$ is a σ-finite measure that generates the exponential family (for the details cf. Appendix A). The gradient of the cumulant generating function is denoted by $g(\cdot)$ and is referred to as the link function. By exploiting the previously

stated *latent variable assumption*, it can be shown that if $p_i(\cdot|\cdot)$ is the 1-dimensional conditional distribution of the component $x_i[k]$, $i = 1, \ldots, d$, of the data point $\mathbf{x}[k]$ given the parameter $\underline{\theta}_i[k]$, then the vector $\mathbf{x}[k]$ follows a d-dimensional conditional exponential distribution $p(\cdot|\cdot)$ given the vector parameter $\underline{\boldsymbol{\theta}}[k]$.

Proof. Considering the most general case, each component x_i for $i = 1, \ldots, d$ is assumed to be exponentially distributed according to the distribution p_i with parameter $\underline{\theta}_i$, and the components are independent when conditioned on theirs parameters. Following the definition of standard exponential families presented in Appendix A.2, a σ-finite measure ν_i is assumed for each distribution, $i = 1, \ldots, d$. Let $\nu = (\nu_1, \ldots, \nu_d)$ with $\nu(\mathrm{d}\mathbf{x}) = \nu_1(\mathrm{d}x_1) \cdot \nu_2(\mathrm{d}x_2) \cdot \ldots \cdot \nu_d(\mathrm{d}x_d)$ be the σ-finite measure in the d-dimensional data space. The 1-dimensional distribution can be written as $p_i(x_i|\underline{\theta}_i) = \exp\left\{\underline{\theta}_i x_i - G_i(\underline{\theta}_i)\right\}$. Based on the *latent variable assumption*, the distribution of the vector $\mathbf{x}$ can be written as follows:

$$p(\mathbf{x}|\underline{\boldsymbol{\theta}}) = \prod_{i=1}^{d} p_i(x_i|\underline{\theta}_i) = \prod_{i=1}^{d} \mathrm{e}^{\underline{\theta}_i x_i - G_i(\underline{\theta}_i)} = \mathrm{e}^{\underline{\boldsymbol{\theta}} \cdot \mathbf{x} - \sum_{i=1}^{d} G_i(\underline{\theta}_i)},$$

where, by definition of an exponential family distribution and using Fubini's theorem [76],

$$\begin{aligned}
\sum_{i=1}^{d} G_i(\underline{\theta}_i) &= \sum_{i=1}^{d} \log \int_{\mathcal{X}_i} \mathrm{e}^{\underline{\theta}_i x_i} \nu_i(\mathrm{d}x_i) = \log \prod_{i=1}^{d} \int_{\mathcal{X}_i} \mathrm{e}^{\underline{\theta}_i x_i} \nu_i(\mathrm{d}x_i) \\
&= \log \left\{ \int_{\mathcal{X}_1} \mathrm{e}^{\underline{\theta}_1 x_1} \nu_1(\mathrm{d}x_1) \cdot \int_{\mathcal{X}_2} \mathrm{e}^{\underline{\theta}_2 x_2} \nu_2(\mathrm{d}x_2) \cdot \ldots \cdot \int_{\mathcal{X}_d} \mathrm{e}^{\underline{\theta}_d x_d} \nu_d(\mathrm{d}x_d) \right\} \\
&= \log \int_{\mathcal{X}_1} \cdots \int_{\mathcal{X}_d} \mathrm{e}^{\underline{\theta}_1 x_1 + \underline{\theta}_2 x_2 + \cdots + \underline{\theta}_d x_d} \nu_1(\mathrm{d}x_1) \cdot \ldots \cdot \nu_d(\mathrm{d}x_d) \\
&= \log \int_{\mathcal{X}} \mathrm{e}^{\underline{\boldsymbol{\theta}} \mathbf{x}} \nu(\mathrm{d}\mathbf{x}) = G(\underline{\boldsymbol{\theta}}),
\end{aligned} \tag{4.5}$$

where $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_d$ defines the (product) space of the d-dimensional vector $\mathbf{x}$. As a result, $G(\underline{\boldsymbol{\theta}}) = \sum_{i=1}^{d} G_i(\underline{\theta}_i)$ is the cumulant generating function of the multivariate exponential family distribution $p(\mathbf{x}|\underline{\boldsymbol{\theta}}) = \prod_{i=1}^{d} p_i(x_i|\underline{\theta}_i)$. □

4.2 Estimation procedures for a single exponential family

First, the case of a single common exponential family, i.e., $p_i(\cdot|\cdot) = p(\cdot|\cdot)$ for all $i = 1, \ldots, d$, is considered.

4.2.1 Loss function and convexity

The loss function is given by

$$L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) = -\sum_{k=1}^{n}\sum_{i=1}^{d} \log p\big(x_i[k]|\underline{\theta}_i[k]\big), \tag{4.6}$$

where $p(\cdot|\cdot)$ is an exponential family distribution with parameter $\underline{\theta}_i[k]$ satisfying the constraint $\underline{\theta}_i[k] = \sum_{j=1}^{q} \underline{a}_j[k]v_{ji} + b_i$. Also, using the definition of an exponential family distribution, the loss function can be written as:

$$L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) = -\sum_{k=1}^{n}\sum_{i=1}^{d} \Big\{ x_i[k]\underline{\theta}_i[k] - G\big(\underline{\theta}_i[k]\big) \Big\},$$

and

$$\begin{aligned}
\underset{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}}{\arg\min}\, L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) &= \underset{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}}{\arg\min} -\sum_{k=1}^{n}\sum_{i=1}^{d} \Big\{ x_i[k]\underline{\theta}_i[k] - G\big(\underline{\theta}_i[k]\big) \Big\} \qquad (4.7)\\
&= \underset{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}}{\arg\min} \sum_{k=1}^{n}\sum_{i=1}^{d} \Big\{ G\big(\underline{\theta}_i[k]\big) - x_i[k]\underline{\theta}_i[k] \Big\}\\
&= \underset{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}}{\arg\min} \sum_{k=1}^{n} \left\{ \sum_{i=1}^{d} G\big(\underline{\theta}_i[k]\big) - \sum_{i=1}^{d} x_i[k]\underline{\theta}_i[k] \right\}\\
&= \underset{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}}{\arg\min} \sum_{k=1}^{n} \Big\{ G\big(\underline{\boldsymbol{\theta}}[k]\big) - \underline{\boldsymbol{\theta}}[k]\mathbf{x}[k]^T \Big\}\\
&= \underset{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}}{\arg\min} \sum_{k=1}^{n} \Big\{ G\big(\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}\big) - \big(\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}\big)\mathbf{x}[k]^T \Big\}. \qquad (4.8)
\end{aligned}$$

Alternatively, Appendix A shows that the negative log-likelihood of the density of an exponential family distribution $p(x_i[k]|\underline{\theta}_i[k])$ can be expressed through a Bregman distance $B_F(\cdot\|\cdot)$:

$$-\log p(x_i[k]|\underline{\theta}_i[k]) = B_F\big(x_i[k]\big\|g(\underline{\theta}_i[k])\big) - F\big(x_i[k]\big),$$

where $F(\cdot)$ is the Fenchel conjugate of the cumulant generating function $G(\cdot)$. Then, the loss function in (4.6) becomes:

$$\begin{aligned} L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) &= \sum_{k=1}^{n}\sum_{i=1}^{d}\Big\{B_F\big(x_i[k]\big\|g(\underline{\theta}_i[k])\big) - F\big(x_i[k]\big)\Big\} \\ &= \sum_{k=1}^{n}\sum_{i=1}^{d}\Big\{B_F\big(x_i[k]\big\|g(\underline{\theta}_i[k])\big)\Big\} - \underline{\sum_{k=1}^{n}\sum_{i=1}^{d}F\big(x_i[k]\big)}, \end{aligned}$$

where $\underline{\theta}_i[k] = \sum_{j=1}^{q}\underline{a}_j[k]v_{ji} + b_i$. As shown in Appendix A.3, the underlined term in the above equation does not depend on either $\underline{\mathbf{A}}$, $\mathbf{V}$ or $\mathbf{b}$, resulting in the following minimization problem:

$$\begin{aligned} \underset{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}}{\arg\min}\, L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) &= \underset{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}}{\arg\min}\sum_{k=1}^{n}\sum_{i=1}^{d}B_F\big(x_i[k]\big\|g(\underline{\theta}_i[k])\big) \\ &= \underset{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}}{\arg\min}\sum_{k=1}^{n}\sum_{i=1}^{d}B_F\left(x_i[k]\Big\|g\Big(\sum_{j=1}^{q}\underline{a}_j[k]v_{ji} + b_i\Big)\right). \end{aligned} \tag{4.9}$$

It can be shown that the loss function is convex in either of its arguments with the others fixed. Indeed, the dual divergences property of the Bregman distance presented in Appendix A.3.2 implies that, if $G\big(\underline{\theta}_i[k]\big)$ is strictly convex, then

$$\begin{aligned} B_F\big(x_i[k]\big\|g(\underline{\theta}_i[k])\big) &= B_G\Big(f\big(g(\underline{\theta}_i[k])\big)\big\|f\big(x_i[k]\big)\Big) \\ &= B_G\big(\underline{\theta}_i[k]\big\|f(x_i[k])\big), \end{aligned} \tag{4.10}$$

since the function $f(\cdot)$ is the inverse of the link function $g(\cdot)$. The fact that $f(\cdot)$ and $g(\cdot)$ are inverse functions of each other is easily shown and explained in Appendix A.2. Using equation (4.10) in equation (4.9), the minimization problem becomes, if $G\big(\underline{\theta}_i[k]\big)$ is strictly convex:

$$\begin{aligned} \underset{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}}{\arg\min}\, L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) &= \underset{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}}{\arg\min}\sum_{k=1}^{n}\sum_{i=1}^{d}B_G\big(\underline{\theta}_i[k]\big\|f(x_i[k])\big) \\ &= \underset{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}}{\arg\min}\sum_{k=1}^{n}\sum_{i=1}^{d}B_G\big(\sum_{j=1}^{q}\underline{a}_j[k]v_{ji} + b_i\big\|f(x_i[k])\big). \end{aligned} \tag{4.11}$$

This is a critical step of GLS because the minimization problem is moved from data space fully into parameter space.

It is well-known that $G\big(\underline{\theta}_i[k]\big)$ is a convex function on $\underline{\theta}_i[k]$ and strictly convex if the exponential family is minimal [55]. The convexity property in Appendix A.3.2 states that $B_G(\cdot\|\cdot)$ is always convex in the first argument, resulting in the fact that $B_G\big(\underline{\theta}_i[k]\big\|f(x_i[k])\big)$ is convex in $\underline{\theta}_i[k]$ for all $k = 1, \ldots, n$ and $i = 1, \ldots, d$. Then, since $\underline{\theta}_i[k] = \sum_{j=1}^{q} \underline{a}_j[k]v_{ji} + b_i$ is a convex relationship in either $a_j[k], j = 1, \ldots, q$, $v_{ji}, j = 1, \ldots, q$, or b_i with the others fixed, for all $k = 1, \ldots, n$ and $i = 1, \ldots, d$, the Bregman distance $B_G\big(\sum_{j=1}^{q} \underline{a}_j[k]v_{ji} + b_i\big\|f(x_i[k])\big)$ is convex in any of the three arguments when the other two are fixed. Therefore, as a sum of convex functions, the loss function is convex in either of its arguments with the others fixed, i.e., the loss function is convex in $\underline{\mathbf{\Theta}} = \underline{\mathbf{A}}\mathbf{V} + \mathbf{B}$.

The strict convexity of the function $G(\cdot)$ for common one-dimensional exponential family distributions [80] is shown below:

(i) Gaussian with unit-variance:

$$G(\underline{\theta}_i) = \frac{\underline{\theta}_i^2}{2}, \quad \frac{\mathrm{d}G(\underline{\theta}_i)}{\mathrm{d}\underline{\theta}_i} = G'(\underline{\theta}_i) = \underline{\theta}_i, \quad \frac{\mathrm{d}^2G(\underline{\theta}_i)}{\mathrm{d}\underline{\theta}_i^2} = G''(\underline{\theta}_i) = 1 > 0.$$

(ii) Exponential:

$$G(\underline{\theta}_i) = -\log(-\underline{\theta}_i), \quad G'(\underline{\theta}_i) = \frac{-1}{\underline{\theta}_i}, \quad G''(\underline{\theta}_i) = \frac{1}{\underline{\theta}_i^2} > 0.$$

(iii) Bernoulli:

$$G(\underline{\theta}_i) = \log(1 + \mathrm{e}^{\underline{\theta}_i}), \quad G'(\underline{\theta}_i) = \frac{\mathrm{e}^{\underline{\theta}_i}}{1 + \mathrm{e}^{\underline{\theta}_i}}, \quad G''(\underline{\theta}_i) = \frac{\mathrm{e}^{\underline{\theta}_i}}{(1 + \mathrm{e}^{\underline{\theta}_i})^2} > 0.$$

(iv) Binomial:

$$G(\underline{\theta}_i) = N\log(1 + \mathrm{e}^{\underline{\theta}_i}), \quad G'(\underline{\theta}_i) = N\frac{e^{\underline{\theta}_i}}{1 + \mathrm{e}^{\underline{\theta}_i}}, \quad G''(\underline{\theta}_i) = N\frac{\mathrm{e}^{\underline{\theta}_i}}{(1 + \mathrm{e}^{\underline{\theta}_i})^2} > 0.$$

(v) Poisson:

$$G(\underline{\theta}_i) = \mathrm{e}^{\underline{\theta}_i}, \quad G'(\underline{\theta}_i) = \mathrm{e}^{\underline{\theta}_i}, \quad G''(\underline{\theta}_i) = \mathrm{e}^{\underline{\theta}_i} > 0.$$

Since the loss function is convex in either of its arguments with the others fixed, its minimization can be attacked by using an iterative approach. Then, the first step, given a fixed matrix $\mathbf{V}$ and a fixed vector $\mathbf{b}$, is to obtain the matrix $\underline{\mathbf{A}}$ or the set of vectors $\underline{\mathbf{a}}[k]$ for $k = 1, \ldots, n$, that minimizes the loss function

$$L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) = \sum_{k=1}^{n} \Big\{ G\big(\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}\big) - \big(\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}\big)\mathbf{x}[k]^T \Big\}. \tag{4.12}$$

The second step, given a fixed matrix $\underline{\mathbf{A}}$ and a fixed vector $\mathbf{b}$, is to obtain the matrix $\mathbf{V}$ or the set of vectors $\mathbf{v}_j$ for $j = 1, \ldots, q$, that minimizes the loss function (4.12). The last step, given a fixed matrix $\underline{\mathbf{A}}$ and a fixed matrix $\mathbf{V}$, is to obtain the vector $\mathbf{b}$ that minimizes the loss function (4.12).

4.2.2 Iterative minimization of the loss function

The classical Newton-Raphson method is used for the iterative minimization of the loss function (4.12), cf. Appendix B.

The first step in the $(t+1)^{\text{st}}$ iteration consists of the update $\underline{\mathbf{A}}^{(t+1)} = \arg\min_{\underline{\mathbf{A}}} L(\underline{\mathbf{A}}, \mathbf{V}^{(t)}, \mathbf{b}^{(t)})$, with $\underline{\mathbf{A}}^{(t)}$, $\mathbf{V}^{(t)}$ and $\mathbf{b}^{(t)}$ being the updates obtained at the end of the t^{th} iteration. It then requires the computation of the gradient vector $\nabla_{\underline{\mathbf{a}}} l\big(\underline{\mathbf{a}}[k]\big)$ and the Hessian matrix $\nabla^2_{\underline{\mathbf{a}}} l\big(\underline{\mathbf{a}}[k]\big)$ of the loss function $l\big(\underline{\mathbf{a}}[k]\big)$ with respect to the vector $\underline{\mathbf{a}}[k]$, for all $k = 1, \ldots, n$, where $l\big(\underline{\mathbf{a}}[k]\big) = l\big(\underline{\mathbf{a}}[k], \mathbf{V}^{(t)}, \mathbf{b}^{(t)}\big)$ collects the elements of the loss function $L(\underline{\mathbf{A}}, \mathbf{V}^{(t)}, \mathbf{b}^{(t)})$ that depend only on the vector $\underline{\mathbf{a}}[k]$:

$$\begin{aligned} l\big(\underline{\mathbf{a}}[k]\big) &= G\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)\mathbf{x}[k]^T \\ &= \sum_{i=1}^{d} \left\{ G\left(\sum_{j=1}^{q} \underline{a}_j[k] v_{ji}^{(t)} + b_i^{(t)} \right) - x_i[k] \left(\sum_{j=1}^{q} \underline{a}_j[k] v_{ji}^{(t)} + b_i^{(t)} \right) \right\}. \end{aligned} \tag{4.13}$$

The gradient vector of the loss function $l\big(\underline{\mathbf{a}}[k]\big)$ with respect to the vector $\underline{\mathbf{a}}[k]$, for $k = 1, \ldots, n$, is given by

$$\begin{aligned} \nabla_{\underline{\mathbf{a}}} l\big(\underline{\mathbf{a}}[k]\big) = \frac{\partial l\big(\underline{\mathbf{a}}[k]\big)}{\partial \underline{\mathbf{a}}[k]} &= \mathbf{V}^{(t)} G'\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{V}^{(t)}\mathbf{x}[k]^T \\ &= \mathbf{V}^{(t)} \Big(G'\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T \Big), \end{aligned}$$

where

$$G'\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)=\left.\frac{\partial G\big(\underline{\boldsymbol{\theta}}[k]\big)}{\partial\underline{\boldsymbol{\theta}}[k]}\right|_{\underline{\boldsymbol{\theta}}[k]=\underline{\mathbf{a}}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}}\quad\text{and}\quad\frac{\partial\underline{\boldsymbol{\theta}}[k]}{\partial\underline{\mathbf{a}}[k]}=\mathbf{V}^{(t)}.$$

Here, the following convention for derivatives with respect to a row vector is used: for $\underline{\mathbf{a}}[k]$, a $(1\times q)$ vector, and $l(\cdot)$, a scalar function of $\underline{\mathbf{a}}[k]$, $\partial l\big(\underline{\mathbf{a}}[k]\big)/\partial\underline{\mathbf{a}}[k]=\big[\partial l\big(\underline{\mathbf{a}}[k]\big)/\partial\underline{a}_1[k],\ldots,\partial l\big(\underline{\mathbf{a}}[k]\big)/\partial\underline{a}_q[k]\big]^T$ is a $(q\times 1)$ vector. Similarly, for $\underline{\boldsymbol{\theta}}[k]$, a $(1\times d)$ vector, and $G(\cdot)$, a scalar function of $\underline{\boldsymbol{\theta}}[k]$, $\partial G\big(\underline{\boldsymbol{\theta}}[k]\big)/\partial\underline{\boldsymbol{\theta}}[k]$ is a $(d\times 1)$ vector as follows: $\partial G\big(\underline{\boldsymbol{\theta}}[k]\big)/\partial\underline{\boldsymbol{\theta}}[k]=\big[\partial G\big(\underline{\boldsymbol{\theta}}[k]\big)/\partial\underline{\theta}_1[k],\ldots,\partial G\big(\underline{\boldsymbol{\theta}}[k]\big)/\partial\underline{\theta}_d[k]\big]^T$. Then,

$$\begin{aligned}G'\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)&=\left.\left[\frac{\partial G\big(\underline{\boldsymbol{\theta}}[k]\big)}{\partial\underline{\theta}_1[k]},\ldots,\frac{\partial G\big(\underline{\boldsymbol{\theta}}[k]\big)}{\partial\underline{\theta}_d[k]}\right]\right|^T_{\underline{\boldsymbol{\theta}}[k]=\underline{\mathbf{a}}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}}\\&=\left[\frac{\partial}{\partial\underline{\theta}_1[k]}\sum_{i=1}^{d}G\big(\underline{\theta}_i[k]\big),\ldots,\frac{\partial}{\partial\underline{\theta}_d[k]}\sum_{i=1}^{d}G\big(\underline{\theta}_i[k]\big)\right]^T\end{aligned}$$

for $\underline{\boldsymbol{\theta}}[k]=\underline{\mathbf{a}}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}$ and using equation (4.5)

$$=\big[g\big(\underline{\theta}_1[k]\big),\ldots,g\big(\underline{\theta}_d[k]\big)\big]\big|^T_{\underline{\boldsymbol{\theta}}[k]=\underline{\mathbf{a}}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}},$$

where $g\big(\underline{\theta}_i[k]\big)=\partial G\big(\underline{\theta}_i[k]\big)/\partial\underline{\theta}_i[k]$ as seen in Appendix A. The Hessian matrix of the loss function with respect to the vector $\underline{\mathbf{a}}[k]$ is given by

$$\nabla^2_{\underline{\mathbf{a}}}l\big(\underline{\mathbf{a}}[k]\big)=\frac{\partial^2 l\big(\underline{\mathbf{a}}[k]\big)}{\partial\underline{\mathbf{a}}[k]^2}=\mathbf{V}^{(t)}G''\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)\mathbf{V}^{(t),T},$$

where

$$G''\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)=\left.\begin{bmatrix}\frac{\partial^2 G(\underline{\boldsymbol{\theta}}[k])}{\partial\underline{\theta}_1[k]^2}&\cdots&\frac{\partial^2 G(\underline{\boldsymbol{\theta}}[k])}{\partial\underline{\theta}_d[k]\partial\underline{\theta}_1[k]}\\\vdots&\vdots&\vdots\\\frac{\partial^2 G(\underline{\boldsymbol{\theta}}[k])}{\partial\underline{\theta}_1[k]\partial\underline{\theta}_d[k]}&\cdots&\frac{\partial^2 G(\underline{\boldsymbol{\theta}}[k])}{\partial\underline{\theta}_d[k]^2}\end{bmatrix}\right|_{\underline{\boldsymbol{\theta}}[k]=\underline{\mathbf{a}}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}}.$$

Furthermore,

$$\begin{aligned}\frac{\partial^2 G\big(\underline{\boldsymbol{\theta}}[k]\big)}{\partial\underline{\theta}_r[k]\partial\underline{\theta}_s[k]}&=\frac{\partial^2}{\partial\underline{\theta}_r[k]\partial\underline{\theta}_s[k]}G\big(\underline{\boldsymbol{\theta}}[k]\big)=\frac{\partial^2}{\partial\underline{\theta}_r[k]\partial\underline{\theta}_s[k]}\sum_{i=1}^{d}G(\underline{\theta}_i[k])\\&=\frac{\partial}{\partial\underline{\theta}_s[k]}g(\underline{\theta}_r[k]),\end{aligned}$$

so that

$$\frac{\partial^2 G\big(\underline{\boldsymbol{\theta}}[k]\big)}{\partial \underline{\theta}_r[k]\partial \underline{\theta}_s[k]} = \begin{cases} 0 & \text{if } r \neq s, \\ \partial g(\underline{\theta}_r[k])/\partial \underline{\theta}_r[k] & \text{if } r = s. \end{cases}$$

As a result,

$$G''\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) = \left[\begin{array}{ccc} \frac{\partial g(\underline{\theta}_1[k])}{\partial \underline{\theta}_1[k]} & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & \frac{\partial g(\underline{\theta}_d[k])}{\partial \underline{\theta}_d[k]} \end{array} \right] \Bigg|_{\underline{\boldsymbol{\theta}}[k]=\underline{\mathbf{a}}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}},$$

i.e., $G''\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)$ is a $(d \times d)$ diagonal matrix.

The Newton-Raphson technique simply solves the minimization problem $\arg\min_{\underline{\mathbf{a}}} l(\underline{\mathbf{a}}[k], \mathbf{V}^{(t)}, \mathbf{b}^{(t)})$ at iteration $(t+1)$ by using the update

$$\underline{\mathbf{a}}^{(t+1)}[k] = \underline{\mathbf{a}}^{(t)}[k] - \alpha_{\underline{\mathbf{a}}}^{(t+1)} \Big(\nabla^2_{\underline{\mathbf{a}}} l\big(\underline{\mathbf{a}}^{(t)}[k], \mathbf{V}^{(t)}, \mathbf{b}^{(t)}\big) \Big)^{-1} \nabla_{\underline{\mathbf{a}}} l\big(\underline{\mathbf{a}}^{(t)}[k], \mathbf{V}^{(t)}, \mathbf{b}^{(t)}\big),$$

where $\nabla l(\cdot)$ is the gradient of the function $l(\cdot)$, $\nabla^2 l(\cdot)$ its Hessian matrix and $\alpha_{\underline{\mathbf{a}}}^{(t+1)}$ the so-called step size, cf. Appendix B. It yields the following update equation for the set of vectors $\underline{\mathbf{a}}^{(t+1)}[k]$ at iteration $(t+1)$ for $k = 1, \ldots, n$:

$$\begin{aligned} \underline{\mathbf{a}}^{(t+1)}[k]^T = \underline{\mathbf{a}}^{(t)}[k]^T - \alpha_{\underline{\mathbf{a}}}^{(t+1)} \Big(\mathbf{V}^{(t)} G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)\mathbf{V}^{(t),T} \Big)^{-1} \\ \cdot \Big(\mathbf{V}^{(t)} \big(G'(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}) - \mathbf{x}[k]^T \big) \Big). \end{aligned} \tag{4.14}$$

As in [5, 7], it can be shown that solving the minimization problem $\underline{\mathbf{A}}^{(t+1)} = \arg\min_{\underline{\mathbf{A}}} L(\underline{\mathbf{A}}, \mathbf{V}^{(t)}, \mathbf{b}^{(t)})$ by using a Newton-Raphson method reduces to a repeated weighted least squares in which the inverse of the diagonal values of the matrix $G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)$ are the appropriate weights. Indeed, the update equation (4.14) can be written as

$$\begin{aligned} &\Big(\mathbf{V}^{(t)} G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)\mathbf{V}^{(t),T} \Big) \underline{\mathbf{a}}^{(t+1)}[k]^T \\ &\quad = \Big(\mathbf{V}^{(t)} G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)\mathbf{V}^{(t),T} \Big) \underline{\mathbf{a}}^{(t)}[k]^T \\ &\qquad - \alpha_{\underline{\mathbf{a}}}^{(t+1)} \Big(\mathbf{V}^{(t)} \big(G'(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}) - \mathbf{x}[k]^T \big) \Big). \end{aligned} \tag{4.15}$$

The vector on the right side of equation (4.15) can be written as

$$\mathbf{V}^{(t)} G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)\mathbf{z}^{(t)},$$

where the $(d \times 1)$ vector $\mathbf{z}^{(t)}$ is defined as follows:

$$\begin{aligned}\mathbf{z}^{(t)} &= \mathbf{V}^{(t),T}\underline{\mathbf{a}}^{(t)}[k]^T \\ &\quad - \alpha_{\underline{\mathbf{a}}}^{(t+1)} G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)^{-1}\Big(G'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T\Big).\end{aligned}$$

Then, the update equation (4.15) takes the following form:

$$\begin{aligned}&\Big(\mathbf{V}^{(t)} G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)\mathbf{V}^{(t),T}\Big)\underline{\mathbf{a}}^{(t+1)}[k]^T \\ &\qquad = \mathbf{V}^{(t)} G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)\mathbf{z}^{(t)},\end{aligned} \tag{4.16}$$

which are the so-called *normal equations* of a weighted least squares problem [6,81]. Because the matrix of weights $G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)$ is updated at each iteration, equation (4.16) corresponds to the update of a method known as Iterative Weighted Least Squares (IWLS) [6,82], also called Iterative Reweighted Least Squares (IRLS) [7, 15], or iteratively weighted least squares [5].

The second step in the iterative minimization method consists of the update $\mathbf{V}^{(t+1)} = \arg\min_{\mathbf{V}} L(\underline{\mathbf{A}}^{(t+1)}, \mathbf{V}, \mathbf{b}^{(t)})$. It requires the computation of the gradient vector $\nabla_{\mathbf{v}} l\big(\mathbf{v}_j\big)$ and the Hessian matrix $\nabla^2_{\mathbf{v}} l\big(\mathbf{v}_j\big)$ of the loss function $l\big(\mathbf{v}_j\big)$ with respect to the vector $\mathbf{v}_j$, for all $j = 1, \ldots, q$, where $l\big(\mathbf{v}_j\big) = l\big(\underline{\mathbf{A}}^{(t+1)}, \{\mathbf{v}_j\}_{j=1}^q, \mathbf{b}^{(t)}\big)$ collects the elements of the loss function $L(\underline{\mathbf{A}}^{(t+1)}, \mathbf{V}, \mathbf{b}^{(t)})$ that depend only on the vector $\mathbf{v}_j$.

$$\begin{aligned}l(\mathbf{v}_j) &= \sum_{k=1}^{n}\left\{G\left(\sum_{r=1}^{q}\underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)}\right) - \left(\sum_{r=1}^{q}\underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)}\right)\mathbf{x}[k]^T\right\}, \\ \nabla_{\mathbf{v}} l(\mathbf{v}_j) &= \frac{\partial l(\mathbf{v}_j)}{\partial \mathbf{v}_j} \\ &= \sum_{k=1}^{n}\left\{\underline{a}_j^{(t+1)}[k]G'\left(\sum_{r=1}^{q}\underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)}\right) - \underline{a}_j^{(t+1)}[k]\mathbf{x}[k]^T\right\} \\ &= \sum_{k=1}^{n}\underline{a}_j^{(t+1)}[k]\big\{G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T\big\}, \\ \nabla^2_{\mathbf{v}} l(\mathbf{v}_j) &= \frac{\partial^2 l(\mathbf{v}_j)}{\partial \mathbf{v}_j^2} = \sum_{k=1}^{n}\underline{a}_j^{(t+1)}[k]^2 G''\left(\sum_{r=1}^{q}\underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)}\right) \\ &= \sum_{k=1}^{n}\underline{a}_j^{(t+1)}[k]^2 G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V} + \mathbf{b}^{(t)}\big).\end{aligned}$$

Then, the update equation is given as follows for $j = 1, \ldots, q$:

$$\begin{aligned}\mathbf{v}_j^{(t+1),T} = \mathbf{v}_j^{(t),T} - \alpha_{\mathbf{v}}^{(t+1)} &\left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)\right)^{-1} \\ &\cdot \left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]\big\{G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T\big\}\right),\end{aligned} \tag{4.17}$$

where

$$\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 G''(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}) = \begin{bmatrix} \sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 \frac{\partial g(\underline{\theta}_1[k])}{\partial \underline{\theta}_1[k]} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sum_{k=1}^{n} \underline{a}_j^{(m+1)}[k]^2 \frac{\partial g(\underline{\theta}_d[k])}{\partial \underline{\theta}_d[k]} \end{bmatrix}$$

for $\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}$.

As previously for $\underline{\mathbf{a}}[k]$, the update equation (4.17) for each $\mathbf{v}_j$ can be represented as normal equations of a weighted least squares problem. Because the matrix $G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)$ is updated at each iteration, equation (4.17) also corresponds to the update of the Iterative Reweighted Least Squares method.

The last step in the iterative minimization method consists of the update $\mathbf{b}^{(t+1)} = \arg\min_{\mathbf{b}} L(\underline{\mathbf{A}}^{(t+1)}, \mathbf{V}^{(t+1)}, \mathbf{b})$. It requires the computation of the gradient vector $\nabla_{\mathbf{b}} l\big(\mathbf{b}\big)$ and the Hessian matrix $\nabla^2_{\mathbf{b}} l\big(\mathbf{b}\big)$ of the loss function $l\big(\mathbf{b}\big)$ with respect to the offset vector $\mathbf{b}$, where $l\big(\mathbf{b}\big) = l\big(\underline{\mathbf{A}}^{(t+1)}, \mathbf{V}^{(t+1)}, \mathbf{b}\big)$ collects the elements of the loss function $L(\underline{\mathbf{A}}^{(t+1)}, \mathbf{V}^{(t+1)}, \mathbf{b})$ that depend only on the vector $\mathbf{b}$.

$$\begin{aligned} l(\mathbf{b}) &= \sum_{k=1}^{n} \Big\{G\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) - \big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big)\mathbf{x}[k]^T\Big\}, \\ \nabla_{\mathbf{b}} l(\mathbf{b}) &= \frac{\partial l(\mathbf{b})}{\partial \mathbf{b}} = \sum_{k=1}^{n} \Big\{G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) - \mathbf{x}[k]^T\Big\}, \\ \nabla^2_{\mathbf{b}} l(\mathbf{b}) &= \frac{\partial^2 l(\mathbf{b})}{\partial \mathbf{b}^2} = \sum_{k=1}^{n} G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big). \end{aligned}$$

Then, the update equation is given as follows:

$$\begin{aligned}\mathbf{b}^{(t+1),T} = \mathbf{b}^{(t),T} - \alpha_{\mathbf{b}}^{(t+1)} &\left(\sum_{k=1}^{n} G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}^{(t)}\big)\right)^{-1} \\ &\cdot \left(\sum_{k=1}^{n} \big\{G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T\big\}\right),\end{aligned} \tag{4.18}$$

where

$$G''(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}^{(t)}) = \begin{bmatrix} \frac{\partial g(\underline{\theta}_1[k])}{\partial \underline{\theta}_1[k]} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{\partial g(\underline{\theta}_d[k])}{\partial \underline{\theta}_d[k]} \end{bmatrix}$$

for $\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}^{(t)}$.

Note that only the cumulant generating function $G(\cdot)$ needs to be changed in order to get an algorithm for a loss function involving a new exponential family. This is pertinent since the cumulant generating function uniquely defines the exponential family, cf. Appendix A.

4.2.3 Angle between subspaces

The angle between the estimated lower dimensional subspace $\mathcal{N}$ and the original lower dimensional subspace $\mathcal{M}$ of the parameter space is proposed as a measure to assess the performance of the estimation. As stated in [83], defining the angle between subspaces in $\mathbb{R}^d$, $d \gg 1$, is not as straightforward as the visual geometry of $\mathbb{R}$ or $\mathbb{R}^3$ might suggest.

The *minimal angle* between nonzero subspaces $\mathcal{M}, \mathcal{N} \subseteq \mathbb{R}^d$ is defined to be the number $0 \leq \omega_{min} \leq \pi/2$ for which

$$\cos\omega_{min} = \max_{\substack{\mathbf{u}\in\mathcal{M},\mathbf{v}\in\mathcal{N} \\ \|\mathbf{u}\|_2=\|\mathbf{v}\|_2=1}} \mathbf{v}^T\mathbf{u}. \tag{4.19}$$

If $\mathbf{P}_{\mathcal{M}}$ and $\mathbf{P}_{\mathcal{N}}$ are the orthogonal projectors onto $\mathcal{M}$ and $\mathcal{N}$, respectively, then

$$\cos\omega_{min} = \|\mathbf{P}_{\mathcal{N}}\mathbf{P}_{\mathcal{M}}\|_2.$$

If $\mathcal{M}$ and $\mathcal{N}$ are complementary subspaces ($\mathcal{M} \oplus \mathcal{N} = \mathbb{R}^d$) and if $\mathbf{P}_{\mathcal{MN}}$ is the oblique projector onto $\mathcal{M}$ along $\mathcal{N}$, then

$$\sin \omega_{min} = \frac{1}{\|\mathbf{P}_{\mathcal{MN}}\|_2}.$$

$\mathcal{M}$ and $\mathcal{N}$ are complementary subspaces if and only if $\mathbf{P}_{\mathcal{M}} - \mathbf{P}_{\mathcal{N}}$ is invertible; in this case

$$\sin \omega_{min} = \frac{1}{\|(\mathbf{P}_{\mathcal{M}} - \mathbf{P}_{\mathcal{N}})^{-1}\|_2}.$$

While the minimum angle works fine for complementary subspaces, it may not convey much information about the separation between non-complementary subspaces. For example, $\omega_{min} = 0$ whenever $\mathcal{M}$ and $\mathcal{N}$ have a nontrivial intersection, but there might be a nontrivial "gap" between $\mathcal{M}$ and $\mathcal{N}$ nevertheless.

The *maximal angle* between subspaces $\mathcal{M}, \mathcal{N} \subseteq \mathbb{R}^d$ is defined to be the number $0 \leq \omega_{max} \leq \pi/2$ for which

$$\sin \omega_{max} = \|\mathbf{P}_{\mathcal{M}} - \mathbf{P}_{\mathcal{N}}\|_2. \tag{4.20}$$

The maximum angle is chosen for the assessment of the lower dimensional subspace estimation performance. Since $\mathbf{P}_{\mathcal{M}}$ is the orthogonal projector onto the lower dimensional subspace, and since the matrix $\mathbf{V} \in \mathbb{R}^{q \times d}$ defines this subspace, then

$$\mathbf{P}_{\mathcal{M}} = \mathbf{V}^T \mathbf{V}^+ = \mathbf{V}^T \left(\mathbf{V}\mathbf{V}^T\right)^{-1} \mathbf{V},$$

where the subscript $^+$ denotes a pseudo-inverse.

4.2.4 Positivity constraints and penalty function

For the Exponential, Gamma and Inverse Gaussian distributions, for example, an additional positivity constraint on the natural parameter values has to be taken into account in order to fully comply with the definition of the distributions [80]. Indeed, the natural parameter of the Gamma distribution is the opposite of the inverse of the scale parameter, which is assumed to be strictly positive, cf. Appendix A. As a result, the natural parameter must be constrained to be strictly negative.

Three alternative ways to deal with the positivity constraint are: (1) the use of Lagrange multipliers and Kuhn-Tucker theory; (2) the use of penalty functions; and (3) the use of a non-canonical link function, which enables one to work in an unconstrained parameter space. The latter option is investigated first; second, the penalty functions approach is examined.

4.2.4.1 Non-canonical link approach

The non-canonical link function of an exponential family distribution is the composition of its canonical link with a chosen function. If, for example, the natural parameter has to be strictly negative, then one can choose the composition of the canonical link with the negative square function or the negative absolute value function. The absolute value function might be discarded because of its discontinuity at the origin. Using a composition with the negative square function, the loss function becomes:

$$\widetilde{L}(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) = \sum_{k=1}^{n} B_F\Big(\mathbf{x}[k] \Big\| g\big(- \big[\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}\big]^2\big)\Big) \tag{4.21}$$

instead of

$$L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) = \sum_{k=1}^{n} B_F\big(\mathbf{x}[k] \big\| g(\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b})\big),$$

where $g\big(- \big[\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}\big]^2\big) = g(-\underline{\boldsymbol{\theta}}^2[k])$ with $[\underline{\boldsymbol{\theta}}^2[k]] = \big[\underline{\theta}_1^2[k], \ldots, \underline{\theta}_d[k]^2\big]$. In this case, even though the predictor remains linear, i.e., $\underline{\boldsymbol{\theta}} = \underline{\mathbf{a}}\mathbf{V} + \mathbf{b}$, it is not $\underline{\boldsymbol{\theta}}$ but $-\underline{\boldsymbol{\theta}}^2$ that is used. Hence, the lower dimensional subspace becomes curved instead of flat. A possibility would be to then consider the angle between the original flat subspace and the tangent to the estimated curved subspace instead of the angle between the original flat and estimated flat subspaces in order to assess the parameter subspace estimation performance.

The previously developed iterative minimization algorithm is then used on $\widetilde{L}(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b})$. However, the gradients and the Hessian matrices of the loss function

$\widetilde{L}(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b})$ with respect to $\underline{\mathbf{a}}[k]$, $k = 1, \ldots, n$, to $\mathbf{v}_j$, $j = 1, \ldots, q$ and to $\mathbf{b}$ are then different from the gradients and Hessian matrices of the loss function $L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b})$.

The first step of the iterative minimization problem goes as follows for $k = 1, \ldots, n$:

$$\widetilde{l}\big(\underline{\mathbf{a}}[k]\big) = G\big(-\big[\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big]^2\big) + \big[\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big]^2\mathbf{x}[k]^T,$$
$$\nabla_{\underline{\mathbf{a}}}\widetilde{l}\big(\underline{\mathbf{a}}[k]\big) = \frac{\partial\widetilde{l}\big(\underline{\mathbf{a}}[k]\big)}{\partial\underline{\mathbf{a}}[k]}.$$

For $\underline{\mathbf{a}}[k]$, a $(1 \times q)$ vector, and $\widetilde{l}(\cdot)$, a scalar function of $\underline{\mathbf{a}}[k]$, the gradient $\partial\widetilde{l}\big(\underline{\mathbf{a}}[k]\big)/\partial\underline{\mathbf{a}}[k] = \big[\partial\widetilde{l}\big(\underline{\mathbf{a}}[k]\big)/\partial\underline{a}_1[k], \ldots, \partial\widetilde{l}\big(\underline{\mathbf{a}}[k]\big)/\partial\underline{a}_q[k]\big]^T$ is a $(q \times 1)$ vector. For $j = 1, \ldots, q$:

$$\begin{aligned}
\frac{\partial\widetilde{l}\big(\underline{\mathbf{a}}[k]\big)}{\partial\underline{a}_j[k]} &= \frac{\partial}{\partial\underline{a}_j[k]}\Big\{G\big(-\big[\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big]^2\big)\Big\} + \frac{\partial}{\partial\underline{a}_j[k]}\Big\{\big[\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big]^2\mathbf{x}[k]^T\Big\} \\
&= \sum_{i=1}^{d}\frac{\partial}{\partial\underline{a}_j[k]}\big\{G\big(-\underline{\theta}_i^2[k]\big)\big\} + \sum_{i=1}^{d}\frac{\partial}{\partial\underline{a}_j[k]}\big\{\underline{\theta}_i^2[k]x_i[k]\big\} \\
&= \sum_{i=1}^{d}\big\{-2\underline{\theta}_i[k]\cdot v_{ji}\cdot G'\big(-\underline{\theta}_i^2[k]\big) + 2\underline{\theta}_i[k]\cdot v_{ji}\cdot x_i[k]\big\} \\
&= \sum_{i=1}^{d}-2\underline{\theta}_i[k]\cdot v_{ji}\cdot\big\{G'\big(-\underline{\theta}_i^2[k]\big) - x_i[k]\big\},
\end{aligned}$$

with $\underline{\boldsymbol{\theta}}[k] = \big[\underline{\theta}_1[k], \ldots, \underline{\theta}_d[k]\big] = \underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}$.

Similarly, the Hessian matrix $\nabla^2_{\underline{\mathbf{a}}}\widetilde{l}\big(\underline{\mathbf{a}}[k]\big)$ is a $(q \times q)$ matrix with the element of column $j = 1, \ldots, q$ and row $r = 1, \ldots, q$ defined as:

$$\begin{aligned}
\frac{\partial^2\widetilde{l}\big(\underline{\mathbf{a}}[k]\big)}{\partial\underline{a}_j[k]\partial\underline{a}_r[k]} &= \frac{\partial}{\partial\underline{a}_r[k]}\sum_{i=1}^{d}-2\underline{\theta}_i[k]\cdot v_{ji}\cdot\big\{G'\big(-\underline{\theta}_i^2[k]\big) - x_i[k]\big\} \\
&= \sum_{i=1}^{d}-2v_{ri}\cdot v_{ji}\cdot\big\{G'\big(-\underline{\theta}_i^2[k]\big) - x_i[k]\big\} \\
&\quad + \sum_{i=1}^{d}-2\underline{\theta}_i[k]\cdot v_{ji}\cdot -2\underline{\theta}_i[k]\cdot v_{ri}\cdot G''\big(-\underline{\theta}_i^2[k]\big) \\
&= \sum_{i=1}^{d}-2v_{ri}\cdot v_{ji}\cdot\big\{G'\big(-\underline{\theta}_i^2[k]\big) - x_i[k] - 2\underline{\theta}_i^2[k]\cdot G''\big(-\underline{\theta}_i^2[k]\big)\big\}.
\end{aligned}$$

Recall that the update equation at iteration $(t+1)$ is

$$\underline{\mathbf{a}}^{(t+1)}[k] = \underline{\mathbf{a}}^{(t)}[k] - \alpha_{\underline{\mathbf{a}}}^{(t+1)} \Big(\nabla^2_{\underline{\mathbf{a}}} \widetilde{l}\big(\underline{\mathbf{a}}^{(t)}[k], \mathbf{V}^{(t)}, \mathbf{b}^{(t)}\big) \Big)^{-1} \nabla_{\underline{\mathbf{a}}} \widetilde{l}\big(\underline{\mathbf{a}}^{(t)}[k], \mathbf{V}^{(t)}, \mathbf{b}^{(t)}\big),$$

where $\nabla \widetilde{l}(\cdot)$ is the gradient of the function $\widetilde{l}(\cdot)$, $\nabla^2 \widetilde{l}(\cdot)$ its Hessian matrix and $\alpha_{\underline{\mathbf{a}}}^{(t+1)}$ the so-called step size.

The second step in the iterative minimization problem goes as follows for $j = 1, \ldots, q$:

$$\widetilde{l}(\mathbf{v}_j) = \sum_{k=1}^{n} \left\{ G\left(- \left[\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k] \mathbf{v}_r + \mathbf{b}^{(t)} \right]^2 \right) + \left(\left[\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k] \mathbf{v}_r + \mathbf{b}^{(t)} \right]^2 \right) \mathbf{x}[k]^T \right\},$$

$$\nabla_{\mathbf{v}} \widetilde{l}(\mathbf{v}_j) = \frac{\partial \widetilde{l}(\mathbf{v}_j)}{\partial \mathbf{v}_j}$$

is a $(d \times 1)$ gradient vector whose components are as follows for $i = 1, \ldots, d$:

$$\begin{aligned} \frac{\partial \widetilde{l}(\mathbf{v}_j)}{\partial v_{ji}} &= \frac{\partial}{\partial v_{ji}} \sum_{k=1}^{n} \sum_{s=1}^{d} \big\{ G\big(- \underline{\theta}_s^2[k] \big) + \underline{\theta}_s^2[k] x_s[k] \big\} \\ &= \sum_{k=1}^{n} \big\{ -2\underline{\theta}_i[k] \cdot \underline{a}_j[k] \cdot G'\big(- \underline{\theta}_i^2[k] \big) + 2\underline{\theta}_i[k] \cdot \underline{a}_j[k] \cdot x_i[k] \big\} \\ &= \sum_{k=1}^{n} -2\underline{\theta}_i[k] \cdot \underline{a}_j[k] \cdot \big\{ G'\big(- \underline{\theta}_i^2[k] \big) - x_i[k] \big\}, \end{aligned}$$

with $\underline{\boldsymbol{\theta}}[k] = \big[\underline{\theta}_1[k], \ldots, \underline{\theta}_d[k]\big] = \underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V} + \mathbf{b}^{(t)}$.

Similarly, the Hessian matrix $\nabla^2_{\mathbf{v}_j} \widetilde{l}(\mathbf{v}_j)$ is a $(d \times d)$ matrix with the element of column $i = 1, \ldots, d$ and row $r = 1, \ldots, d$:

$$\begin{aligned} \frac{\partial^2 \widetilde{l}(\mathbf{v}_j)}{\partial v_{ji} \partial v_{jr}} &= \frac{\partial}{\partial v_{jr}} \sum_{k=1}^{n} -2\underline{\theta}_i[k] \cdot \underline{a}_j[k] \cdot \big\{ G'\big(- \underline{\theta}_i^2[k] \big) - x_i[k] \big\} \\ &= \begin{cases} 0 & \text{if } r \neq i, \\ \sum_{k=1}^{n} -2\underline{a}_j^2[k] \cdot \big\{ G'\big(- \underline{\theta}_i^2[k] \big) - x_i[k] - 2\underline{\theta}_i^2[k] \cdot G''\big(- \underline{\theta}_i^2[k] \big) \big\} & \text{if } r = i. \end{cases} \end{aligned}$$

Then, the last step of the iterative minimization problem goes as follows:

$$\widetilde{l}(\mathbf{b}) = \sum_{k=1}^{n} \Big\{ G\big(- \big[\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big]^2\big) + \big[\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big]^2 \mathbf{x}[k]^T \Big\},$$

$$\nabla_{\mathbf{b}}\widetilde{l}(\mathbf{b}) = \frac{\partial \widetilde{l}(\mathbf{b})}{\partial \mathbf{b}} = \sum_{k=1}^{n} \left\{ \frac{\partial}{\partial \mathbf{b}} \Big\{ G\big(- \big[\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big]^2\big) \Big\} \right\} + \sum_{k=1}^{n} \left\{ \frac{\partial}{\partial \mathbf{b}} \Big\{ \big[\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big]^2 \mathbf{x}[k]^T \Big\} \right\}$$

is a $(d \times 1)$ gradient vector whose components are as follows for $i = 1, \ldots, d$:

$$\frac{\partial \widetilde{l}(\mathbf{b})}{\partial b_i} = \sum_{k=1}^{n} \left\{ \sum_{i=1}^{d} \big\{ -2\underline{\theta}_i[k] \cdot G'\big(- \underline{\theta}_i^2[k]\big) + 2\underline{\theta}_i[k] \cdot x_i[k] \big\} \right\},$$

with $\underline{\boldsymbol{\theta}}[k] = \big[\underline{\theta}_1[k], \ldots, \underline{\theta}_d[k]\big] = \underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}$.

Similarly, the Hessian matrix $\nabla^2_{\mathbf{b}}\widetilde{l}(\mathbf{b})$ is a $(d \times d)$ matrix with the element of column $i = 1, \ldots, d$ and row $r = 1, \ldots, d$:

$$\frac{\partial^2 \widetilde{l}(\mathbf{b})}{\partial b_i \partial b_r} = \frac{\partial}{\partial b_r} \sum_{k=1}^{n} \left\{ \sum_{i=1}^{d} \big\{ -2\underline{\theta}_i[k] \cdot G'\big(- \underline{\theta}_i^2[k]\big) + 2\underline{\theta}_i[k] \cdot x_i[k] \big\} \right\}$$

$$= \begin{cases} 0 & \text{if } r \neq i, \\ \sum_{k=1}^{n} -2\big\{ G'\big(- \underline{\theta}_i^2[k]\big) - x_i[k] - 2\underline{\theta}_i^2[k] \cdot G''\big(- \underline{\theta}_i^2[k]\big) \big\} & \text{if } r = i. \end{cases}$$

4.2.4.2 Penalty function approach

As noted in [10], it is possible for the atoms obtained with the extreme GLS case corresponding to exponential family Principal Component Analysis to diverge since the optimum may be at infinity. To avoid such behavior, we introduce a penalty function that defines and places a set of constraints into the loss function via a penalty parameter in a way so that any divergence to infinity is avoided.

The penalty function approach is used to convert the nonlinear programming problem with equality and inequality constraints into an unconstrained problem, or into a problem with simple constraints [84–86]. This transformation is accomplished by defining an appropriate auxiliary function in terms of the problem functions to define a new objective or loss function. In other words, the constraints

are placed into the loss function via a penalty parameter in a way that penalizes any violation of the constraints. It can be difficult to find a penalty function, which is an effective and efficient surrogate for the constraints. As such, there are no general guidelines on designing penalty functions, and constructing an efficient penalty function is quite problem-dependent.

The penalty function is defined as follows for $\underline{\boldsymbol{\theta}} = [\underline{\theta}_1, \ldots, \underline{\theta}_d]$:

$$\psi(\underline{\boldsymbol{\theta}}) = \sum_{i=1}^{d} \left\{ \exp\left(-\beta_{min}(\underline{\theta}_i - \theta_{min}) \right) + \exp\left(\beta_{max}(\underline{\theta}_i - \theta_{max}) \right) \right\}, \tag{4.22}$$

and was designed so that $\psi(\underline{\boldsymbol{\theta}})$ is close to zero for $\theta_{min} \leq \underline{\theta}_i \leq \theta_{max}$, $i = 1, \ldots, d$, and reaches infinity otherwise. Figure 4.1 shows possible shapes for the penalty

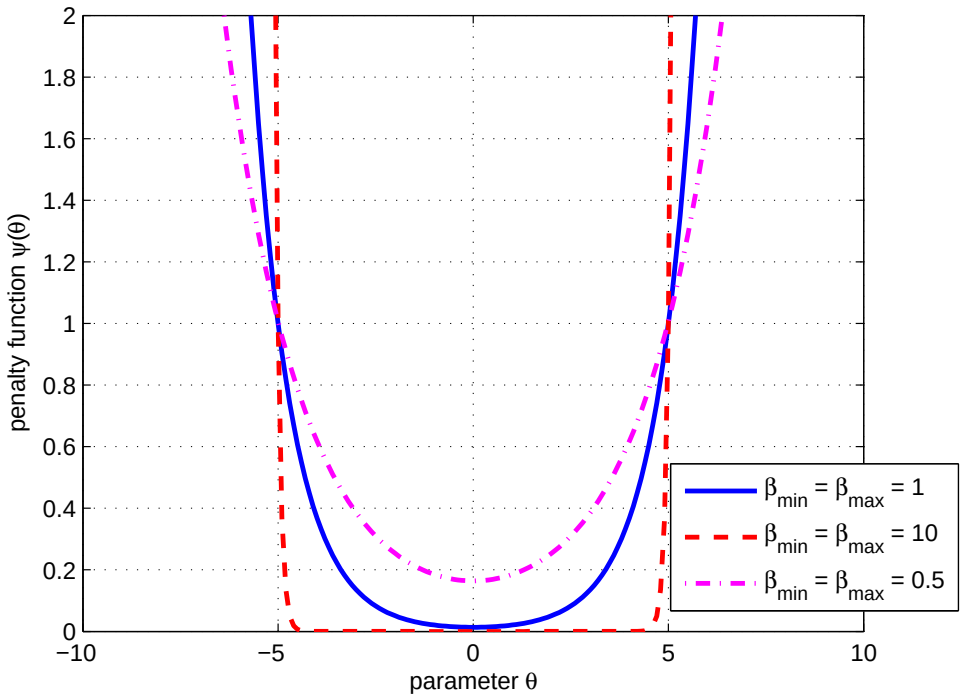

Figure 4.1 Sketches for a possible penalty function ($\theta_{min} = -5, \theta_{max} = 5$): solid line for penalty function parameters $\beta_{min} = \beta_{max} = 1$, dashed line for parameters $\beta_{min} = \beta_{max} = 10$ and dashdot line for $\beta_{min} = \beta_{max} = 0.5$.

function depending on the parameters β_{min} and β_{max} values.

The loss function becomes:

$$\bar{L}(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) = \sum_{k=1}^{n} \Big\{ B_F\big(\mathbf{x}[k] \big\| g(\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b})\big) + c \cdot \psi\big(\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}\big) \Big\} \tag{4.23}$$

instead of

$$L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) = \sum_{k=1}^{n} B_F\big(\mathbf{x}[k] \big\| g(\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b})\big),$$

where the scalar c is called the *penalty parameter*.

The previously developed iterative minimization algorithm is then used on $\bar{L}(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b})$. As did previously happen with the non-canonical link approach, the gradients and the Hessian matrices of the loss function $\bar{L}(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b})$ with respect to $\underline{\mathbf{a}}[k]$, $k = 1, \ldots, n$, to $\mathbf{v}_j$, $j = 1, \ldots, q$ and to $\mathbf{b}$ are then dissimilar to the gradients and Hessian matrices of the loss function $L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b})$.

The first step of the iterative minimization problem goes as follows for $k = 1, \ldots, n$:

$$\begin{aligned}
\bar{l}\big(\underline{\mathbf{a}}[k]\big) &= G\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)\mathbf{x}[k]^T + c \cdot \psi\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big), \\
\nabla_{\underline{\mathbf{a}}} \bar{l}\big(\underline{\mathbf{a}}[k]\big) &= \frac{\partial \bar{l}\big(\underline{\mathbf{a}}[k]\big)}{\partial \underline{\mathbf{a}}[k]} = \mathbf{V}^{(t)} \Big\{ G'\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T + c \cdot \psi'\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \Big\}, \\
\nabla^2_{\underline{\mathbf{a}}} \bar{l}\big(\underline{\mathbf{a}}[k]\big) &= \frac{\partial^2 \bar{l}\big(\underline{\mathbf{a}}[k]\big)}{\partial \underline{\mathbf{a}}[k]^2} = \mathbf{V}^{(t)} \Big\{ G''\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) + c \cdot \psi''\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \Big\} \mathbf{V}^{(t),T}.
\end{aligned}$$

Hence, the update equation becomes for $k = 1, \ldots, n$:

$$\begin{aligned}
\underline{\mathbf{a}}^{(t+1)}[k]^T = \underline{\mathbf{a}}^{(t)}[k]^T &- \alpha_{\underline{\mathbf{a}}}^{(t+1)} \\
&\cdot \Big\{ \mathbf{V}^{(t)} \big\{ G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) + c \cdot \psi''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \big\} \mathbf{V}^{(t),T} \Big\}^{-1} \\
&\cdot \mathbf{V}^{(t)} \Big\{ G'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T + c \cdot \psi'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \Big\}.
\end{aligned}$$

Note that the gradient and Hessian of the penalty function are given by:

$$\psi'\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) = \left. \frac{\partial \psi\big(\underline{\boldsymbol{\theta}}[k]\big)}{\partial \underline{\boldsymbol{\theta}}[k]} \right|_{\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}} \quad \text{and} \quad \frac{\partial \underline{\boldsymbol{\theta}}[k]}{\partial \underline{\mathbf{a}}[k]} = \mathbf{V}^{(t)},$$

with

$$\psi'\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) = \left[\frac{\partial \psi\big(\underline{\boldsymbol{\theta}}[k]\big)}{\partial \underline{\theta}_1[k]}, \ldots, \frac{\partial \psi\big(\underline{\boldsymbol{\theta}}[k]\big)}{\partial \underline{\theta}_d[k]} \right] \Bigg|^T_{\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}}$$

where, for $i = 1, \ldots, d$ and $k = 1, \ldots, n$,

$$\begin{aligned}\frac{\partial \psi\big(\underline{\boldsymbol{\theta}}[k]\big)}{\partial \underline{\theta}_i[k]} &= \frac{\partial}{\partial \underline{\theta}_i[k]} \sum_{r=1}^{d} \exp\big(-\beta_{min}(\underline{\theta}_r[k] - \theta_{min})\big) \\ &\quad + \frac{\partial}{\partial \underline{\theta}_i[k]} \sum_{r=1}^{d} \exp\big(\beta_{max}(\underline{\theta}_r[k] - \theta_{max})\big) \\ &= -\beta_{min} \exp\big(-\beta_{min}(\underline{\theta}_i[k] - \theta_{min})\big) + \beta_{max} \exp\big(\beta_{max}(\underline{\theta}_i[k] - \theta_{max})\big).\end{aligned}$$

The Hessian matrix is given by:

$$\psi''\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) = \left[\begin{array}{ccc} \frac{\partial^2 \psi(\underline{\boldsymbol{\theta}}[k])}{\partial \underline{\theta}_1[k]^2} & \cdots & \frac{\partial^2 \psi(\underline{\boldsymbol{\theta}}[k])}{\partial \underline{\theta}_d[k] \partial \underline{\theta}_1[k]} \\ \vdots & \vdots & \vdots \\ \frac{\partial^2 \psi(\underline{\boldsymbol{\theta}}[k])}{\partial \underline{\theta}_1[k] \partial \underline{\theta}_d[k]} & \cdots & \frac{\partial^2 \psi(\underline{\boldsymbol{\theta}}[k])}{\partial \underline{\theta}_d[k]^2} \end{array}\right]\Bigg|_{\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}},$$

with

$$\begin{aligned}\frac{\partial^2 \psi(\underline{\boldsymbol{\theta}}[k])}{\partial \underline{\theta}_r[k] \partial \underline{\theta}_s[k]} &= \frac{\partial^2}{\partial \underline{\theta}_r[k] \partial \underline{\theta}_s[k]} \sum_{i=1}^{d} \exp\big(-\beta_{min}(\underline{\theta}_i[k] - \theta_{min})\big) \\ &\quad + \frac{\partial^2}{\partial \underline{\theta}_r[k] \partial \underline{\theta}_s[k]} \sum_{i=1}^{d} \exp\big(\beta_{max}(\underline{\theta}_i[k] - \theta_{max})\big) \\ &= \beta_{min}^2 \exp\big(-\beta_{min}(\underline{\theta}_r[k] - \theta_{min})\big) + \beta_{max}^2 \exp\big(\beta_{max}(\underline{\theta}_r[k] - \theta_{max})\big)\end{aligned}$$

for $r = s$, and $\frac{\partial^2 \psi(\underline{\boldsymbol{\theta}}[k])}{\partial \underline{\theta}_r[k] \partial \underline{\theta}_s[k]} = 0$ otherwise.

The second step of the iterative minimization follows for $j = 1, \ldots, q$:

$$\begin{aligned}\bar{l}(\mathbf{v}_j) &= \sum_{k=1}^{n} G\bigg(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)}\bigg) \\ &\quad + \sum_{k=1}^{n} \Big\{ \sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)} \Big\} \mathbf{x}[k]^T + c \cdot \psi\Big(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)} \Big).\end{aligned}$$

Then,

$$\begin{aligned}\nabla_{\mathbf{v}} \bar{l}(\mathbf{v}_j) &= \frac{\partial \bar{l}(\mathbf{v}_j)}{\partial \mathbf{v}_j} \\ &= \sum_{k=1}^{n} \frac{\partial}{\partial \mathbf{v}_j} G\bigg(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)}\bigg) + \sum_{k=1}^{n} \frac{\partial}{\partial \mathbf{v}_j} \Big\{ \sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)} \Big\} \mathbf{x}[k]^T \\ &\quad + c \cdot \sum_{k=1}^{n} \frac{\partial}{\partial \mathbf{v}_j} \psi\Big(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)} \Big)\end{aligned}$$

$$\nabla_{\mathbf{v}}\bar{l}(\mathbf{v}_j) = \sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]\Big\{G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}+\mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T + c\cdot\psi'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}+\mathbf{b}^{(t)}\big)\Big\},$$

$$\begin{aligned}\nabla^2_{\mathbf{v}}\bar{l}(\mathbf{v}_j) &= \frac{\partial^2\bar{l}(\mathbf{v}_j)}{\partial\mathbf{v}_j^2}\\ &= \sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2\Big\{G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}+\mathbf{b}^{(t)}\big) + c\cdot\psi''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}+\mathbf{b}^{(t)}\big)\Big\},\end{aligned}$$

so that the update equation becomes for $j = 1, \ldots, q$:

$$\begin{aligned}&\mathbf{v}_j^{(t+1),T} = \mathbf{v}_j^{(t),T} - \alpha_{\mathbf{v}}^{(t+1)}\cdot\\ &\Bigg(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2\Big\{G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big) + c\cdot\psi''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)\Big\}\Bigg)^{-1}\cdot\\ &\Bigg(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]\big\{G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T + c\cdot\psi'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)\big\}\Bigg).\end{aligned}$$

Finally, the last step of the minimization problem goes as follows:

$$\begin{aligned}\bar{l}(\mathbf{b}) &= \sum_{k=1}^{n}\Big\{G\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}\big) - \big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}\big)\mathbf{x}[k]^T\\ &\quad + c\cdot\psi\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}\big)\Big\},\\ \nabla_{\mathbf{b}}\bar{l}(\mathbf{b}) &= \frac{\partial\bar{l}(\mathbf{b})}{\partial\mathbf{b}}\\ &= \sum_{k=1}^{n}\Big\{G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}\big) - \mathbf{x}[k]^T + c\cdot\psi'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}\big)\Big\},\\ \nabla^2_{\mathbf{b}}\bar{l}(\mathbf{b}) &= \frac{\partial^2\bar{l}(\mathbf{b})}{\partial\mathbf{b}^2} = \sum_{k=1}^{n}\Big\{G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}\big) + c\cdot\psi''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}\big)\Big\}.\end{aligned}$$

Then, the update equation is given as follows:

$$\begin{aligned}&\mathbf{b}^{(t+1),T} = \mathbf{b}^{(t),T} - \alpha_{\mathbf{b}}^{(t+1)}\cdot\\ &\Bigg(\sum_{k=1}^{n}\Big\{G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}^{(t)}\big) + c\cdot\psi''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}^{(t)}\big)\Big\}\Bigg)^{-1}\cdot\\ &\Bigg(\sum_{k=1}^{n}\big\{G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T + c\cdot\psi'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}^{(t)}\big)\big\}\Bigg).\end{aligned}$$

4.2.5 Uniqueness and identifiability

The matrix $\mathbf{V} \in \mathbb{R}^{q\times d}$ defines the low-dimensional parameter subspace. It can be shown that the matrix $\mathbf{V}$, when $q > 1$, is not unique and that other

equivalent representations can be derived by orthogonal transformations of it [45]. Indeed, if $q = 1$, then $\mathbf{V}$ reduces to a row vector of d elements. It is unique, apart from a possible change of sign of all its elements, which corresponds merely to changing the sign of the latent variable. In cases where $q > 1$, there are an infinity of choices for $\mathbf{V}$. The constraint $\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}$ for $k = 1, \ldots, n$, is still satisfied if $\underline{\mathbf{a}}[k]$ is replaced by $\underline{\mathbf{a}}[k]\mathbf{M}$ and $\mathbf{V}$ by $\mathbf{M}^T\mathbf{V}$, where $\mathbf{M}$ is any orthogonal matrix of dimension $(q \times q)$.

In order to reduce the identifiability problem of the matrix $\underline{\boldsymbol{\Theta}} = \underline{\mathbf{A}}\mathbf{V} + \mathbf{B}$, an orthonormality constraint is used, i.e., the condition

$$\mathbf{V}\mathbf{V}^T = \mathbf{I}_{q\times q} \tag{4.24}$$

is enforced. Consider the matrix space $\mathcal{M} = \mathbb{R}^{q\times d}$, then $\mathbf{V} \in \mathcal{M}$. As the iterative minimization process proposed earlier goes on, the successive updates of the matrix $\mathbf{V}$ evolve, giving rise to a curve $\mathbf{V}(t)$ in $\mathcal{M}$, where t describes time. The constraint $\mathbf{V}\mathbf{V}^T = \mathbf{I}_{q\times q}$ corresponds to a hyperplane in $\mathcal{M}$ and an easy way to comply with it would be to impose that the curve $\mathbf{V}(t)$ remains tangential to the $\mathbf{V}\mathbf{V}^T = \mathbf{I}_{q\times q}$ hyperplane. In other words, the progression along the curve $\mathbf{V}(t)$ should remain on the tangent of the $\mathbf{V}\mathbf{V}^T = \mathbf{I}_{q\times q}$ hyperplane. Considering the notation $\dot{\mathbf{V}} = \mathrm{d}\mathbf{V}/\mathrm{d}t$, the tangent to the $\mathbf{V}\mathbf{V}^T = \mathbf{I}_{q\times q}$ hyperplane can be defined by the equation

$$\frac{\dot{\mathbf{V}}\mathbf{V}^T + \mathbf{V}\dot{\mathbf{V}}^T}{2} = \mathbf{0}, \tag{4.25}$$

where the denominator is used for later convenience. Let $\mathbf{M} = \dot{\mathbf{V}} \in \mathcal{M}$ and the following operator is defined:

$$\mathcal{A}(\mathbf{M}) \triangleq \frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2}. \tag{4.26}$$

The operator $\mathcal{A} : \mathcal{M} \to \mathcal{S}$, where $\mathcal{S} \subset \mathbb{R}^{q\times q}$, is linear and onto. Note that $\mathcal{A}(\mathbf{M})^T = \mathcal{A}(\mathbf{M})$, so that $\mathcal{S}$ only contains symmetric matrices.

Proof. This proof is divided into two parts:

- $\mathcal{A}$ is a linear operator;

  For any $\beta \in \mathbb{R}$, any $\mathbf{M}$ and $\widetilde{\mathbf{M}} \in \mathcal{M}$,

$$\begin{aligned}\mathcal{A}(\beta\mathbf{M} + \widetilde{\mathbf{M}}) &= \frac{(\beta\mathbf{M} + \widetilde{\mathbf{M}})\mathbf{V}^T + \mathbf{V}(\beta\mathbf{M} + \widetilde{\mathbf{M}})^T}{2} \\ &= \beta\,\frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2} + \frac{\widetilde{\mathbf{M}}\mathbf{V}^T + \mathbf{V}\widetilde{\mathbf{M}}^T}{2} \\ &= \beta\mathcal{A}(\mathbf{M}) + \mathcal{A}(\widetilde{\mathbf{M}}).\end{aligned}$$

- $\mathcal{A}$ is onto, i.e., $\mathcal{R}(\mathcal{A}) = \mathcal{S}$ the range of $\mathcal{A}$ or $\mathcal{N}(\mathcal{A}^*) = \{\mathbf{0}\}$ the null space of its adjoint operator.

  For any $\mathbf{M} \in \mathcal{M}$ and any $\mathbf{W} \in \mathcal{S}$, the adjoint operator $\mathcal{A}^*$ is defined by

$$\langle \mathbf{W}, \mathcal{A}(\mathbf{M}) \rangle = \langle \mathcal{A}^*(\mathbf{W}), \mathbf{M} \rangle\,, \tag{4.27}$$

  where, using the trace operator tr,

$$\begin{aligned}\langle \mathbf{W}, \mathcal{A}(\mathbf{M}) \rangle &= \operatorname{tr} \mathbf{W}^T \mathcal{A}(\mathbf{M}) \\ &= \operatorname{tr} \mathbf{W}^T \left(\frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2} \right) \\ &= \frac{1}{2}\operatorname{tr}\big(\mathbf{W}^T\mathbf{M}\mathbf{V}^T + \mathbf{W}^T\mathbf{V}\mathbf{M}^T\big) \\ &= \frac{1}{2}\operatorname{tr}\big(\mathbf{W}^T\mathbf{M}\mathbf{V}^T + \mathbf{M}\mathbf{V}^T\mathbf{W}\big) = \frac{1}{2}\operatorname{tr}\big(\mathbf{W}^T\mathbf{M}\mathbf{V}^T + \mathbf{W}\mathbf{M}\mathbf{V}^T\big)\end{aligned}$$

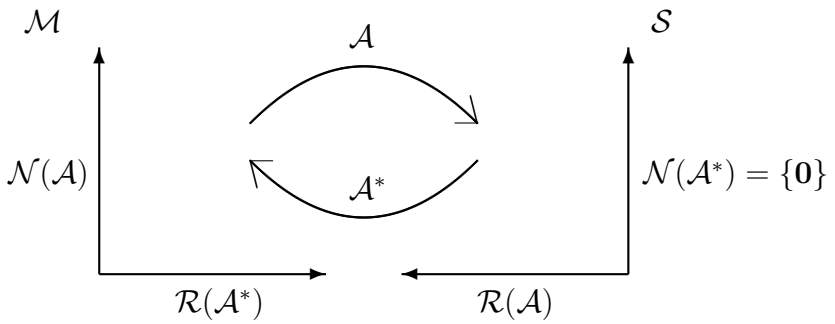

Figure 4.2 The operator $\mathcal{A}$, its range $\mathcal{R}(\mathcal{A})$ and null space $\mathcal{N}(\mathcal{A})$ in relation with its adjoint operator $\mathcal{A}^*$, its range $\mathcal{R}(\mathcal{A}^*)$ and null space $\mathcal{N}(\mathcal{A}^*) = \{\mathbf{0}\}$, with $\mathcal{M} = \mathcal{N}(\mathcal{A}) \cup \mathcal{R}(\mathcal{A}^*)$ and $\mathcal{S} = \mathcal{N}(\mathcal{A}^*) \cup \mathcal{R}(\mathcal{A})$.

using properties of the trace operator,

$$= \operatorname{tr}\left(\frac{\mathbf{W}^T + \mathbf{W}}{2}\right)\mathbf{M}\mathbf{V}^T = \operatorname{tr}\mathbf{W}\mathbf{M}\mathbf{V}^T$$

since $\mathbf{W} \in \mathcal{S}$ is symmetric,

$$\begin{aligned} &= \operatorname{tr}\mathbf{V}^T\mathbf{W}\mathbf{M} = \operatorname{tr}\mathbf{V}^T\mathbf{W}^T\mathbf{M} = \operatorname{tr}(\mathbf{W}\mathbf{V})^T\mathbf{M} \\ &= \langle \mathbf{W}\mathbf{V}, \mathbf{M} \rangle . \end{aligned} \tag{4.28}$$

Now, combining equations (4.27) and (4.28) results in

$$\langle \mathcal{A}^*(\mathbf{W}), \mathbf{M} \rangle = \langle \mathbf{W}\mathbf{V}, \mathbf{M} \rangle .$$

Consequently,

$$\mathcal{A}^*(\mathbf{W}) = \mathbf{W}\mathbf{V}. \tag{4.29}$$

If $\mathcal{A}^*(\mathbf{W}) = \mathbf{0}$, then $\mathbf{W}\mathbf{V} = \mathbf{0}$, meaning $\mathbf{W} = \mathbf{0}$ since $\mathbf{V}$ cannot be the $\mathbf{0}$ matrix. Therefore, $\mathcal{N}(\mathcal{A}^*) = \{\mathbf{0}\}$ and $\mathcal{A}$ is onto.

Figure 4.2 shows the relationship between the range and null space of $\mathcal{A}$ and the range and null space of its adjoint operator $\mathcal{A}^*$. □

Imposing that the curve $\mathbf{V}(t)$ remains tangential to the $\mathbf{V}\mathbf{V}^T = \mathbf{I}_{q\times q}$ hyperplane is equivalent to imposing $\mathcal{A}(\dot{\mathbf{V}}) = \mathbf{0}$, i.e., $\dot{\mathbf{V}}(= \mathrm{d}\mathbf{V}/\mathrm{d}t \simeq \delta\mathbf{V})$ or the increment $\delta\mathbf{V}$ in the $\mathbf{V}$ update equation $\left(\mathbf{V}^{(m+1)} = \mathbf{V}^{(m)} - \alpha_{\mathbf{v}}^{(m+1)}\delta\mathbf{V}\right.$, cf. equation $\left.(4.17)\right)$ needs to be projected onto the null space of $\mathcal{A}$. The projection operator onto the null space of $\mathcal{A}$ is defined as $\mathrm{P}_{\mathcal{N}(\mathcal{A})} = \mathrm{I} - \mathrm{P}_{\mathcal{R}(\mathcal{A}^*)} = \mathrm{I} - \mathcal{A}^+\mathcal{A}$, where the subscript $^+$ denotes a pseudo-inverse. The goal now is to compute $\mathcal{A}^+$. Since $\mathcal{A}$ is onto, $\mathcal{R}(\mathcal{A}) = \mathcal{S}$, for any $\mathbf{M} \in \mathcal{M}$ there exists a matrix $\mathbf{W} \in \mathcal{S}$ such that

$$\mathcal{A}(\mathbf{M}) = \mathbf{W}. \tag{4.30}$$

Similarly, for any $\mathbf{M} \in \mathcal{M}$ there exists a matrix $\mathbf{\Lambda} \in \mathcal{S}$ such that

$$\mathbf{M} = \mathcal{A}^*(\mathbf{\Lambda}). \tag{4.31}$$

Then, combining equations (4.29) and (4.31) gives

$$\mathbf{M} = \mathcal{A}^*(\mathbf{\Lambda}) = \mathbf{\Lambda}\mathbf{V}. \tag{4.32}$$

Now, using both equations (4.30) and (4.32) gives

$$\begin{aligned}\mathcal{A}(\mathbf{M}) &= \mathcal{A}(\mathbf{\Lambda}\mathbf{V}) = \mathbf{W} \\ &= \frac{\mathbf{\Lambda}\mathbf{V}\mathbf{V}^T + \mathbf{V}\mathbf{V}^T\mathbf{\Lambda}^T}{2} = \frac{\mathbf{\Lambda} + \mathbf{\Lambda}^T}{2} = \mathbf{\Lambda},\end{aligned}$$

since $\mathbf{V}\mathbf{V}^T = \mathbf{I}_{q\times q}$ and $\mathbf{\Lambda} \in \mathcal{S}$ is symmetric. As a result, $\mathbf{\Lambda} = \mathbf{W}$. Consequently, $\mathbf{M} = \mathbf{W}\mathbf{V} = \mathcal{A}^+(\mathbf{W})$. The projection operator onto the range of $\mathcal{A}^*$ is defined by

$$\begin{aligned}\mathrm{P}_{\mathcal{R}(\mathcal{A}^*)}(\mathbf{M}) &= \mathcal{A}^+\mathcal{A}(\mathbf{M}) \\ &= \mathcal{A}^+\left(\frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2}\right) = \frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2}\mathbf{V}.\end{aligned}$$

It can be shown that $\mathrm{P}_{\mathcal{R}(\mathcal{A}^*)}$ is correctly defined as a projection operator since it is idempotent, i.e., $\mathrm{P}^2_{\mathcal{R}(\mathcal{A}^*)} = \mathrm{P}_{\mathcal{R}(\mathcal{A}^*)}$.

Proof.

$$\begin{aligned}\mathrm{P}^2_{\mathcal{R}(\mathcal{A}^*)}(\mathbf{M}) &= \mathrm{P}_{\mathcal{R}(\mathcal{A}^*)}\left(\frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2}\mathbf{V}\right) \\ &= \frac{\left(\frac{\mathbf{M}\mathbf{V}^T+\mathbf{V}\mathbf{M}^T}{2}\mathbf{V}\right)\mathbf{V}^T + \mathbf{V}\left(\frac{\mathbf{M}\mathbf{V}^T+\mathbf{V}\mathbf{M}^T}{2}\mathbf{V}\right)^T}{2}\mathbf{V} \\ &= \frac{1}{2}\left\{\left(\frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2}\right)\mathbf{V}\mathbf{V}^T\mathbf{V} + \mathbf{V}\mathbf{V}^T\left(\frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2}\right)^T\mathbf{V}\right\} \\ &= \frac{1}{2}\left\{\left(\frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2}\right)\mathbf{V} + \left(\frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2}\right)\mathbf{V}\right\} \\ &= \left(\frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2}\right)\mathbf{V} = \mathrm{P}_{\mathcal{R}(\mathcal{A}^*)}(\mathbf{M}),\end{aligned}$$

since $\mathbf{V}\mathbf{V}^T = \mathbf{I}_{q\times q}$ and $\mathcal{A}(\mathbf{M})$ is symmetric. □

Then, the projection operator onto the null space of $\mathcal{A}$ is defined by

$$\mathrm{P}_{\mathcal{N}(\mathcal{A})}(\mathbf{M}) = \left(\mathrm{I} - \mathrm{P}_{\mathcal{R}(\mathcal{A}^*)}\right)(\mathbf{M}) = \mathbf{M} - \left(\frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2}\right)\mathbf{V}.$$

It is indeed a projector onto the null space of $\mathcal{A}$ since

$$\begin{aligned}\mathcal{A}\left(\mathrm{P}_{\mathcal{N}(\mathcal{A})}(\mathbf{M})\right) &= \frac{1}{2}\left(\mathbf{M} - \left(\frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2}\right)\mathbf{V}\right)\mathbf{V}^T \\ &\quad + \frac{1}{2}\mathbf{V}\left(\mathbf{M} - \left(\frac{\mathbf{M}\mathbf{V}^T + \mathbf{V}\mathbf{M}^T}{2}\right)\mathbf{V}\right)^T \\ &= \frac{1}{2}\left(\frac{\mathbf{M}\mathbf{V}^T - \mathbf{V}\mathbf{M}^T}{2}\right) - \frac{1}{2}\left(\frac{\mathbf{M}\mathbf{V}^T - \mathbf{V}\mathbf{M}^T}{2}\right) = \mathbf{0}.\end{aligned}$$

Finally, the update equation for $j = 1, \ldots, q$ becomes:

$$\begin{aligned}\mathbf{v}_j^{(t+1),T} = \mathbf{v}_j^{(t),T} - \alpha_{\mathbf{v}}^{(t+1)}\mathrm{P}_{\mathcal{N}(\mathcal{A})}&\left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 G''\left(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\right)\right)^{-1} \\ &\cdot \left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]\left\{G'\left(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\right) - \mathbf{x}[k]^T\right\}\right).\end{aligned}$$

4.2.6 Synthetic data examples

Synthetic data examples for several single exponential family distributions are presented here. Both continuous and discrete distributions are portrayed: the Gaussian and Gamma distributions are examples of continuous distributions whereas the Poisson and Binomial distributions represent discrete ones. The iterative minimization algorithm update equations are described and figures in both data space and parameter space provide insight about the relationship between the parameter space low-dimensional subspace originally used to create the synthetic data sets and the subspace estimated within the GLS framework.

Gaussian data set

The Gaussian distribution describes data that cluster around a mean or average.

A Gaussian random variable with unit-variance has the following probability density function:

$$\begin{aligned}p(x|\mu) &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2}\right) \\ &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)\exp\left(x\mu - \frac{\mu^2}{2}\right), \quad \text{for } x \in \mathbb{R} \text{ and } \mu \in \mathbb{R}.\end{aligned}$$

Identifying the various terms in the definition of the probability density of a standard exponential family yields, cf. Appendix A:

$$\begin{aligned}
\mathcal{N} &= \mathbb{R}, \\
\theta &= \mu, \\
G(\theta) &= \frac{\theta^2}{2}, \\
g(\theta) &= \theta, \\
g'(\theta) &= 1, \\
F(\mu) &= \frac{\mu^2}{2}, \\
B_F\big(x\big\|g(\theta)\big) &= \frac{1}{2}(x-\theta)^2.
\end{aligned}$$

The Gaussian distribution does not require any penalty term that penalizes any divergence to infinity. Therefore, the update equation for the first step of the iterative minimization algorithm takes the following form, for $k = 1, \ldots, n$:

$$\begin{aligned}
\underline{\mathbf{a}}^{(t+1)}[k]^T = \underline{\mathbf{a}}^{(t)}[k]^T - \alpha_{\underline{\mathbf{a}}}^{(t+1)} \cdot \Big\{\mathbf{V}^{(t)}\big\{G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)\big\}\mathbf{V}^{(t),T}\Big\}^{-1} \\
\cdot \mathbf{V}^{(t)}\Big\{G'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T\Big\},
\end{aligned}$$

where

$$\begin{aligned}
G'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) &= \big[g\big(\underline{\theta}_1[k]\big), \ldots, g\big(\underline{\theta}_d[k]\big)\big]\big|^T_{\underline{\boldsymbol{\theta}}[k]=\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}} \\
&= \big[\underline{\theta}_1[k], \ldots, \underline{\theta}_d[k]\big]\big|^T_{\underline{\boldsymbol{\theta}}[k]=\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}}
\end{aligned}$$

is a $(d \times 1)$ vector and

$$G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big) = \left[\begin{array}{ccc} \frac{\partial g(\underline{\theta}_1[k])}{\partial \underline{\theta}_1[k]} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial g(\underline{\theta}_d[k])}{\partial \underline{\theta}_d[k]} \end{array}\right]\Bigg|_{\underline{\boldsymbol{\theta}}[k]=\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}} = \left[\begin{array}{ccc} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{array}\right]$$

is a $(d \times d)$ matrix. Hence, for $k = 1, \ldots, n$,

$$\begin{aligned}
\underline{\mathbf{a}}^{(t+1)}[k]^T &= \underline{\mathbf{a}}^{(t)}[k]^T - \alpha_{\underline{\mathbf{a}}}^{(t+1)} \cdot \Big\{\mathbf{V}^{(t)}\mathbf{V}^{(t),T}\Big\}^{-1}\mathbf{V}^{(t)}\Big\{\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)} - \mathbf{x}[k]\Big\}^T \\
&= \underline{\mathbf{a}}^{(t)}[k]^T - \alpha_{\underline{\mathbf{a}}}^{(t+1)} \cdot \mathbf{V}^{(t)}\Big\{\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)} - \mathbf{x}[k]\Big\}^T.
\end{aligned}$$

Then, for the second step, the update equation is given as follows for $j = 1, \ldots, q$:

$$\begin{aligned}
\mathbf{v}_j^{(t+1),T} &= \mathbf{v}_j^{(t),T} - \alpha_{\mathbf{v}}^{(t+1)} \left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \right)^{-1} \\
&\quad \cdot \left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k] \big\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T \big\} \right) \\
&= \mathbf{v}_j^{(t),T} - \alpha_{\mathbf{v}}^{(t+1)} \left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 \mathbf{I}_{d\times d} \right)^{-1} \\
&\quad \cdot \left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k] \big\{ \underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)} - \mathbf{x}[k] \big\}^T \right).
\end{aligned}$$

For the last step, the update equation is given as follows:

$$\begin{aligned}
\mathbf{b}^{(t+1),T} &= \mathbf{b}^{(t),T} - \alpha_{\mathbf{b}}^{(t+1)} \left(\sum_{k=1}^{n} G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}^{(t)}\big) \right)^{-1} \\
&\quad \cdot \left(\sum_{k=1}^{n} \big\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T \big\} \right) \\
&= \mathbf{b}^{(t),T} - \alpha_{\mathbf{b}}^{(t+1)} \frac{1}{n} \cdot \sum_{k=1}^{n} \big\{ \underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}^{(t)} - \mathbf{x}[k] \big\}^T.
\end{aligned}$$

Figure 4.3 presents a mixture of two Gaussian distributions in data space. Figure 4.4 presents the corresponding parameter space, with the 1-dimensional subspace containing the parameters of the Gaussian distributions. Recall that the extreme case of GLS studied here corresponds to a form of Principal Component Analysis (PCA) performed in parameter space. Also, a characteristic unique to the Gaussian distribution is that, the link function $g(\cdot)$ being the identity, there is no difference between data space and parameter space. Hence, the parameter subspaces estimated within the GLS framework and by classical PCA can both be compared to the original subspace. However, for other exponential family distributions, the subspace estimated by classical PCA lives in data space whereas the subspace estimated within the GLS framework is in parameter space as the original subspace is. Figure 4.5 shows the parameter space 1-dimensional subspace learned with classical PCA. The sine of the angle between the original subspace and the subspace learned with classical PCA is $\sin(\measuredangle_{PCA}) = 0.0052666$. Figure

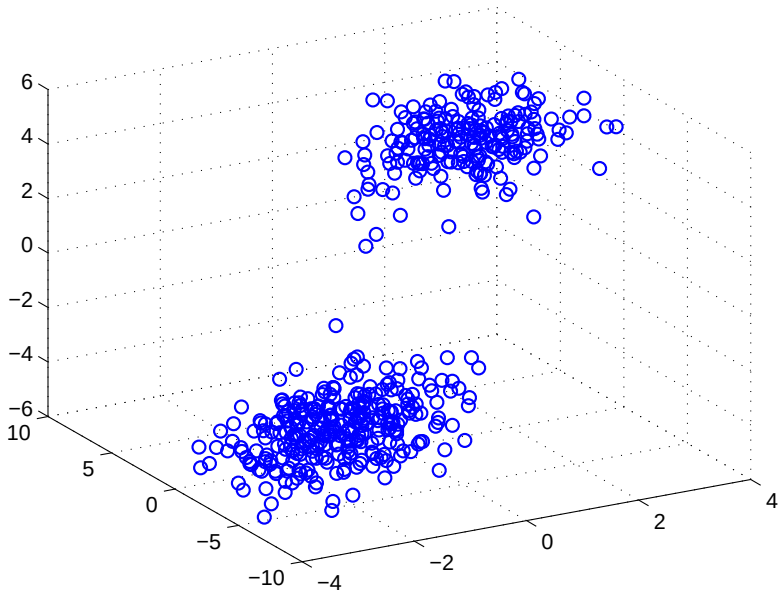

Figure 4.3 Data space: mixture of two Gaussian distributions with parameters constrained on a 1-dimensional subspace.

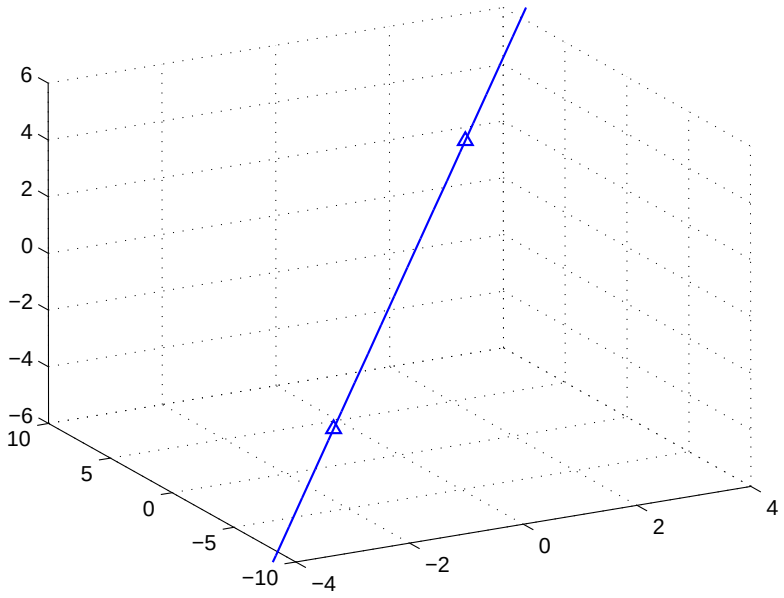

Figure 4.4 Parameter space: parameters ($\triangle$) of two Gaussian distributions, constrained on a 1-dimensional subspace.

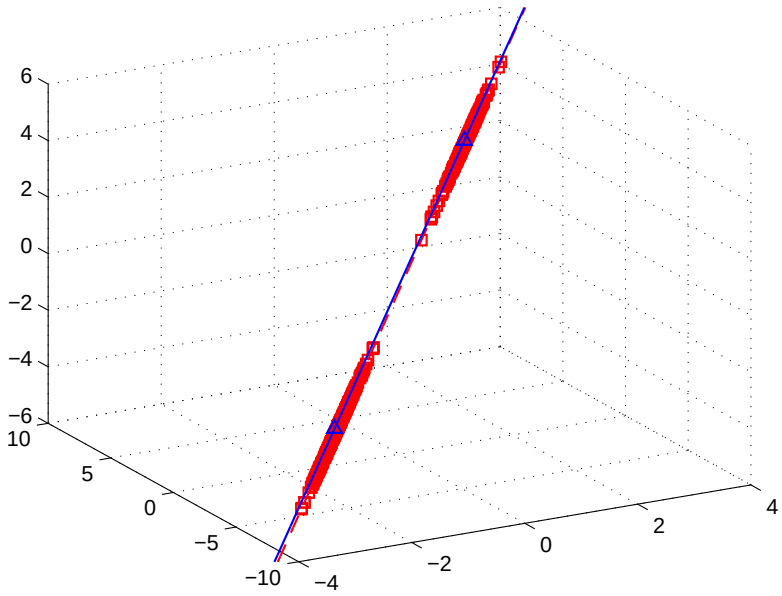

Figure 4.5 Parameter space: original 1-dimensional subspace (solid line) and 1-dimensional subspace estimated with classical PCA (dashed line) with the corresponding data-point projections (□).

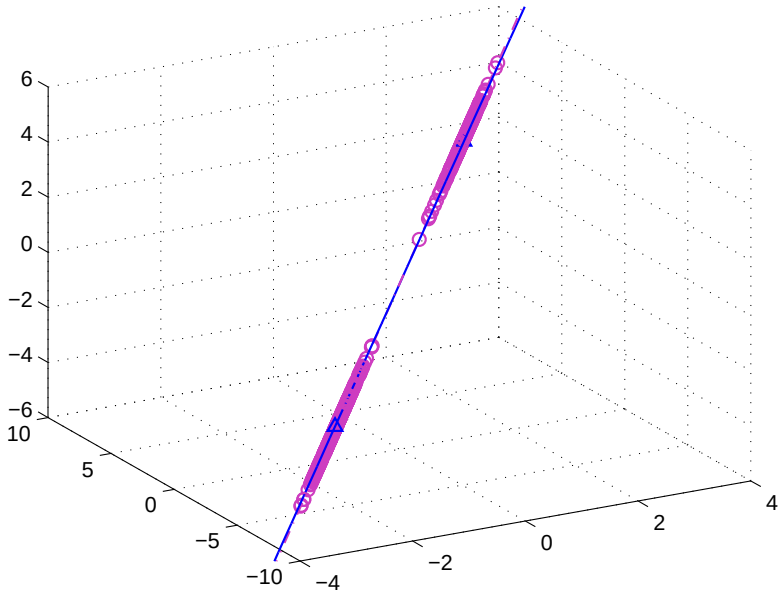

Figure 4.6 Parameter space: original 1-dimensional subspace (solid line) and 1-dimensional subspace estimated with GLS (dashdot line) with the corresponding data-point projections (○).

4.6 shows in parameter space the 1-dimensional subspace learned within the GLS framework. The sine of the angle between the original subspace and the subspace learned with GLS is $\sin(\measuredangle_{GLS}) = 0.0044633$ and $\sin(\measuredangle_{GLS}) < \sin(\measuredangle_{PCA})$.

Gamma data set

The Gamma distribution is often a probability model for waiting times. For instance, in life testing, the waiting time until death is a random variable which is frequently modeled with a Gamma distribution.

A Gamma random variable with shape parameter c and scale parameter b has the following probability density function:

$$p(x|b) = (x/b)^{c-1}[\exp(-x/b)]/b\Gamma(c), \quad \text{for } x \in \mathbb{R}_{\geq 0} \text{ and } b > 0,\ c > 0 \text{ (fixed)}.$$

If c is an integer, then the distribution represents the sum of c independent exponentially distributed random variables, each of which has a mean of b. Hence, if $c = 1$, then the Gamma distribution becomes an Exponential distribution.

Identifying the various terms in the definition of the probability density of a standard exponential family yields, cf. Appendix A:

$$\begin{aligned}
\mathcal{N} &= \mathbb{R}_{<0}, \\
\theta &= -1/b < 0, \\
G(\theta) &= \log[(-1/\theta)^c], \\
g(\theta) &= -c/\theta, \\
g'(\theta) &= c/\theta^2, \\
F(\mu) &= -\log[(\mu/c)^c] - c, \\
B_F\big(x \big\| g(\theta)\big) &= -\log[(-x\theta/c)^c] - x\theta - c.
\end{aligned}$$

Because the scale parameter b is strictly positive, the natural parameter θ has to be strictly negative. One could use the non-canonical link function as described in

Section 4.2.4. However, the loss function is defined as follows:

$$L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) = \sum_{k=1}^{n} \Big\{ G\big(\underline{\boldsymbol{\theta}}[k]\big) - \underline{\boldsymbol{\theta}}[k]\mathbf{x}[k]^T \Big\} = \sum_{k=1}^{n} \sum_{i=1}^{d} \Big\{ G\big(\underline{\theta}_i[k]\big) - \underline{\theta}_i[k] x_i[k] \Big\},$$

and the generative cumulant function associated with the Gamma distribution is:

$$G(\underline{\theta}_i[k]) = \log[(-1/\underline{\theta}_i[k])^c].$$

Assuming that the initialization of $\underline{\boldsymbol{\theta}}$ is strictly negative, as each $\underline{\theta}_i[k]$ goes to 0, $G(\underline{\theta}_i[k])$ goes to $+\infty$, i.e., the loss function $L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b})$ goes to $+\infty$. Hence, if the loss minimization is performed with small steps, the form of the generative cumulant function itself should prevent the parameter values to become nonnegative.

The update equation for the iterative minimization algorithm takes the following form, for $k = 1, \ldots, n$:

$$\begin{aligned}
\underline{\mathbf{a}}^{(t+1)}[k]^T = \underline{\mathbf{a}}^{(t)}[k]^T - \alpha_{\underline{\mathbf{a}}}^{(t+1)} \\
\cdot \Big\{ \mathbf{V}^{(t)} \big\{ G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) + c \cdot \psi''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \big\} \mathbf{V}^{(t),T} \Big\}^{-1} \\
\cdot \mathbf{V}^{(t)} \Big\{ G'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T + c \cdot \psi'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \Big\},
\end{aligned}$$

where

$$\begin{aligned}
G'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) &= \big[g\big(\underline{\theta}_1[k]\big), \ldots, g\big(\underline{\theta}_d[k]\big) \big] \big|^T_{\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}} \\
&= \big[-c/\underline{\theta}_1[k], \ldots, -c/\underline{\theta}_d[k] \big] \big|^T_{\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}}
\end{aligned}$$

and

$$G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) = \left[\begin{array}{ccc} c/\big(\underline{\theta}_1[k]\big)^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & c/\big(\underline{\theta}_d[k]\big)^2 \end{array} \right] \Bigg|_{\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}}.$$

Then, for the second step, the update equation is given as follows for $j = 1, \ldots, q$:

$$\begin{aligned}
&\mathbf{v}_j^{(t+1),T} = \mathbf{v}_j^{(t),T} - \alpha_{\mathbf{v}}^{(t+1)} \cdot \\
&\left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 \Big\{ G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) + c \cdot \psi''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \Big\} \right)^{-1} \cdot \\
&\left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k] \big\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T + c \cdot \psi'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \big\} \right).
\end{aligned}$$

For the last step, the update equation is given as follows:

$$\mathbf{b}^{(t+1),T} = \mathbf{b}^{(t),T} - \alpha_{\mathbf{b}}^{(t+1)} \cdot \left(\sum_{k=1}^{n} \left\{ G''\left(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}^{(t)}\right) + c \cdot \psi''\left(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}^{(t)}\right) \right)^{-1} \cdot \left(\sum_{k=1}^{n} \left\{ G'\left(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}^{(t)}\right) - \mathbf{x}[k]^T + c \cdot \psi'\left(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}^{(t)}\right) \right\} \right).$$

With the Gamma distribution, one might get unlucky and the values of the matrix $\underline{\mathbf{A}}$ stop updating before the matrix $\mathbf{V}$ is completely settled down. In this case, it seems that the iterative minimization approach converges to a local minimum. Fortunately, changing the random seed and starting the minimization with a new random initialization solves this local minimum problem. Additionally, a step-size value of 0.01 for both $\underline{\mathbf{A}}$ and $\mathbf{V}$ seems to be optimum.

Figure 4.7 presents a mixture of two Gamma distributions in data space. Figure 4.8 presents the corresponding parameter space. The original 1-dimensional subspace containing the parameters of the Gamma distributions is represented with a solid line and the subspace learned within the GLS framework with a dashdot line. The sine of the angle between the original subspace and the subspace learned with GLS is $\sin(\measuredangle_{GLS}) = 0.10468$. The estimation result is not as good as the one obtained in the Gaussian case, but the Gamma distribution is intrinsically more challenging than the Gaussian distribution as seen in Figure 4.7.

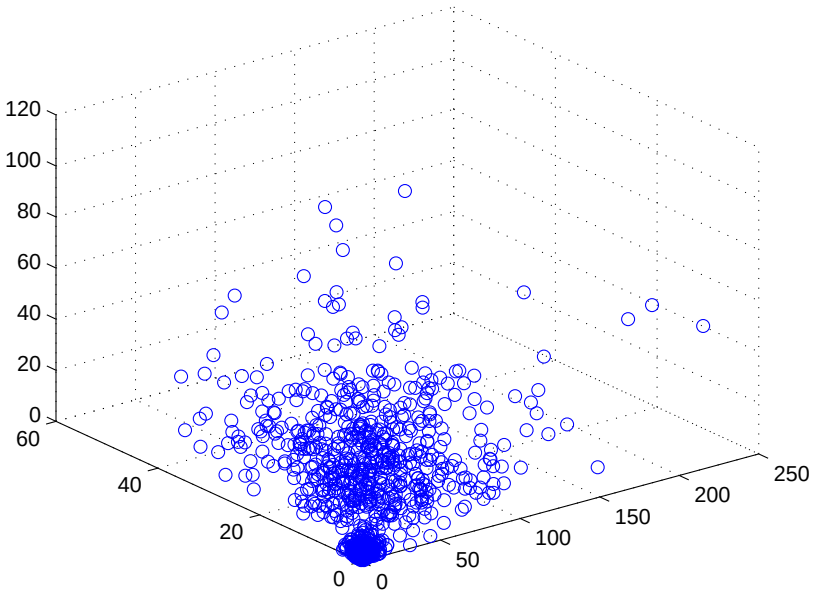

Figure 4.7 Data space: mixture of two Gamma distributions with parameters constrained on a 1-dimensional subspace.

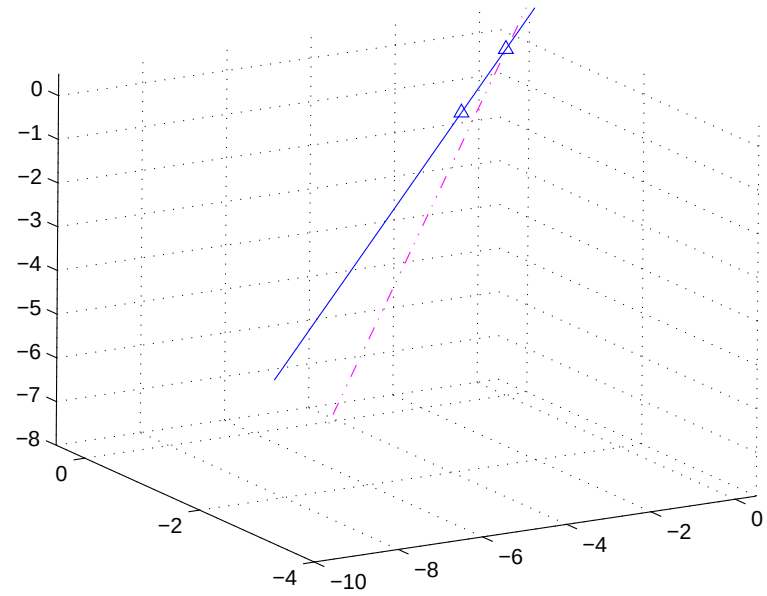

Figure 4.8 Parameter space: parameters ($\triangle$) of two Gamma distributions, original 1-dimensional subspace (solid line with $\triangle$) and 1-dimensional subspace estimated with GLS (dashdot line).

Poisson data set

The Poisson distribution expresses the probability of a number of events occurring in a fixed period of time if these events occur with a known average rate and independently of the time since the last event.

A Poisson random variable has the following probability density function:

$$\begin{aligned} p(x|\lambda) &= \frac{\lambda^x \exp(-\lambda)}{x!} \\ &= \frac{\exp(x \log \lambda - \lambda)}{x!}, \quad \text{for } x \in \mathcal{X} = \{0, 1, 2, \dots\} \text{ and } \lambda > 0. \end{aligned}$$

Identifying the various terms in the definition of the probability density of a standard exponential family yields, cf. Appendix A:

$$\begin{aligned} \mathcal{N} &= \mathbb{R}, \\ \theta &= \log \lambda, \\ G(\theta) &= \exp(\theta), \\ g(\theta) &= \exp(\theta), \\ g'(\theta) &= \exp(\theta), \\ F(\mu) &= \mu \log \mu - \mu, \\ B_F\big(x \big\| g(\theta)\big) &= x \log x - x\theta + \exp(\theta) - x. \end{aligned}$$

The update equation for the iterative minimization algorithm takes the following form, for $k = 1, \dots, n$:

$$\begin{aligned} \underline{\mathbf{a}}^{(t+1)}[k]^T = \underline{\mathbf{a}}^{(t)}[k]^T &- \alpha_{\underline{\mathbf{a}}}^{(t+1)} \\ &\cdot \Big\{ \mathbf{V}^{(t)} \big\{ G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) + c \cdot \psi''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \big\} \mathbf{V}^{(t),T} \Big\}^{-1} \\ &\cdot \mathbf{V}^{(t)} \Big\{ G'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T + c \cdot \psi'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \Big\}, \end{aligned}$$

where

$$\begin{aligned} G'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) &= \big[g\big(\underline{\theta}_1[k]\big), \dots, g\big(\underline{\theta}_d[k]\big) \big] \big|_{\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}}^{T} \\ &= \big[\exp\big(\underline{\theta}_1[k]\big), \dots, \exp\big(\underline{\theta}_d[k]\big) \big] \big|_{\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}}^{T} \end{aligned}$$

and

$$G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)=\left[\begin{array}{ccc}\exp\big(\underline{\theta}_1[k]\big) & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \exp\big(\underline{\theta}_d[k]\big)\end{array}\right]\Bigg|_{\underline{\boldsymbol{\theta}}[k]=\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}}.$$

Then, for the second step, the update equation is given as follows for $j=1,\ldots,q$:

$$\begin{aligned}&\mathbf{v}_j^{(t+1),T}=\mathbf{v}_j^{(t),T}-\alpha_{\mathbf{v}}^{(t+1)}\cdot\\&\left(\sum_{k=1}^{n}\underline{a}_j^{(t+1)}[k]^2\Big\{G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)+c\cdot\psi''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)\Big\}\right)^{-1}\cdot\\&\left(\sum_{k=1}^{n}\underline{a}_j^{(t+1)}[k]\big\{G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)-\mathbf{x}[k]^T+c\cdot\psi'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)\big\}\right).\end{aligned}$$

For the last step, the update equation is given as follows:

$$\begin{aligned}\mathbf{b}^{(t+1),T}&=\mathbf{b}^{(t),T}-\alpha_{\mathbf{b}}^{(t+1)}\cdot\\&\left(\sum_{k=1}^{n}\Big\{G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}^{(t)}\big)+c\cdot\psi''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}^{(t)}\big)\right)^{-1}\cdot\\&\left(\sum_{k=1}^{n}\big\{G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}^{(t)}\big)-\mathbf{x}[k]^T+c\cdot\psi'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}^{(t)}\big)\big\}\right).\end{aligned}$$

Figure 4.9 presents a mixture of two Poisson distributions in data space. Figure 4.10 shows the corresponding parameter space. The original 1-dimensional subspace containing the parameters of the Poisson distributions is represented with a solid line and the subspace learned within the GLS framework with a dashdot line. The sine of the angle between the original subspace and the subspace learned with GLS is $\sin(\measuredangle_{GLS})=0.021943$.

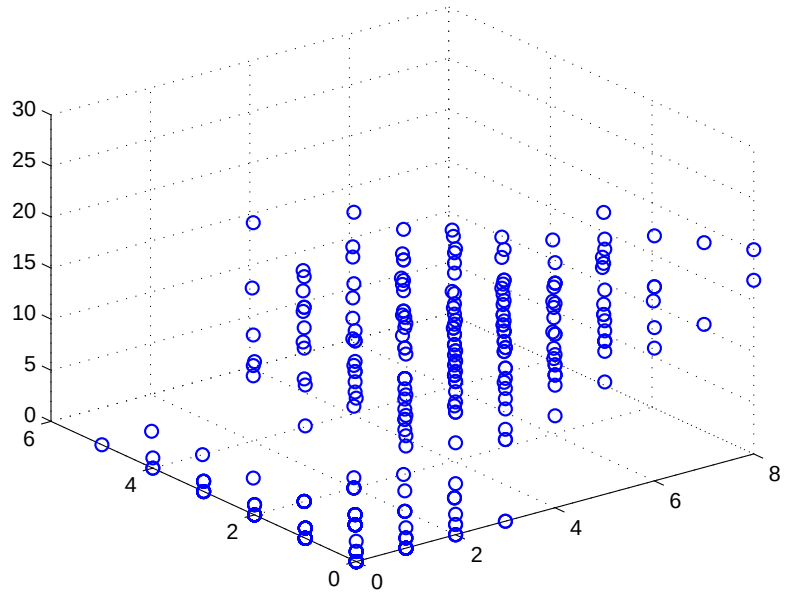

Figure 4.9 Data space: mixture of two Poisson distributions with parameters constrained on a 1-dimensional subspace.

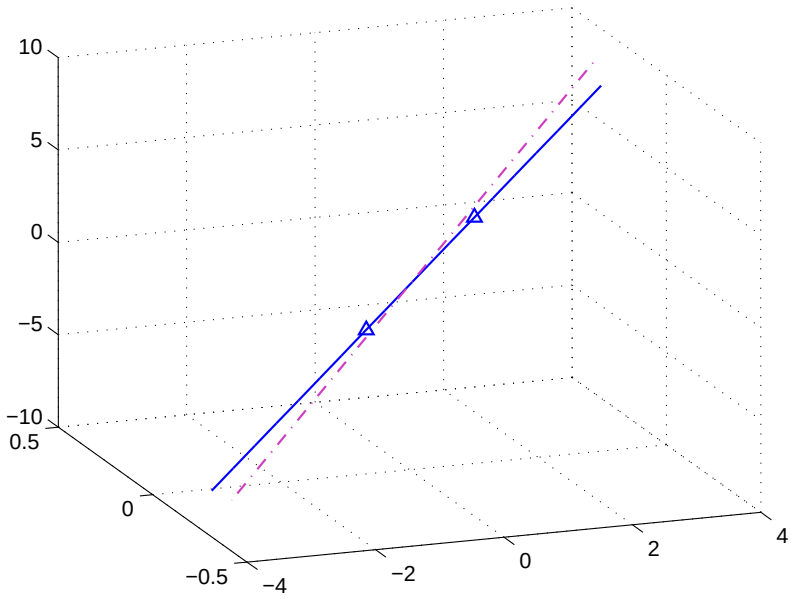

Figure 4.10 Parameter space: parameters ($\triangle$) of two Poisson distributions, original 1-dimensional subspace (solid line) and 1-dimensional subspace estimated with GLS (dashdot line).

Binomial data set

The Binomial distribution depicts the number of successes in a sequence of N independent yes/no experiments, each of which yields success with probability p. Such a success/failure experiment is also called a Bernoulli experiment or Bernoulli trial. In fact, when $N = 1$, the Binomial distribution is a Bernoulli distribution.

A Binomial random variable with unit-variance has the following probability density function:

$$p(x|p) = \frac{N!}{x!(N-x)!} p^x (1-p)^{(N-x)}, \quad \text{for } x \in \mathcal{X} = \{0, 1, 2, \ldots, N\},\ 0 \le p \le 1.$$

Identifying the various terms in the definition of the probability density of a standard exponential family yields, cf. Appendix A:

$$\begin{aligned}
\mathcal{N} &= \mathbb{R}, \\
\theta &= \log \frac{p}{1-p}, \\
G(\theta) &= N \log\big(1 + \exp(\theta)\big), \\
g(\theta) &= N \frac{\exp(\theta)}{1+\exp(\theta)} = N \frac{1}{1+\exp(-\theta)}, \\
g'(\theta) &= N \frac{\exp(\theta)}{\big(1+\exp(\theta)\big)^2}, \\
F(\mu) &= \mu \log \frac{\mu}{N} + (N-\mu) \log \frac{N-\mu}{N}, \\
B_F\big(x \big\| g(\theta)\big) &= N \log \frac{1+\exp(\theta)}{\exp(\theta)} + (N-x)\theta \\
&\quad + x \log \frac{x}{N} + (N-x) \log \frac{N-x}{N}.
\end{aligned}$$

The update equation for the iterative minimization algorithm takes the following form, for $k = 1, \ldots, n$:

$$\begin{aligned}
\underline{\mathbf{a}}^{(t+1)}[k]^T = \underline{\mathbf{a}}^{(t)}[k]^T &- \alpha_{\underline{\mathbf{a}}}^{(t+1)} \\
&\cdot \Big\{ \mathbf{V}^{(t)} \big\{ G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) + c \cdot \psi''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \big\} \mathbf{V}^{(t),T} \Big\}^{-1} \\
&\cdot \mathbf{V}^{(t)} \Big\{ G'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T + c \cdot \psi'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \Big\},
\end{aligned}$$

where

$$G'\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)=\big[g\big(\underline{\theta}_1[k]\big),\ldots,g\big(\underline{\theta}_d[k]\big)\big]\,\big|^{T}_{\underline{\boldsymbol{\theta}}[k]=\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}}$$
$$=\big[N\frac{\exp\big(\underline{\theta}_1[k]\big)}{1+\exp\big(\underline{\theta}_1[k]\big)},\ldots,N\frac{\exp\big(\underline{\theta}_d[k]\big)}{1+\exp\big(\underline{\theta}_d[k]\big)}\big]\Big|^{T}_{\underline{\boldsymbol{\theta}}[k]=\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}}$$

and

$$G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)=\left[\begin{array}{ccc} N\frac{\exp(\underline{\theta}_1[k])}{\{1+\exp(\underline{\theta}_1[k])\}^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & N\frac{\exp(\underline{\theta}_d[k])}{\{1+\exp(\underline{\theta}_d[k])\}^2} \end{array}\right]\Bigg|_{\underline{\boldsymbol{\theta}}[k]=\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}}.$$

Then, for the second step, the update equation is given as follows for $j=1,\ldots,q$:

$$\mathbf{v}_j^{(t+1),T}=\mathbf{v}_j^{(t),T}-\alpha_{\mathbf{v}}^{(t+1)}\cdot$$
$$\Big(\sum_{k=1}^{n}\underline{a}_j^{(t+1)}[k]^2\Big\{G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)+c\cdot\psi''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)\Big\}\Big)^{-1}\cdot$$
$$\Big(\sum_{k=1}^{n}\underline{a}_j^{(t+1)}[k]\big\{G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)-\mathbf{x}[k]^T+c\cdot\psi'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)}+\mathbf{b}^{(t)}\big)\big\}\Big).$$

For the last step, the update equation is given as follows:

$$\mathbf{b}^{(t+1),T}=\mathbf{b}^{(t),T}-\alpha_{\mathbf{b}}^{(t+1)}\cdot$$
$$\Big(\sum_{k=1}^{n}\Big\{G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}^{(t)}\big)+c\cdot\psi''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}^{(t)}\big)\Big)^{-1}\cdot$$
$$\Big(\sum_{k=1}^{n}\big\{G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}^{(t)}\big)-\mathbf{x}[k]^T+c\cdot\psi'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)}+\mathbf{b}^{(t)}\big)\big\}\Big).$$

Figure 4.11 presents a mixture of two Binomial distributions in data space. Figure 4.12 shows the corresponding parameter space. The original 1-dimensional subspace containing the parameters of the Binomial distributions is represented with a solid line and the subspace learned within the GLS framework with a dashdot line. The sine of the angle between the original subspace and the subspace learned with GLS is $\sin(\measuredangle_{GLS})=0.0093628$.

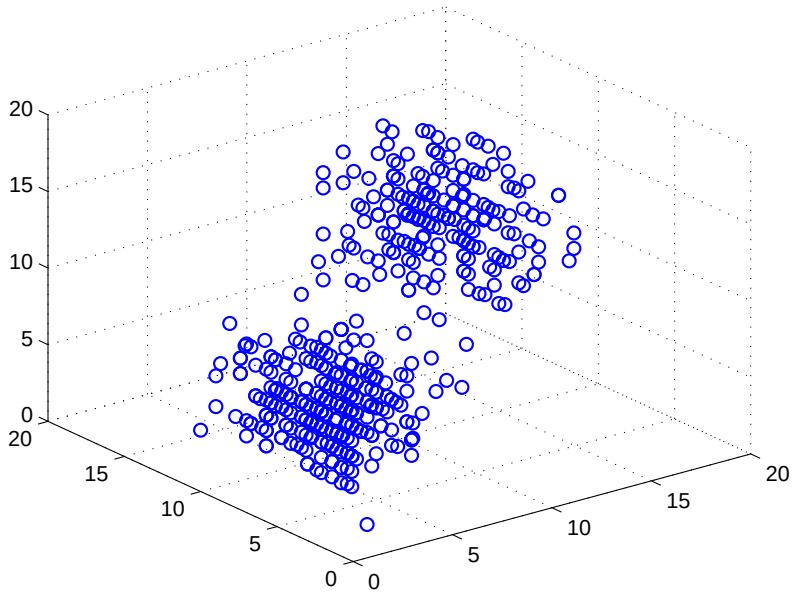

Figure 4.11 Data space: mixture of two Binomial distributions with parameters constrained on a 1-dimensional subspace.

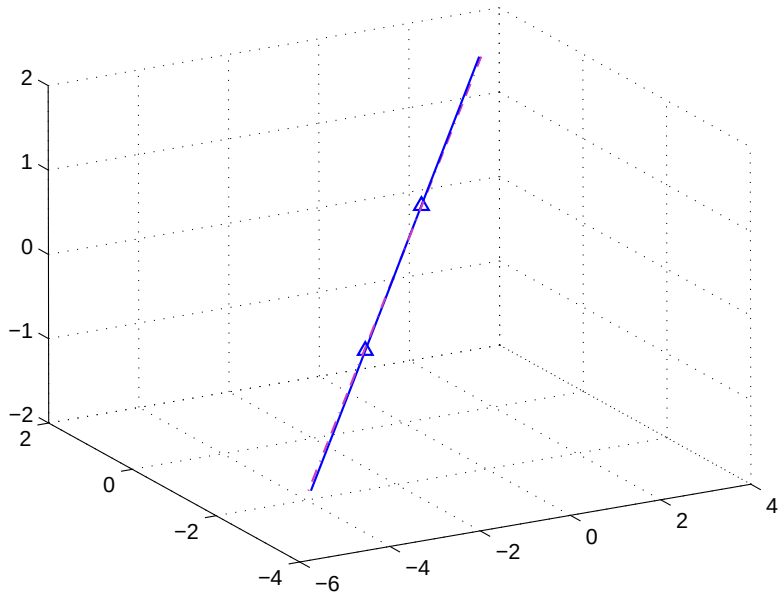

Figure 4.12 Parameter space: parameters ($\triangle$) of two Binomial distributions, original 1-dimensional subspace (solid line) and 1-dimensional subspaces estimated with GLS (dashdot line).

4.3 Mixed data types

The above approach was proposed assuming that the data attributes have the same distribution. It can be extended to the hybrid dimensionality reduction problem. By hybrid we mean a problem in which different types of distributions can be used for different attributes. Often, records have attributes that can be both continuous (with different underlying distributions) and discrete, such as categorical, count or Boolean. This situation is referred to as the mixed data-type case throughout this dissertation. Below, a derivation of the algorithm for the hybrid dimensionality problem is presented. Only two types of exponential family distributions are considered (for example, the Bernoulli distribution and the Gaussian distribution). Of course, this approach generalizes to any number of exponential family distributions.

For simplicity of presentation, we consider that the f first attributes are distributed according to the exponential family distribution $p^{(\mathbf{1})}$ and the $(d-f)$ last attributes are distributed according to the exponential family distribution $p^{(\mathbf{2})}$. Following the previously stated example, the bold superscript $^{(\mathbf{1})}$ would correspond to Bernoulli distributed attributes and $^{(\mathbf{2})}$ to Gaussian distributed attributes. Then,

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}[1] \\ \mathbf{x}[2] \\ \vdots \\ \mathbf{x}[n] \end{pmatrix} = \left(\begin{array}{ccc|ccc} x_1[1] & \dots & x_f[1] & x_{f+1}[1] & \dots & x_d[1] \\ x_1[2] & \dots & x_f[2] & x_{f+1}[2] & \dots & x_d[2] \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_1[n] & \dots & x_f[n] & x_{f+1}[n] & \dots & x_d[n] \end{array}\right) = \left(\mathbf{X}^{(\mathbf{1})} \;\middle|\; \mathbf{X}^{(\mathbf{2})}\right).$$

The loss function is expressed as follows:

$$L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) = -\log p(\mathbf{X}|\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) = -\sum_{k=1}^{n} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[k]\big),$$

using the *iid statistical samples assumption*, where $\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}$. Then, using

the *latent variable assumption*,

$$
\begin{aligned}
&p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[k]\big) \\
&\quad = p_1(x_1[k]|\underline{\theta}_1[k])\cdots p_f(x_f[k]|\underline{\theta}_f[k])p_{f+1}(x_{f+1}[k]|\underline{\theta}_{f+1}[k])\cdots p_d(x_d[k]|\underline{\theta}_d[k]) \\
&\quad = p^{(\mathbf{1})}(x_1[k]|\underline{\theta}_1[k])\cdots p^{(\mathbf{1})}(x_f[k]|\underline{\theta}_f[k])p^{(\mathbf{2})}(x_{f+1}[k]|\underline{\theta}_{f+1}[k])\cdots p^{(\mathbf{2})}(x_d[k]|\underline{\theta}_d[k]) \\
&\quad = p^{(\mathbf{1})}\big(\mathbf{x}^{(\mathbf{1})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{1})}[k]\big)p^{(\mathbf{2})}\big(\mathbf{x}^{(\mathbf{2})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{2})}[k]\big),
\end{aligned}
$$

where

$$
\underline{\boldsymbol{\Theta}} = \begin{pmatrix} \underline{\boldsymbol{\theta}}[1] \\ \underline{\boldsymbol{\theta}}[2] \\ \vdots \\ \underline{\boldsymbol{\theta}}[n] \end{pmatrix} = \left(\begin{array}{ccc|ccc} \underline{\theta}_1[1] & \dots & \underline{\theta}_f[1] & \underline{\theta}_{f+1}[1] & \dots & \underline{\theta}_d[1] \\ \underline{\theta}_1[2] & \dots & \underline{\theta}_f[2] & \underline{\theta}_{f+1}[2] & \dots & \underline{\theta}_d[2] \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \underline{\theta}_1[n] & \dots & \underline{\theta}_f[n] & \underline{\theta}_{f+1}[n] & \dots & \underline{\theta}_d[n] \end{array}\right) = \left(\underline{\boldsymbol{\Theta}}^{(\mathbf{1})} \;\middle|\; \underline{\boldsymbol{\Theta}}^{(\mathbf{2})}\right).
$$

The matrix of parameters $\underline{\boldsymbol{\Theta}} = \underline{\mathbf{A}}\mathbf{V} + \mathbf{B}$ results in the following decompositions:

$$
\mathbf{V} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_q \end{pmatrix} = \left(\begin{array}{ccc|ccc} v_{11} & \dots & v_{1f} & v_{1(f+1)} & \dots & v_{1d} \\ v_{21} & \dots & v_{2f} & v_{2(f+1)} & \dots & v_{2d} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ v_{q1} & \dots & v_{qf} & v_{q(f+1)} & \dots & v_{qd} \end{array}\right) = \left(\mathbf{V}^{(\mathbf{1})} \;\middle|\; \mathbf{V}^{(\mathbf{2})}\right),
$$

and

$$
\mathbf{B} = \begin{pmatrix} \mathbf{b} \\ \mathbf{b} \\ \vdots \\ \mathbf{b} \end{pmatrix} = \left(\begin{array}{ccc|ccc} b_1 & \dots & b_f & b_{(f+1)} & \dots & b_d \\ b_1 & \dots & b_f & b_{(f+1)} & \dots & b_d \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ b_1 & \dots & b_f & b_{(f+1)} & \dots & b_d \end{array}\right) = \left(\mathbf{B}^{(\mathbf{1})} \;\middle|\; \mathbf{B}^{(\mathbf{2})}\right),
$$

where $\mathbf{B}^{(\mathbf{1})} = [\mathbf{b}^{(\mathbf{1})}, \dots, \mathbf{b}^{(\mathbf{1})}]^T$ and $\mathbf{b}^{(\mathbf{1})} = [b_1, \dots, b_f]$, $\mathbf{B}^{(\mathbf{2})} = [\mathbf{b}^{(\mathbf{2})}, \dots, \mathbf{b}^{(\mathbf{2})}]^T$ and $\mathbf{b}^{(\mathbf{2})} = [b_{f+1}, \dots, b_d]$. Then,

$$
\underline{\boldsymbol{\Theta}} = \begin{pmatrix} \underline{\boldsymbol{\theta}}[1] \\ \underline{\boldsymbol{\theta}}[2] \\ \vdots \\ \underline{\boldsymbol{\theta}}[n] \end{pmatrix} = \underline{\mathbf{A}}\mathbf{V} + \mathbf{B} = \left(\underline{\mathbf{A}}\mathbf{V}^{(\mathbf{1})} + \mathbf{B}^{(\mathbf{1})} \;\middle|\; \underline{\mathbf{A}}\mathbf{V}^{(\mathbf{2})} + \mathbf{B}^{(\mathbf{2})}\right).
$$

Now, notice that

$$\underline{\mathbf{A}}\mathbf{V}^{(\mathbf{1})}+\mathbf{B}^{(\mathbf{1})}=\underline{\begin{pmatrix}\underline{a}_1[1] & \dots & \underline{a}_q[1]\\ \underline{a}_1[2] & \dots & \underline{a}_q[2]\\ \vdots & \ddots & \vdots\\ \underline{a}_1[n] & \dots & \underline{a}_q[n]\end{pmatrix}\begin{pmatrix}v_{11} & \dots & v_{1f}\\ v_{21} & \dots & v_{2f}\\ \vdots & \ddots & \vdots\\ v_{q1} & \dots & v_{qf}\end{pmatrix}}+\begin{pmatrix}b_1 & \dots & b_f\\ b_1 & \dots & b_f\\ \vdots & \ddots & \vdots\\ b_1 & \dots & b_f\end{pmatrix}.$$

The underlined term is a $(n \times f)$ matrix whose elements are $\sum_{j=1}^{q} \underline{a}_j[k]v_{ji}$ for all $k = 1, \dots, n$ and $i = 1, \dots, f$. Consequently, the elements of the $(n \times f)$ matrix $\underline{\mathbf{A}}\mathbf{V}^{(\mathbf{1})} + \mathbf{B}^{(\mathbf{1})}$ are of the form $\sum_{j=1}^{q} \underline{a}_j[k]v_{ji} + b_i$ for all $k = 1, \dots, n$ and $i = 1, \dots, f$. The matrix $\underline{\mathbf{\Theta}}^{(\mathbf{1})}$ is also $(n \times f)$ and its elements take the following form: $\underline{\theta}_i[k] = \sum_{j=1}^{q} \underline{a}_j[k]v_{ji} + b_i$ for all $k = 1, \dots, n$ and $i = 1, \dots, f$. Besides, knowing that $\underline{\theta}_i[k] = \sum_{j=1}^{q} \underline{a}_j[k]v_{ji} + b_i$ for all $k = 1, \dots, n$ and $i = 1, \dots, d$, the matrix $\underline{\mathbf{\Theta}}$ can be expressed as:

$$\underline{\mathbf{\Theta}}=\left(\begin{array}{ccc|ccc}\sum_{j=1}^{q}\underline{a}_j[1]v_{11}+b_1 & \dots & \sum_{j=1}^{q}\underline{a}_j[1]v_{jf}+b_f & \dots & \sum_{j=1}^{q}\underline{a}_j[1]v_{jd}+b_d\\ \sum_{j=1}^{q}\underline{a}_j[2]v_{j1}+b_1 & \dots & \sum_{j=1}^{q}\underline{a}_j[2]v_{jf}+b_f & \dots & \sum_{j=1}^{q}\underline{a}_j[2]v_{jd}+b_d\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ \sum_{j=1}^{q}\underline{a}_j[n]v_{j1}+b_1 & \dots & \sum_{j=1}^{q}\underline{a}_j[n]v_{jf}+b_f & \dots & \sum_{j=1}^{q}\underline{a}_j[n]v_{jd}+b_d\end{array}\right).$$

As a result, it becomes clear that $\underline{\mathbf{\Theta}}^{(\mathbf{1})} = \underline{\mathbf{A}}\mathbf{V}^{(\mathbf{1})} + \mathbf{B}^{(\mathbf{1})}$, and $\underline{\mathbf{\Theta}}^{(\mathbf{2})} = \underline{\mathbf{A}}\mathbf{V}^{(\mathbf{2})} + \mathbf{B}^{(\mathbf{2})}$. Note that, even though we are able to separate the matrix $\underline{\mathbf{\Theta}}$ into two blocks, the matrix $\underline{\mathbf{A}}$ is common to both $\underline{\mathbf{\Theta}}^{(\mathbf{1})}$ and $\underline{\mathbf{\Theta}}^{(\mathbf{2})}$. Therefore, the loss function takes the following form:

$$\begin{aligned}L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) &= -\sum_{k=1}^{n} \log p\big(\mathbf{x}[k]|\underline{\mathbf{a}}[k], \mathbf{V}, \mathbf{b}\big)\\ &= -\sum_{k=1}^{n} \log p^{(\mathbf{1})}\big(\mathbf{x}^{(\mathbf{1})}[k]|\underline{\mathbf{a}}[k], \mathbf{V}^{(\mathbf{1})}, \mathbf{b}^{(\mathbf{1})}\big) - \sum_{k=1}^{n} \log p^{(\mathbf{2})}\big(\mathbf{x}^{(\mathbf{2})}[k]|\underline{\mathbf{a}}[k], \mathbf{V}^{(\mathbf{2})}, \mathbf{b}^{(\mathbf{2})}\big),\end{aligned}$$

and the minimization problem can be expressed as:

$$\begin{aligned}
&\arg\min_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} L(\underline{\mathbf{A}},\mathbf{V},\mathbf{b}) \\
&\quad = \arg\min_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \Bigg\{ \sum_{k=1}^{n} \Big\{ G^{(\mathbf{1})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})} + \mathbf{b}^{(\mathbf{1})}\big) - \big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})} + \mathbf{b}^{(\mathbf{1})}\big)\mathbf{x}^{(\mathbf{1})}[k]^T \Big\} \\
&\qquad + \sum_{k=1}^{n} \Big\{ G^{(\mathbf{2})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})} + \mathbf{b}^{(\mathbf{2})}\big) - \big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})} + \mathbf{b}^{(\mathbf{2})}\big)\mathbf{x}^{(\mathbf{2})}[k]^T \Big\} \Bigg\}.
\end{aligned}$$

Since the linear combination of convex functions with nonnegative coefficients is always convex [84], the loss function remains convex in either of its arguments with the others fixed. Therefore, the iterative minimization technique proposed for the single exponential family can be applied in the mixture of exponential families case.

The first step in the Newton-Raphson minimization technique, given a fixed matrix $\mathbf{V}$ and fixed vector $\mathbf{b}$, is to obtain the matrix $\underline{\mathbf{A}}$, or the set of vectors $\underline{\mathbf{a}}[k]$ for $k = 1, \ldots, n$, that minimizes the loss function. The second step, given a fixed matrix $\underline{\mathbf{A}}$ and fixed vector $\mathbf{b}$, is to obtain the matrix $\mathbf{V}$ that minimizes the loss function. The last step, given a fixed matrix $\underline{\mathbf{A}}$ and a fixed matrix $\mathbf{V}$, is to obtain the vector $\mathbf{b}$. The updates are derived in a way similar to the one used in Section 4.2. As previously, the superscript $^{(t)}$ means an estimate obtained at the end of the t^{th} iteration of the iterative minimization process. Note that, in order to avoid confusion, the step superscript is not bold whereas the mixture superscripts $^{(\mathbf{1})}$ and $^{(\mathbf{2})}$ are.

$$\begin{aligned}
l\big(\underline{\mathbf{a}}[k]\big) &= G^{(\mathbf{1})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big) - \big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big)\mathbf{x}^{(\mathbf{1})}[k]^T \\
&\quad + G^{(\mathbf{2})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big) - \big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big)\mathbf{x}^{(\mathbf{2})}[k]^T, \\
\nabla_{\underline{\mathbf{a}}} l\big(\underline{\mathbf{a}}[k]\big) &= \frac{\partial l\big(\underline{\mathbf{a}}[k]\big)}{\partial \underline{\mathbf{a}}[k]} = \mathbf{V}^{(\mathbf{1})(t)} G^{(\mathbf{1})'}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big) - \mathbf{V}^{(\mathbf{1})(t)}\mathbf{x}^{(\mathbf{1})}[k]^T \\
&\quad + \mathbf{V}^{(\mathbf{2})(t)} G^{(\mathbf{2})'}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big) - \mathbf{V}^{(\mathbf{2})(t)}\mathbf{x}^{(2)}[k]^T \\
&= \mathbf{V}^{(\mathbf{1})(t)} \Big\{ G^{(\mathbf{1})'}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big) - \mathbf{x}^{(\mathbf{1})}[k]^T \Big\} \\
&\quad + \mathbf{V}^{(\mathbf{2})(t)} \Big\{ G^{(\mathbf{2})'}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big) - \mathbf{x}^{(\mathbf{2})}[k]^T \Big\},
\end{aligned}$$

$$\nabla^2_{\underline{\mathbf{a}}} l\big(\underline{\mathbf{a}}[k]\big) = \frac{\partial^2 l\big(\underline{\mathbf{a}}[k]\big)}{\partial \underline{\mathbf{a}}[k]^2} = \mathbf{V}^{(\mathbf{1})(t)} G^{(\mathbf{1})''}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big)\mathbf{V}^{(\mathbf{1})(t),T} + \mathbf{V}^{(\mathbf{2})(t)} G^{(\mathbf{2})''}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big)\mathbf{V}^{(\mathbf{2})(t),T}.$$

The update equation for the set of vectors $\underline{\mathbf{a}}[k]$ for $k = 1, \cdots, n$ is:

$$\begin{aligned}\underline{\mathbf{a}}^{(t+1)}[k]^T = \underline{\mathbf{a}}^{(t)}[k]^T - \alpha_{\underline{\mathbf{a}}}^{(t+1)} \Big\{ & \mathbf{V}^{(\mathbf{1})(t)} G^{(\mathbf{1})''}\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big)\mathbf{V}^{(\mathbf{1})(t),T} \\ & + \mathbf{V}^{(\mathbf{2})(t)} G^{(\mathbf{2})''}\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big)\mathbf{V}^{(\mathbf{2})(t),T}\Big\}^{-1} \\ & \cdot \Big\{ \mathbf{V}^{(\mathbf{1})(t)}\Big(G^{(\mathbf{1})'}\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big) - \mathbf{x}[k]^{(\mathbf{1}),T}\Big) \\ & + \mathbf{V}^{(\mathbf{2})(t)}\Big(G^{(\mathbf{2})'}\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big) - \mathbf{x}[k]^{(\mathbf{2}),T}\Big)\Big\}.\end{aligned}$$

For the second step, the two sets of row vectors $\big\{\mathbf{v}_j^{(\mathbf{1})}\big\}_{j=1}^q$ and $\big\{\mathbf{v}_j^{(\mathbf{2})}\big\}_{j=1}^q$ are updated separately. For the sake of simplicity, the following derivations are made for the set $\big\{\mathbf{v}_j\big\}_{j=1}^q$ indistinct of the mixture superscript. The update equation can be used for $\big\{\mathbf{v}_j^{(\mathbf{1})}\big\}_{j=1}^q$ and $\big\{\mathbf{v}_j^{(\mathbf{2})}\big\}_{j=1}^q$ by changing $\mathbf{v}_j$ to $\mathbf{v}_j^{(\mathbf{1})}$, respectively to $\mathbf{v}_j^{(\mathbf{2})}$, $\mathbf{b}$ to $\mathbf{b}^{(\mathbf{1})}$, respectively to $\mathbf{b}^{(\mathbf{2})}$, $G(\cdot), G'(\cdot)$, and $G''(\cdot)$ to $G^{(\mathbf{1})}(\cdot), G^{(\mathbf{1})'}(\cdot)$, and $G^{(\mathbf{1})''}(\cdot)$, respectively to $G^{(\mathbf{2})}(\cdot), G^{(\mathbf{2})'}(\cdot)$, and $G^{(\mathbf{2})''}(\cdot)$.

$$l(\mathbf{v}_j) = \sum_{k=1}^{n}\left\{ G\left(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)}\right) - \left(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)}\right)\mathbf{x}[k]^T\right\},$$

$$\begin{aligned}\nabla_{\mathbf{v}} l(\mathbf{v}_j) = \frac{\partial l(\mathbf{v}_j)}{\partial \mathbf{v}_j} &= \sum_{k=1}^{n}\left\{ \underline{a}_j^{(t+1)}[k] G'\left(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)}\right) - \underline{a}_j^{(t+1)}[k]\mathbf{x}[k]^T\right\} \\ &= \sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]\big\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T\big\},\end{aligned}$$

$$\begin{aligned}\nabla^2_{\mathbf{v}} l(\mathbf{v}_j) = \frac{\partial^2 l(\mathbf{v}_j)}{\partial \mathbf{v}_j^2} &= \sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 G''\left(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)}\right) \\ &= \sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V} + \mathbf{b}^{(t)}\big).\end{aligned}$$

Then, the update equation is given as follows for $j = 1, \ldots, q$:

$$\begin{aligned}\mathbf{v}_j^{(t+1),T} = \mathbf{v}_j^{(t),T} - \alpha_{\mathbf{v}}^{(t+1)} & \left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)\right)^{-1} \\ & \cdot \left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]\big\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T\big\}\right).\end{aligned}$$

For the last step, as for $\left\{\mathbf{v}_j^{(\mathbf{1})}\right\}_{j=1}^{q}$ and $\left\{\mathbf{v}_j^{(\mathbf{2})}\right\}_{j=1}^{q}$, the derivations are made for the vector $\mathbf{b}$ indistinct of the mixture superscript. The update equation can then be used for $\mathbf{b}^{(\mathbf{1})}$ and $\mathbf{b}^{(\mathbf{2})}$ by changing $\mathbf{b}$ to $\mathbf{b}^{(\mathbf{1})}$, respectively to $\mathbf{b}^{(\mathbf{2})}$, $\mathbf{V}$ to $\mathbf{V}^{(\mathbf{1})}$, respectively to $\mathbf{V}^{(\mathbf{2})}$, $G(\cdot), G'(\cdot)$, and $G''(\cdot)$ to $G^{(\mathbf{1})}(\cdot), G^{(\mathbf{1})'}(\cdot)$, and $G^{(\mathbf{1})''}(\cdot)$, respectively to $G^{(\mathbf{2})}(\cdot), G^{(\mathbf{2})'}(\cdot)$, and $G^{(\mathbf{2})''}(\cdot)$.

$$
\begin{aligned}
l(\mathbf{b}) &= \sum_{k=1}^{n} \Big\{ G\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) - \big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big)\mathbf{x}[k]^T \Big\}, \\
\nabla_{\mathbf{b}} l(\mathbf{b}) &= \frac{\partial l(\mathbf{b})}{\partial \mathbf{b}} = \sum_{k=1}^{n} \Big\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) - \mathbf{x}[k]^T \Big\}, \\
\nabla_{\mathbf{b}}^2 l(\mathbf{b}) &= \frac{\partial^2 l(\mathbf{b})}{\partial \mathbf{b}^2} = \sum_{k=1}^{n} G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big).
\end{aligned}
$$

Then, the update equation is given as follows:

$$
\begin{aligned}
\mathbf{b}^{(t+1),T} = \mathbf{b}^{(t),T} - \alpha_{\mathbf{b}}^{(t+1)} &\left(\sum_{k=1}^{n} G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) \right)^{-1} \\
&\cdot \left(\sum_{k=1}^{n} \big\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) - \mathbf{x}[k]^T \big\} \right).
\end{aligned}
$$

4.3.1 Penalty function

The penalty function should dependent on the attributes distributions, hence, a mixed data-type case necessitates an appropriate penalty function that takes into account the requirements of the specific distributions in use. Considering that the f first attributes are distributed according to the exponential family distribution $p^{(\mathbf{1})}$ and the $(d-f)$ last attributes are distributed according to the exponential family distribution $p^{(\mathbf{2})}$, the penalty function takes the following form for $\underline{\boldsymbol{\theta}} = [\underline{\theta}_1, \ldots, \underline{\theta}_d]$:

$$
\begin{aligned}
\psi(\underline{\boldsymbol{\theta}}) = &\sum_{i=1}^{f} \Big\{ \exp\big(-\beta_{min}^{(\mathbf{1})}(\underline{\theta}_i - \theta_{min}^{(\mathbf{1})}) \big) + \exp\big(\beta_{max}^{(\mathbf{1})}(\underline{\theta}_i - \theta_{max}^{(\mathbf{1})}) \big) \Big\} \\
&+ \sum_{i=f+1}^{d} \Big\{ \exp\big(-\beta_{min}^{(\mathbf{2})}(\underline{\theta}_i - \theta_{min}^{(\mathbf{2})}) \big) + \exp\big(\beta_{max}^{(\mathbf{2})}(\underline{\theta}_i - \theta_{max}^{(\mathbf{2})}) \big) \Big\},
\end{aligned}
\tag{4.33}
$$

where the penalty function parameters are $\beta_{min}^{(\mathbf{1})}$ and $\theta_{min}^{(\mathbf{1})}$ for attributes distributed according to $p^{(\mathbf{1})}$, and $\beta_{min}^{(\mathbf{2})}$ and $\theta_{min}^{(\mathbf{2})}$ for attributes distributed according to $p^{(\mathbf{2})}$.

Since

$$\underline{\boldsymbol{\Theta}} = \begin{pmatrix} \underline{\boldsymbol{\theta}}[1] \\ \underline{\boldsymbol{\theta}}[2] \\ \vdots \\ \underline{\boldsymbol{\theta}}[n] \end{pmatrix} = \left(\begin{array}{ccc|ccc} \underline{\theta}_1[1] & \dots & \underline{\theta}_f[1] & \underline{\theta}_{f+1}[1] & \dots & \underline{\theta}_d[1] \\ \underline{\theta}_1[2] & \dots & \underline{\theta}_f[2] & \underline{\theta}_{f+1}[2] & \dots & \underline{\theta}_d[2] \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \underline{\theta}_1[n] & \dots & \underline{\theta}_f[n] & \underline{\theta}_{f+1}[n] & \dots & \underline{\theta}_d[n] \end{array}\right) = \left(\underline{\boldsymbol{\Theta}}^{(\mathbf{1})} \;\middle|\; \underline{\boldsymbol{\Theta}}^{(\mathbf{2})}\right),$$

with

$$\boldsymbol{\Theta}_{min} = \begin{pmatrix} \boldsymbol{\theta}_{min} \\ \boldsymbol{\theta}_{min} \\ \vdots \\ \boldsymbol{\theta}_{min} \end{pmatrix} = \left(\begin{array}{ccc|ccc} \theta_{min,1} & \dots & \theta_{min,f} & \theta_{min,f+1} & \dots & \theta_{min,d} \\ \theta_{min,1} & \dots & \theta_{min,f} & \theta_{min,f+1} & \dots & \theta_{min,d} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \theta_{min,1} & \dots & \theta_{min,f} & \theta_{min,f+1} & \dots & \theta_{min,d} \end{array}\right),$$

i.e.,

$$\boldsymbol{\Theta}_{min} = \left(\boldsymbol{\Theta}_{min}^{(\mathbf{1})} \;\middle|\; \boldsymbol{\Theta}_{min}^{(\mathbf{2})}\right),$$

and

$$\boldsymbol{\Theta}_{max} = \left(\boldsymbol{\Theta}_{max}^{(\mathbf{1})} \;\middle|\; \boldsymbol{\Theta}_{max}^{(\mathbf{2})}\right),$$

then,

$$\underline{\boldsymbol{\Theta}} - \boldsymbol{\Theta}_{min} = \left(\underline{\boldsymbol{\Theta}}^{(\mathbf{1})} - \boldsymbol{\Theta}_{min}^{(\mathbf{1})} \;\middle|\; \underline{\boldsymbol{\Theta}}^{(\mathbf{2})} - \boldsymbol{\Theta}_{min}^{(\mathbf{2})}\right),$$

and

$$\underline{\boldsymbol{\Theta}} - \boldsymbol{\Theta}_{max} = \left(\underline{\boldsymbol{\Theta}}^{(\mathbf{1})} - \boldsymbol{\Theta}_{max}^{(\mathbf{1})} \;\middle|\; \underline{\boldsymbol{\Theta}}^{(\mathbf{2})} - \boldsymbol{\Theta}_{max}^{(\mathbf{2})}\right).$$

Hence, the penalty function takes the following form:

$$\psi(\underline{\boldsymbol{\Theta}}) = \psi^{(\mathbf{1})}(\underline{\boldsymbol{\Theta}}^{(\mathbf{1})}) + \psi^{(\mathbf{2})}(\underline{\boldsymbol{\Theta}}^{(\mathbf{2})}),$$

where

$$\psi^{(\mathbf{1})}(\underline{\boldsymbol{\Theta}}^{(\mathbf{1})}) = \sum_{i=1}^{f} \left\{ \exp\left(-\beta_{min}^{(\mathbf{1})}(\underline{\theta}_i - \theta_{min}^{(\mathbf{1})})\right) + \exp\left(\beta_{max}^{(\mathbf{1})}(\underline{\theta}_i - \theta_{max}^{(\mathbf{1})})\right)\right\},$$

$$\psi^{(\mathbf{2})}(\underline{\boldsymbol{\Theta}}^{(\mathbf{2})}) = \sum_{i=f+1}^{d} \left\{ \exp\left(-\beta_{min}^{(\mathbf{2})}(\underline{\theta}_i - \theta_{min}^{(\mathbf{2})})\right) + \exp\left(\beta_{max}^{(\mathbf{2})}(\underline{\theta}_i - \theta_{max}^{(\mathbf{2})})\right)\right\}.$$

The loss function takes the following form:

$$\begin{aligned} L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) &= -\sum_{k=1}^{n} \log p\big(\mathbf{x}[k]|\underline{\mathbf{a}}[k], \mathbf{V}, \mathbf{b}\big) + c \cdot \sum_{k=1}^{n} \psi\big(\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}\big) \\ &= -\sum_{k=1}^{n} \log p^{(\mathbf{1})}\big(\mathbf{x}^{(\mathbf{1})}[k]|\underline{\mathbf{a}}[k], \mathbf{V}^{(\mathbf{1})}, \mathbf{b}^{(\mathbf{1})}\big) + c \cdot \sum_{k=1}^{n} \psi^{(\mathbf{1})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})} + \mathbf{b}^{(\mathbf{1})}\big) \\ &\quad - \sum_{k=1}^{n} \Big\{ \log p^{(\mathbf{2})}\big(\mathbf{x}^{(\mathbf{2})}[k]|\underline{\mathbf{a}}[k], \mathbf{V}^{(\mathbf{2})}, \mathbf{b}^{(\mathbf{2})}\big) \\ &\quad + c \cdot \sum_{k=1}^{n} \psi^{(\mathbf{2})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})} + \mathbf{b}^{(\mathbf{2})}\big), \end{aligned}$$

and the minimization problem can be expressed as:

$$\begin{aligned} &\arg\min_{\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}} L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) \\ &= \arg\min_{\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}} \Bigg\{ \sum_{k=1}^{n} \Big\{ G^{(\mathbf{1})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})} + \mathbf{b}^{(\mathbf{1})}\big) - \big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})} + \mathbf{b}^{(\mathbf{1})}\big)\mathbf{x}^{(\mathbf{1})}[k]^T \\ &+ c \cdot \sum_{k=1}^{n} \psi^{(\mathbf{1})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})} + \mathbf{b}^{(\mathbf{1})}\big) \Big\} \\ &+ \sum_{k=1}^{n} \Big\{ G^{(\mathbf{2})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})} + \mathbf{b}^{(\mathbf{2})}\big) - \big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})} + \mathbf{b}^{(\mathbf{2})}\big)\mathbf{x}^{(\mathbf{2})}[k]^T \\ &+ c \cdot \sum_{k=1}^{n} \psi^{(\mathbf{2})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})} + \mathbf{b}^{(\mathbf{2})}\big) \Big\} \Bigg\}. \end{aligned}$$

Since the linear combination of convex functions with nonnegative coefficients is always convex [84], the loss function remains convex in either of its arguments with the others fixed. Therefore, the iterative minimization technique proposed for the single exponential family can be applied in the mixture of exponential families case.

The first step in the Newton-Raphson minimization technique is, given a fixed matrix $\mathbf{V}$ and fixed vector $\mathbf{b}$, to obtain the matrix $\underline{\mathbf{A}}$, or the set of vectors $\underline{\mathbf{a}}[k]$ for $k = 1, \ldots, n$, which minimizes the loss function. The second step is, given a fixed matrix $\underline{\mathbf{A}}$ and fixed vector $\mathbf{b}$, to obtain the matrix $\mathbf{V}$ which minimizes the loss function. The last step is, given a fixed matrix $\underline{\mathbf{A}}$ and a fixed matrix $\mathbf{V}$, to obtain the vector $\mathbf{b}$. The updates are derived in a way similar to the one used in

Section 4.2. As previously, the superscript $^{(t)}$ means an estimate obtained at the end of the t^{th} iteration of the iterative minimization process:

$$\begin{aligned} l\big(\underline{\mathbf{a}}[k]\big) &= G^{(\mathbf{1})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big) - \big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big)\mathbf{x}^{(\mathbf{1})}[k]^T \\ &\quad + c \cdot \psi^{(\mathbf{1})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big) \\ &\quad + G^{(\mathbf{2})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big) - \big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big)\mathbf{x}^{(\mathbf{2})}[k]^T \\ &\quad + c \cdot \psi^{(\mathbf{2})}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big). \end{aligned}$$

Then,

$$\begin{aligned} \nabla_{\underline{\mathbf{a}}} l\big(\underline{\mathbf{a}}[k]\big) &= \frac{\partial l\big(\underline{\mathbf{a}}[k]\big)}{\partial \underline{\mathbf{a}}[k]} = \mathbf{V}^{(\mathbf{1})(t)} G^{(\mathbf{1})'}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big) - \mathbf{V}^{(\mathbf{1})(t)}\mathbf{x}^{(\mathbf{1})}[k]^T \\ &\quad + c \cdot \mathbf{V}^{(\mathbf{1})(t)} \psi^{(\mathbf{1})'}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big) \\ &\quad + \mathbf{V}^{(\mathbf{2})(t)} G^{(\mathbf{2})'}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big) - \mathbf{V}^{(\mathbf{2})(t)}\mathbf{x}^{(2)}[k]^T \\ &\quad + c \cdot \mathbf{V}^{(\mathbf{2})(t)} \psi^{(\mathbf{2})'}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big) \\ &= \mathbf{V}^{(\mathbf{1})(t)}\Big\{ G^{(\mathbf{1})'}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big) - \mathbf{x}^{(\mathbf{1})}[k]^T \\ &\quad + c \cdot \psi^{(\mathbf{1})'}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big)\Big\} \\ &\quad + \mathbf{V}^{(\mathbf{2})(t)}\Big\{ G^{(\mathbf{2})'}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big) - \mathbf{x}^{(\mathbf{2})}[k]^T \\ &\quad + c \cdot \psi^{(\mathbf{2})'}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big)\Big\}, \\ \nabla^2_{\underline{\mathbf{a}}} l\big(\underline{\mathbf{a}}[k]\big) &= \frac{\partial^2 l\big(\underline{\mathbf{a}}[k]\big)}{\partial \underline{\mathbf{a}}[k]^2} = \mathbf{V}^{(\mathbf{1})(t)}\Big\{ G^{(\mathbf{1})''}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big) \\ &\quad + c \cdot \psi^{(\mathbf{1})''}\ big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)}\big)\Big\}\mathbf{V}^{(\mathbf{1})(t),T} \\ &\quad + \mathbf{V}^{(\mathbf{2})(t)}\Big\{ G^{(\mathbf{2})''}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big) \\ &\quad + c \cdot \psi^{(\mathbf{2})''}\big(\underline{\mathbf{a}}[k]\mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)}\big)\Big\}\mathbf{V}^{(\mathbf{2})(t),T}. \end{aligned}$$

The update equation for the set of vectors $\underline{\mathbf{a}}[k]$ for $k = 1, \cdots, n$ is:

$$\begin{aligned}
\underline{\mathbf{a}}^{(t+1)}[k]^T = \underline{\mathbf{a}}^{(t)}[k]^T & \\
& - \alpha_{\underline{\mathbf{a}}}^{(t+1)} \Big\{ \mathbf{V}^{(\mathbf{1})(t)} \Big(G^{(\mathbf{1})''} \big(\underline{\mathbf{a}}^{(t)}[k] \mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)} \big) \\
& + c \cdot \psi^{(\mathbf{1})''} \big(\underline{\mathbf{a}}[k] \mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(m)} \big) \Big) \mathbf{V}^{(\mathbf{1})(t),T} \\
& + \mathbf{V}^{(\mathbf{2})(t)} \Big(G^{(\mathbf{2})''} \big(\underline{\mathbf{a}}^{(t)}[k] \mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)} \big) \\
& + c \cdot \psi^{(\mathbf{2})''} \big(\underline{\mathbf{a}}[k] \mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)} \big) \Big) \mathbf{V}^{(\mathbf{2})(t),T} \Big\}^{-1} \\
& \cdot \Big\{ \mathbf{V}^{(\mathbf{1})(t)} \Big(G^{(\mathbf{1})'} \big(\underline{\mathbf{a}}^{(t)}[k] \mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)} \big) - \mathbf{x}[k]^{(\mathbf{1}),T} \\
& + c \cdot \psi^{(\mathbf{1})'} \big(\underline{\mathbf{a}}^{(t)}[k] \mathbf{V}^{(\mathbf{1})(t)} + \mathbf{b}^{(\mathbf{1})(t)} \big) \Big) \\
& + \mathbf{V}^{(\mathbf{2})(t)} \Big(G^{(\mathbf{2})'} \big(\underline{\mathbf{a}}^{(t)}[k] \mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)} \big) - \mathbf{x}[k]^{(\mathbf{2}),T} \\
& + c \cdot \psi^{(\mathbf{2})'} \big(\underline{\mathbf{a}}^{(t)}[k] \mathbf{V}^{(\mathbf{2})(t)} + \mathbf{b}^{(\mathbf{2})(t)} \big) \Big) \Big\}.
\end{aligned}$$

For the second step, the two sets of row vectors $\left\{\mathbf{v}_j^{(\mathbf{1})}\right\}_{j=1}^q$ and $\left\{\mathbf{v}_j^{(\mathbf{2})}\right\}_{j=1}^q$ are updated separately. For the sake of simplicity, the following derivations are made for the set $\left\{\mathbf{v}_j\right\}_{j=1}^q$ indistinctively of the mixture superscript. The update equation can then be used for $\left\{\mathbf{v}_j^{(\mathbf{1})}\right\}_{j=1}^q$ and $\left\{\mathbf{v}_j^{(\mathbf{2})}\right\}_{j=1}^q$ by changing $\mathbf{v}_j$ to $\mathbf{v}_j^{(\mathbf{1})}$, respectively to $\mathbf{v}_j^{(\mathbf{2})}$, $\mathbf{b}$ to $\mathbf{b}^{(\mathbf{1})}$, respectively to $\mathbf{b}^{(\mathbf{2})}$, $G(\cdot), G'(\cdot)$, and $G''(\cdot)$ to $G^{(\mathbf{1})}(\cdot), G^{(\mathbf{1})'}(\cdot)$, and $G^{(\mathbf{1})''}(\cdot)$, respectively to $G^{(\mathbf{2})}(\cdot), G^{(\mathbf{2})'}(\cdot)$, and $G^{(\mathbf{2})''}(\cdot)$.

$$\begin{aligned}
l(\mathbf{v}_j) = \sum_{k=1}^{n} \Big\{ & G\left(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k] \mathbf{v}_r + \mathbf{b}^{(t)} \right) - \left(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k] \mathbf{v}_r + \mathbf{b}^{(t)} \right) \mathbf{x}[k]^T \\
& + c \cdot \psi \left(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k] \mathbf{v}_r + \mathbf{b}^{(t)} \right) \Big\}, \\
\nabla_{\mathbf{v}} l(\mathbf{v}_j) = \frac{\partial l(\mathbf{v}_j)}{\partial \mathbf{v}_j} = \sum_{k=1}^{n} \Big\{ & \underline{a}_j^{(t+1)}[k] G' \left(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k] \mathbf{v}_r + \mathbf{b}^{(t)} \right) - \underline{a}_j^{(t+1)}[k] \mathbf{x}[k]^T \\
& + c \cdot \underline{a}_j^{(t+1)}[k] \psi \left(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k] \mathbf{v}_r + \mathbf{b}^{(t)} \right) \Big\},
\end{aligned}$$

$$\nabla^2_{\mathbf{v}} l(\mathbf{v}_j) = \frac{\partial^2 l(\mathbf{v}_j)}{\partial \mathbf{v}_j^2} = \sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 \Big\{ G''\left(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)} \right) + c \cdot \psi\left(\sum_{r=1}^{q} \underline{a}_r^{(t+1)}[k]\mathbf{v}_r + \mathbf{b}^{(t)} \right) \Big\}$$
$$= \sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 \Big\{ G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V} + \mathbf{b}^{(t)}\big) + c \cdot \psi''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V} + \mathbf{b}^{(t)}\big) \Big\}.$$

Then, the update equation is given as follows for $j = 1, \ldots, q$:

$$\mathbf{v}_j^{(t+1),T} = \mathbf{v}_j^{(t),T} - \alpha_{\mathbf{v}}^{(t+1)} \Bigg(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 \Big(G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) + c \cdot \psi''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \Big) \Bigg)^{-1}$$
$$\cdot \Bigg(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k] \Big(G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T + c \cdot \psi'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \Big) \Bigg).$$

For the last step, as for $\left\{\mathbf{v}_j^{(\mathbf{1})}\right\}_{j=1}^{q}$ and $\left\{\mathbf{v}_j^{(\mathbf{2})}\right\}_{j=1}^{q}$, the derivations are made for the vector $\mathbf{b}$ indistinctively of the mixture superscript. The update equation can then be used for $\mathbf{b}^{(\mathbf{1})}$ and $\mathbf{b}^{(\mathbf{2})}$ by changing $\mathbf{b}$ to $\mathbf{b}^{(\mathbf{1})}$, respectively to $\mathbf{b}^{(\mathbf{2})}$, $\mathbf{V}$ to $\mathbf{V}^{(\mathbf{1})}$, respectively to $\mathbf{V}^{(\mathbf{2})}$, $G(\cdot), G'(\cdot)$, and $G''(\cdot)$ to $G^{(\mathbf{1})}(\cdot), G^{(\mathbf{1})'}(\cdot)$, and $G^{(\mathbf{1})''}(\cdot)$, respectively to $G^{(\mathbf{2})}(\cdot), G^{(\mathbf{2})'}(\cdot)$, and $G^{(\mathbf{2})''}(\cdot)$.

$$l(\mathbf{b}) = \sum_{k=1}^{n} \Big\{ G\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) - \big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big)\mathbf{x}[k]^T + c \cdot \psi\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) \Big\},$$
$$\nabla_{\mathbf{b}} l(\mathbf{b}) = \frac{\partial l(\mathbf{b})}{\partial \mathbf{b}} = \sum_{k=1}^{n} \Big\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) - \mathbf{x}[k]^T + c \cdot \psi'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) \Big\},$$
$$\nabla^2_{\mathbf{b}} l(\mathbf{b}) = \frac{\partial^2 l(\mathbf{b})}{\partial \mathbf{b}^2}$$
$$= \sum_{k=1}^{n} \Big\{ G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) + c \cdot \psi''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) \Big\}.$$

Then, the update equation is given as follows:

$$\mathbf{b}^{(t+1),T} = \mathbf{b}^{(t),T} - \alpha_{\mathbf{b}}^{(t+1)} \Bigg(\sum_{k=1}^{n} G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) + c \cdot \psi''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) \Bigg)^{-1} \cdot \Bigg(\sum_{k=1}^{n} \big\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) - \mathbf{x}[k]^T + c \cdot \psi'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) \big\} \Bigg).$$

4.3.2 Synthetic data examples

Synthetic data examples for several exponential family distributions in the case of mixed data types are presented here. Figures in both data space and parameter space provide insight about the relationship between the low-dimensional parameter subspace originally used to create the synthetic data sets and the subspace estimated within the GLS framework.

Poisson-Gaussian mixed data set

Figure 4.13 presents a mixture of two Poisson-Gaussian mixed distributions in data space. The data are comprised of one Poisson attribute and two Gaussian attributes. Figure 4.14 shows the corresponding parameter space. The original 1-dimensional subspace containing the parameters of the mixed data is represented with a solid line and the subspace learned within the GLS framework with a dashdot line. The sine of the angle between the original subspace and the subspace learned with GLS is $\sin(\measuredangle_{GLS}) = 0.088601$.

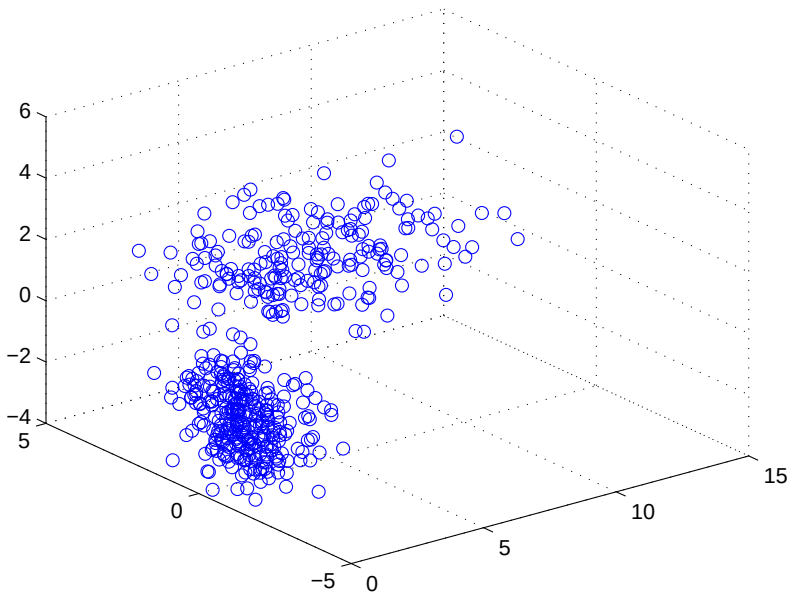

Figure 4.13 Data space: mixture of two Poisson-Gaussian mixed distributions with parameters constrained on a 1-dimensional subspace.

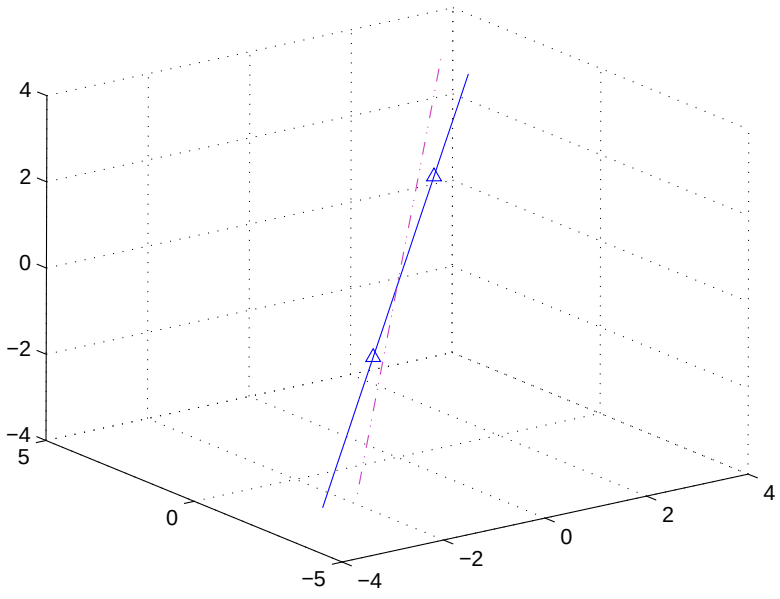

Figure 4.14 Parameter space: parameters ($\triangle$) of two Poisson-Gaussian mixed distributions, original 1-dimensional subspace (solid line) and 1-dimensional subspaces estimated with GLS (dashdot line).

Binomial-Gaussian mixed data set

Figure 4.15 presents a mixture of two Binomial-Gaussian mixed distributions in data space. The data are comprised of one Binomial attribute and two Gaussian attributes. Figure 4.16 shows the corresponding parameter space. The original 1-dimensional subspace containing the parameters of the mixed data is represented with a solid line and the subspace learned within the GLS framework with a dashdot line. The sine of the angle between the original subspace and the subspace learned with GLS is $\sin(\measuredangle_{GLS}) = 0.021565$.

Gamma-Gaussian mixed data set

Figure 4.17 presents a mixture of two Gamma-Gaussian mixed distributions in data space. The data are comprised of one Gamma attribute and two Gaussian attributes. The sine of the angle between the original subspace and the subspace learned with GLS is $\sin(\measuredangle_{GLS}) = 0.01724$.

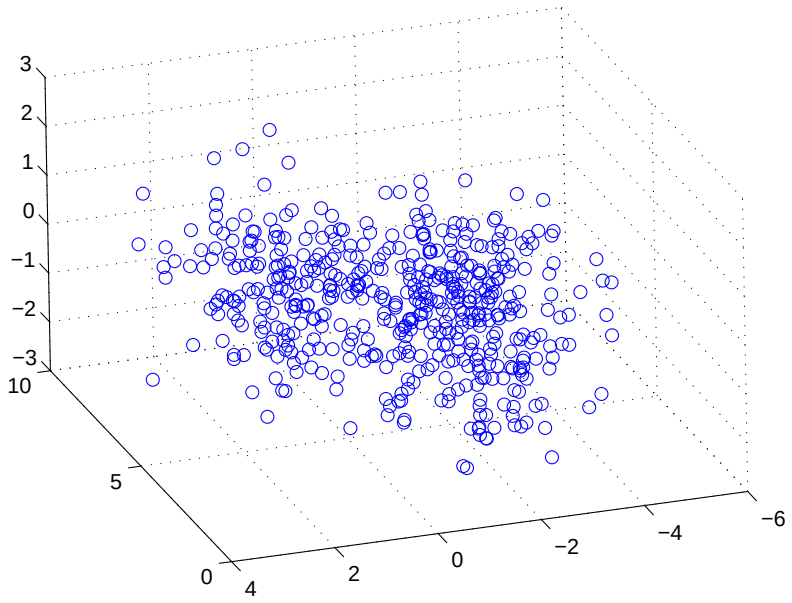

Figure 4.15 Data space: mixture of two Binomial-Gaussian mixed distributions with parameters constrained on a 1-dimensional subspace.

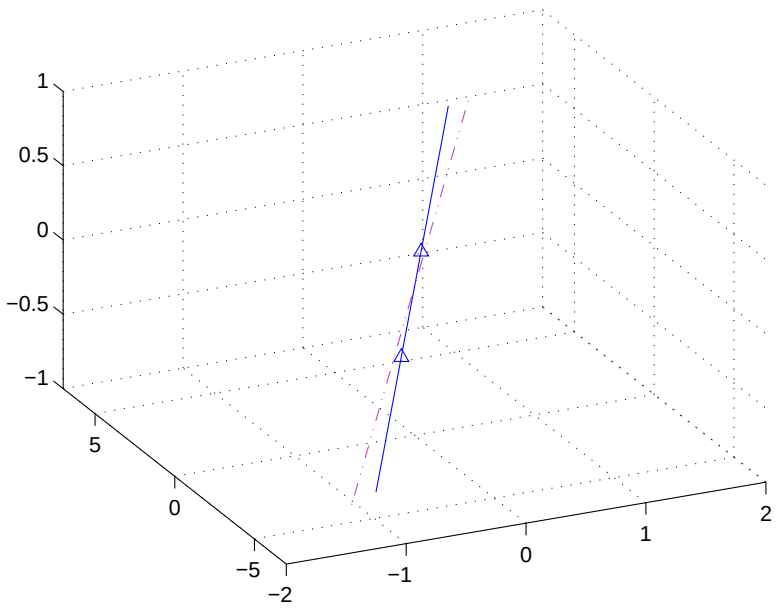

Figure 4.16 Parameter space: parameters (△) of two Binomial-Gaussian mixed distributions, original 1-dimensional subspace (solid line) and 1-dimensional subspaces estimated with GLS (dashdot line).

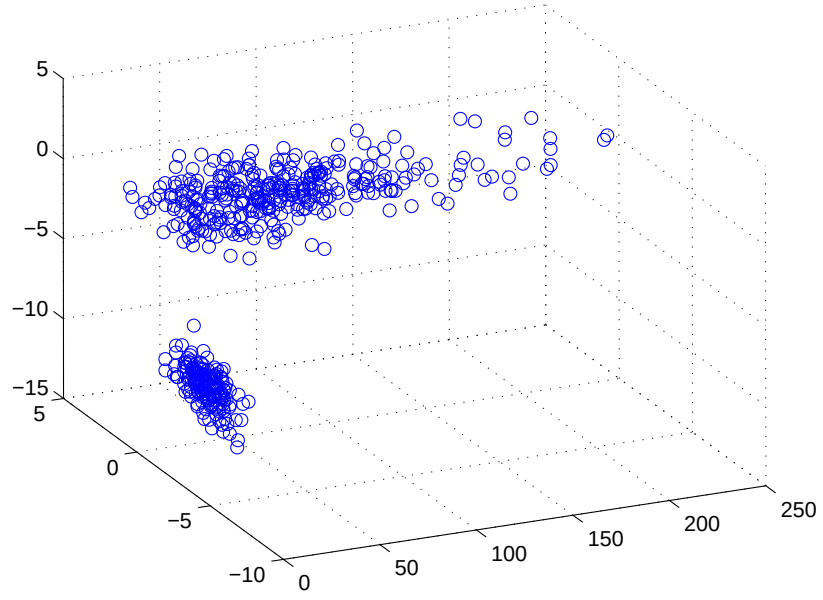

Figure 4.17 Data space: mixture of two Gamma-Gaussian mixed distributions with parameters constrained on a 1-dimensional subspace.

Binomial-Gamma-Gaussian mixed data set

Figure 4.18 presents a mixture of two Binomial-Gaussian-Gamma mixed distributions in data space. The data are comprised of one Gamma attribute, one Gaussian attribute and one Binomial attribute ($N = 10$). The sine of the angle between the original subspace and the subspace learned with a Binomial-Gaussian-Gamma GLS assumption is $\sin(\measuredangle_{GLS:BGG}) = 0.0326$. With a simple Gaussian GLS assumption, the sine becomes $\sin(\measuredangle_{GLS:G}) = 0.93375 > \sin(\measuredangle_{GLS:BGG})$. Since both a Binomial random variable and a Gamma random variable only take on positive values, we could only consider two more assumptions: a Binomial-Gaussian GLS assumption and a Gaussian-Gamma GLS assumption. The sine of the angle between the original subspace and the subspace learned with a Binomial-Gaussian GLS assumption is $\sin(\measuredangle_{GLS:BG}) = 0.93623$ and the sine of the angle between the original subspace and the subspace learned with a Gaussian-Gamma GLS assumption is $\sin(\measuredangle_{GLS:GG}) = 0.08310$. Hence, a Gaussian-Gamma GLS assumption yields results that are close to the results obtained with the Binomial-Gaussian-Gamma GLS assumption. In this particular example, assuming a Gamma distribution for the last attribute seems to be essential for a good estimation performance. We performed a similar experiment with a mixture of two Binomial-Gaussian-Gamma mixed distributions and the Binomial parameter N equal to 5. The results are similar to the ones obtained for the data with Binomial parameter N equal to 10 and are as follows: $\sin(\measuredangle_{GLS:G}) = 0.86571 > \sin(\measuredangle_{GLS:BG}) = 0.86359 > \sin(\measuredangle_{GLS:GG}) = 0.14198 > \sin(\measuredangle_{GLS:BGG}) = 0.098467$.

4.4 Application: unsupervised minority class detection in parameter space on synthetic data

Minority class detection considers a binary class situation where a "minority class" is discriminated from a "majority class". It aims to differentiate rare key events belonging to the minority class from the remainder of the data belonging

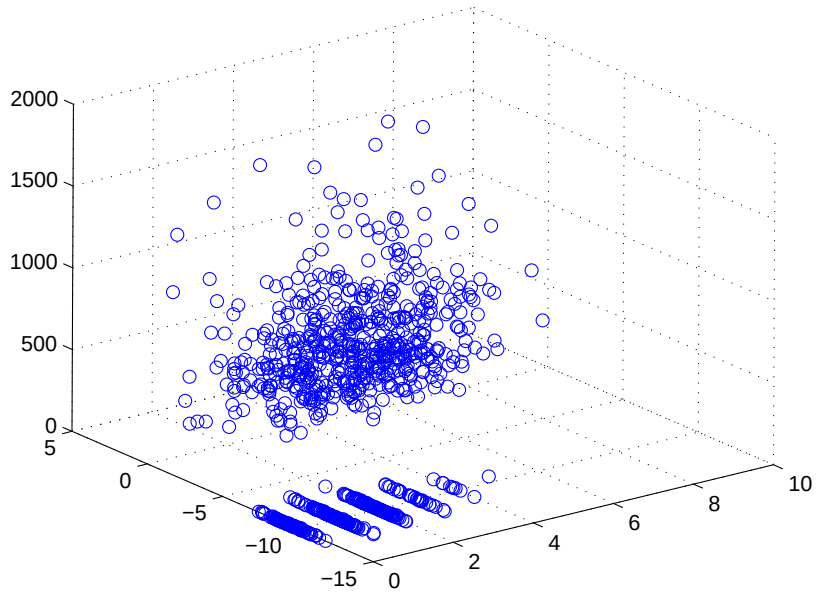

Figure 4.18 Data space: a mixture of two Binomial-Gaussian-Gamma mixed distributions with parameters constrained on a 1-dimensional subspace.

to the majority class.

The problem of unsupervised data-driven minority class (rare event) detection is one of relating property descriptors of a large unlabeled database of "objects" to measured properties of these objects, then using these empirically determined relationships to infer the properties of new objects. Here, the ultimate goal is to correctly characterize the new objects as either belonging to the minority class or not. This work assumes that minority class and majority class objects constitute two distinct, well-separated classes of objects in a latent variable subspace of the parameter space as described in Section 3. In the case of a rare occurrence of objects to be detected, it is believed that modeling the total unlabeled database allows one to discern the statistical structure of the majority class of objects. This experiment considers measured object properties that are non-Gaussian, mixed (comprised of continuous and discrete data), very noisy, and highly non-linearly related for which the resulting minority class detection problem is very difficult.

Unsupervised methods for feature extraction, such as Principal Compo-

nent Analysis (PCA), are commonly used to process data before using discriminative classifiers, such as Support Vector Machines (SVMs) or neural networks. However, methods such as Independent Component Analysis (ICA) and PCA assume the same form of the distribution for all components of the data. In contrast, the Generalized Linear Statistics (GLS) framework developed in Section 3 allows each component to have its own parametric form. The proposed minority class detection technique is based on the GLS framework, enabling the use of exponential family distributions to model the various mixed types of data measurements (continuous or discrete). A key aspect is that the parameters of the exponential family distributions are constrained to a lower dimensional latent variable subspace to model the belief that the intrinsic dimensionality of the data is smaller than the dimensionality of the data space. The proposed minority class detection technique is performed in parameter space rather than in data space, as in more classical approaches, and exploits the low dimensional information provided by the latent variables $\underline{\mathbf{a}}[k]$, $k = 1, \ldots, n$.

Figure 4.19 shows an example of synthetic three-dimensional mixed data ($d = 3$), with each data sample comprised of a Binomial component with values between 0 and 5, an Exponential distribution component, and a Gaussian component. The data are generated by two different classes, a minority one and a majority one, and for each class the parameters are assumed to be constrained to lie on a (different) one-dimensional subspace of the parameter space ($q = 1$). To assess the unsupervised minority class detection performance, we consider a situation where the minority class is a rare occurrence (1 percent of 10000 data samples), and the data are equally divided into a training set and a test set. The unsupervised minority class detection technique using the GLS information learned in parameter space works as follows: first, given the training set $\{\mathbf{x}[k]\}_{k=1}^{n}$, we learn the low-dimensional parameter subspace or direction of projection in parameter space, namely the matrix $\mathbf{V}$, by using the GLS modeling approach, and compute the training set mean-image on the lower dimensional parameter subspace, namely

$1/n \sum_{k=1}^{n} \underline{\mathbf{a}}[k]$. The training set mean-image is then taken as an approximation to the training cluster mean of the majority class in the lower dimensional subspace. Then, for each test sample, the data point is moved from data space to parameter space using the inverse link function. We project the obtained point onto the direction of projection estimated by GLS and compute its distance to the training set mean-image. Finally, we compare the obtained distance to a given threshold λ to make a decision. The test point is declared to be part of the minority class if the distance is higher than λ, otherwise it is declared to be part of the majority class. This procedure is conducted for all of the test set samples, and the detection performance is assessed by plotting the ROC curve found from varying the value of λ. The ROC curve shows the probability of detection P_D versus the probability of false alarm P_{FA} as λ varies. The proposed technique is compared to classical PCA used in data space with a threshold test performed on new test data projected along the first principal axis, as well as to a supervised Bayes (minimum rate) detector for the sake of an optimal benchmark.

Data for which classical PCA will fail to provide accurate detection are easily created, using the knowledge that classical PCA defines the direction of projection as the direction of maximum variance in data space. The classical PCA approach will therefore give poor performance on data for which the direction of maximum variance is inappropriate for separating minority from majority class data. The Exponential distribution $p(x;\theta) = \beta \exp(-\beta x)$, with $\theta = -\beta$, is used as a component of the data. Because the link function for this distribution is $f(x) = -1/x$, *the direction of maximum variance in data space is actually the direction of minimum variance in feature space,* and for this situation classical PCA is expected to perform poorly, and indeed it does.

Figure 4.20 shows a comparison between the supervised Bayes detector, the minority class detector based on GLS information and performed in parameter space, and the minority class detector based on classical PCA information and performed in data space. This illuminating example shows that there are domains

for which classical PCA performs far from optimal.

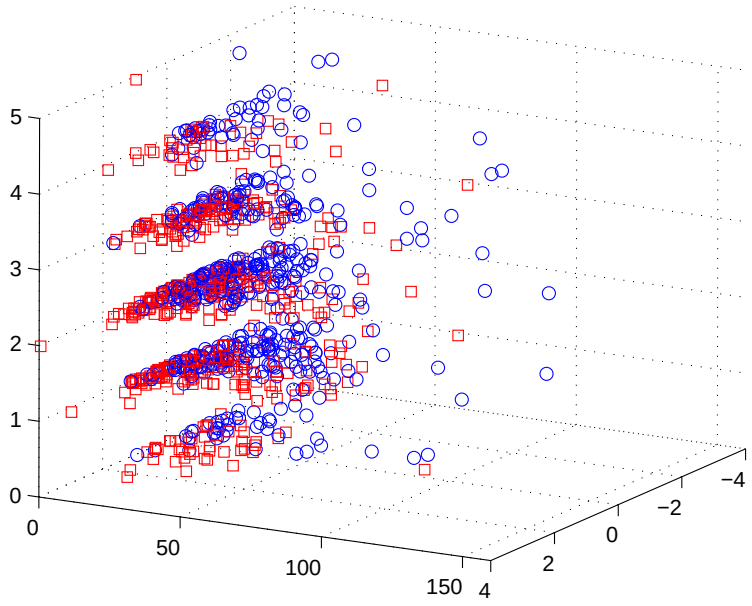

Figure 4.19 Data space: data samples of a 3-dimensional mixed data set with Binomial, Exponential and Gaussian components (blue circles for one class and red squares for the other class).

Acknowledgement

Chapter 4, in part, is a reprint of the material as it appears in "Data-pattern discovery methods for detection in nongaussian high-dimensional data sets," C. Levasseur, K. Kreutz-Delgado, U. Mayer and G. Gancarz, in *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, pp. 545–549, Nov. 2005, "Generalized statistical methods for unsupervised minority class detection in mixed data sets," C. Levasseur, U. F. Mayer, B. Burdge and K. Kreutz-Delgado, in *Proceedings of the First IAPR Workshop on Cognitive Information Processing (CIP)*, pp. 126–131, June 2008, "Generalized statistical methods for mixed exponential families, part I: theoretical foundations," C. Levasseur, K. Kreutz-Delgado and U. F. Mayer, submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Sept. 2009 and "Generalized

statistical methods for mixed exponential families, part II: applications," C. Levasseur, U. F. Mayer and K. Kreutz-Delgado, submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Sept. 2009. The dissertation author was the primary author of these papers.

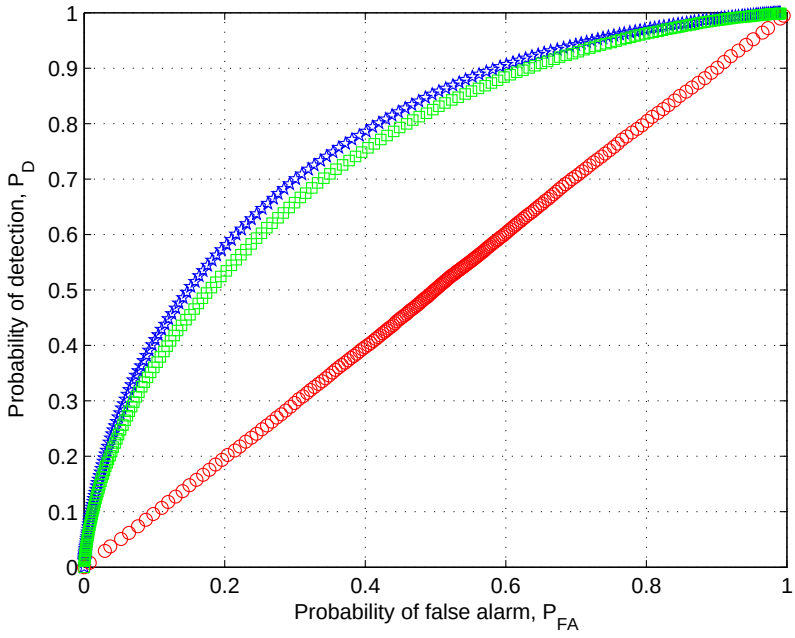

Figure 4.20 Comparison of supervised Bayes optimal (top blue with pentagrams), proposed GLS technique (middle green with squares) and classical PCA (bottom red with circles) ROC curves.

5 A unifying viewpoint and extensions to mixed data sets

This section presents a general point of view that relates the exponential family Principal Component Analysis (exponential PCA) technique of [10] to both the Semi-Parametric exponential family Principal Component Analysis (SP-PCA) technique of [15, 82] and the Bregman soft clustering method presented in [14, 16]. The proposed viewpoint is then illustrated with a clustering problem in mixed data sets.

The three techniques considered here all utilize Bregman distances and can all be explained within a single hierarchical Bayes graphical model framework shown in Figure 3.2. They are not separate unrelated algorithms but different manifestations of model assumptions and parameter choices taken within a common framework. The proposed model is mathematically equivalent to equation (5.6) and this work demonstrates that selecting a Bayesian or a classical approach as well as various parametric choices sketched in Figure 5.1, Figure 5.2 and Figure 5.3 determine the three algorithms. Because of this insight, these algorithms are readily extended to deal with the important mixed data-type case.

5.1 Theoretical background

Following the Bayesian approach presented in Section 3, the maximum likelihood identification of the blind random effect model

$$p(\mathbf{x}) = \int p(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} = \int \prod_{i=1}^{d} p_i(x_i|\theta_i)\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}, \tag{5.1}$$

is a quite difficult problem. It corresponds to identifying $\pi(\boldsymbol{\theta})$, which, under the condition $\boldsymbol{\theta} = \mathbf{aV} + \mathbf{b}$, corresponds to identifying the matrix $\mathbf{V}$, the vector $\mathbf{b}$, and a density function, $\mu(\mathbf{a})$, on the random effect $\mathbf{a}$ via a maximization of the likelihood function $p(\mathbf{X})$ with respect to $\mathbf{V}$, $\mathbf{b}$, and $\mu(\mathbf{a})$, where

$$\begin{aligned} p(\mathbf{X}) &= \prod_{k=1}^{n} p\big(\mathbf{x}[k]\big) = \prod_{k=1}^{n} \int p\big(\mathbf{x}[k]|\boldsymbol{\theta}\big)\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \\ &= \prod_{k=1}^{n} \int \prod_{i=1}^{d} p_i\big(x_i[k]|\theta_i\big)\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}, \end{aligned} \tag{5.2}$$

and $\mathbf{X}$ is the $(n \times d)$ observation matrix

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}[1] \\ \mathbf{x}[2] \\ \vdots \\ \mathbf{x}[n] \end{pmatrix} = \begin{pmatrix} x_1[1] & \dots & x_d[1] \\ x_1[2] & \dots & x_d[2] \\ \vdots & \ddots & \vdots \\ x_1[n] & \dots & x_d[n] \end{pmatrix}.$$

In the simpler case of a single common exponential family distribution for all components, i.e.,

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(x_1|\boldsymbol{\theta}) \cdot \ldots \cdot p(x_d|\boldsymbol{\theta}) = p(x_1|\theta_1) \cdot \ldots \cdot p(x_d|\theta_d) = \prod_{i=1}^{d} p(x_i|\theta_i), \tag{5.3}$$

it can be shown that, if the distribution $\pi(\boldsymbol{\theta})$ of the random parameter vector $\boldsymbol{\theta}$ is conjugate to the exponential family distribution $p(\mathbf{x}|\boldsymbol{\theta})$, then maximum likelihood methods are straightforward in principle from the marginal distribution of the observation matrix [27,29]. However, the conjugate approach lacks generality since for the general case a different conjugate distribution has to be assumed for each exponential family distribution.

In some specific applications, the marginal density is mathematically simple to treat [34]: for example, considering the simple one-dimensional case, if

$$p(x|\theta) = p(x|\lambda) = \lambda^x \mathrm{e}^{-\lambda}/x!$$

is a Poisson distribution with parameter $\lambda = \mathrm{e}^{\theta}$, the conjugate prior for λ is the Gamma distribution given by

$$\pi(\lambda) = \gamma^{-\kappa}\lambda^{\kappa-1}\mathrm{e}^{-\lambda/\gamma}/\Gamma(\kappa).$$

Then, the marginal density

$$\begin{aligned} p(x) &= \int_0^{+\infty} p(x|\lambda)\pi(\lambda)\mathrm{d}\lambda = \int_0^{+\infty} \lambda^x \mathrm{e}^{-\lambda}/x!\,\gamma^{-\kappa}\lambda^{\kappa-1}\mathrm{e}^{-\lambda/\gamma}/\Gamma(\kappa)\mathrm{d}\lambda \\ &= \frac{\gamma^{-\kappa}}{\Gamma(x+1)\Gamma(\kappa)} \int_0^{+\infty} \lambda^{x+\kappa-1}\mathrm{e}^{-\lambda(1+1/\gamma)}\mathrm{d}\lambda, \end{aligned}$$

with $\Gamma(x) = (x-1)!$,

$$= \frac{\gamma^{-\kappa}}{\Gamma(x+1)\Gamma(\kappa)} \int_0^{+\infty} \mathrm{e}^{-z} z^{x+\kappa-1} \big(\gamma/(\gamma+1)\big)^{x+\kappa-1} \gamma/(\gamma+1)\mathrm{d}z,$$

with $z = \lambda(1+1/\gamma) = \lambda(\gamma+1)/\gamma$ and $\mathrm{d}z = (\gamma+1)/\gamma\,\mathrm{d}\lambda$,

$$p(x) = \frac{(\gamma+1)^{-\kappa}\Gamma(x+\kappa)}{\Gamma(x+1)\Gamma(\kappa)} \big(\gamma/(\gamma+1)\big)^x$$

is a negative Binomial distribution because of the definition of the Gamma function $\int_0^{+\infty} \mathrm{e}^{-z} z^{x+\kappa-1}\mathrm{d}z = \Gamma(x+\kappa)$.

Another example goes as follows: if

$$p(x|\theta) = p(x|\lambda) = \frac{N!}{x!(N-x)!}\lambda^x(1-\lambda)^{N-x}$$

is a Binomial distribution with parameter $\lambda = \mathrm{e}^{\theta}/(1+\mathrm{e}^{\theta})$, the conjugate prior for λ is the Beta distribution given by

$$\pi(\lambda) = \Gamma(\alpha+\beta)/\big(\Gamma(\alpha)\Gamma(\beta)\big)\lambda^{\alpha-1}(1-\lambda)^{\beta-1}.$$

Then, the marginal density

$$\begin{aligned} p(x) &= \int_{-\infty}^{+\infty} p(x|\lambda)\pi(\lambda)\mathrm{d}\lambda \\ &= \int_{-\infty}^{+\infty} \frac{N!}{x!(N-x)!}\lambda^x(1-\lambda)^{N-x}\Gamma(\alpha+\beta)/\big(\Gamma(\alpha)\Gamma(\beta)\big)\lambda^{\alpha-1}(1-\lambda)^{\beta-1}\mathrm{d}\lambda \\ &= \frac{N!}{x!(N-x)!}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\int_{-\infty}^{+\infty}\lambda^{x+\alpha-1}(1-\lambda)^{N-x+\beta-1}\mathrm{d}\lambda \\ &= \frac{N!}{x!(N-x)!}\frac{\Gamma(\alpha+\beta)\Gamma(x+\alpha)\Gamma(N-x+\beta)}{\Gamma(\alpha)\Gamma(\beta)\Gamma(x+\alpha+N-x+\beta)} \end{aligned}$$

is a special distribution called the Beta-Binomial distribution because

$$\int_{-\infty}^{+\infty}\frac{\Gamma(x+\alpha+N-x+\beta)}{\Gamma(x+\alpha)\Gamma(N-x+\beta)}\lambda^{x+\alpha-1}(1-\lambda)^{N-x+\beta-1}\mathrm{d}\lambda = 1 \tag{5.4}$$

as an integral over the Beta distribution with parameters $(x+\alpha)$ and $(N-x+\beta)$.

A more appealing approach would be to assume a common distribution for the random effect across the exponential family; an obvious choice is the Gaussian distribution. This is particularly natural for link functions generating an unbounded parameter space for the linear predictor. This approach calls for a Gaussian quadrature approximation based on the Expectation-Maximization (EM) algorithm [68], an approximation that does not easily yield correct maximum likelihood estimates [27, 29]. This difficulty can be avoided by Non-Parametric Maximum Likelihood (NPML) estimation of the random effect distribution, concurrently with the structural model parameters (further discussed in Appendix C).

The Non-Parametric Maximum Likelihood (NPML) estimate is known to be a discrete distribution on a finite number of support points or "atoms" [17, 20, 25]. Finding the NPML estimate is widely regarded as computationally intensive, the particular difficulty being the location of the atoms [27].

As shown in Section 3, with $\boldsymbol{\theta} = \mathbf{a}\mathbf{V} + \mathbf{b}$, with $\mathbf{V}$, $\mathbf{b}$ fixed and $\mathbf{a}$ random, the single-sample likelihood (5.1) is then equal to

$$p(\mathbf{x}) = \sum_{l=1}^{m} p\big(\mathbf{x}|\underline{\boldsymbol{\theta}}[l]\big)\pi_l = \sum_{l=1}^{m} p\big(\mathbf{x}|\underline{\mathbf{a}}[l]\mathbf{V}+\mathbf{b}\big)\pi_l \tag{5.5}$$

and the data likelihood (5.2) is equal to

$$p(\mathbf{X}) = \prod_{k=1}^{n} \sum_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l$$
$$= \prod_{k=1}^{n} \sum_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big)\pi_l, \tag{5.6}$$

with point-mass probability estimates π_l, point-mass support points $\underline{\mathbf{a}}[l]$, and linear predictor in the lth mixture component $\underline{\boldsymbol{\theta}}[l] = \underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}$, $l = 1, \ldots, m$.

The data likelihood is thus approximately the likelihood of a finite mixture of exponential family densities with unknown mixture proportions or point-mass probability estimates π_l and unknown point-mass support points $\underline{\mathbf{a}}[l]$, with the linear predictor $\underline{\boldsymbol{\theta}}[l] = \underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}$ in the lth mixture component [29]. The combined problem of Maximum Likelihood Estimation (MLE) of the parameters $\mathbf{V}$, $\mathbf{b}$, the point-mass support points (atoms) $\underline{\mathbf{a}}[l]$ and the point-mass probability estimates $\pi_l, l = 1, \ldots, m$, (as approximations to the unknown, and possibly continuous density $\mu(\mathbf{a})$) is known as the Semiparametric Maximum Likelihood mixture density Estimation (SMLE) problem [53, 66, 67]. For $m < n$, this problem can be attacked by using the Expectation-Maximization (EM) algorithm [1, 17, 20, 22, 29, 34, 53, 68–74], cf. in particular in the Semi-Parametric exponential family Principal Component Analysis (SP-PCA) technique proposed in [15,82] and discussed below. Note that, historically, Laird's classic 1978 paper [17] appears to be generally acknowledged as the first paper that proposed the EM algorithm for NPML estimation in the mixture density context; then, Lindsay's classic 1983 papers [20, 21] improved upon the theoretical foundations of the NPML estimation approach and later Mallet's 1986 paper [22] further explored some of the fundamental issues raised by Lindsay.

However, a classical approach to the GLS estimation problem can also be considered. The classical approach can be seen as an extreme case of the Bayesian approach for which the probability density function $\pi(\boldsymbol{\theta})$ is a delta function and the total number of distinct parameter values m (referred to in this dissertation as support points or atoms in both Bayesian and classical approaches) equals the

number of data points n, i.e., $m = n$. Then, to each data point corresponds a (generally different) parameter point, yielding a total of n points $\underline{\boldsymbol{\theta}}[k]$, $k = 1, \ldots, n$, in parameter space as presented in the exponential family Principal Component Analysis technique [10]. Note that while the $m < n$ parameter points of the Bayesian approach are shared by all the data points, the classical approach assigns one parameter point to each data point (hence $m = n$).

This section aims at presenting a general point of view, considers and compares both approaches, and relates them to a simpler Bregman soft clustering technique proposed in [14, 16].

5.2 Semi-Parametric exponential family PCA approach

The Semi-Parametric exponential family Principal Component Analysis (SP-PCA) approach presented in [15, 82] attacks the Semiparametric Maximum Likelihood mixture density Estimation (SMLE) problem by using the Expectation-Maximization (EM) algorithm [68]. A detailed derivation of the EM algorithm is presented below with a few changes compared to [15, 82].

Following the notations introduced in Appendix C, the *mixing distribution* is denoted by $\mathcal{Q} = \left\{\underline{\boldsymbol{\theta}}[l], \pi_l\right\}_{l=1}^{m}$ and encompasses the parameters $\underline{\boldsymbol{\theta}}[l], l = 1, \ldots, m$, and their associated point-mass probabilities $\pi_l, l = 1, \ldots, m$. Estimation is performed conventionally by maximum likelihood and the Non-Parametric Maximum Likelihood estimator is represented by $\widehat{\mathcal{Q}} = \left\{\widehat{\underline{\boldsymbol{\theta}}}[l], \widehat{\pi}_l\right\}_{l=1}^{m}$. The EM approach considers an *incomplete* log-likelihood function, which is defined by the following equation

$$L(\mathcal{Q}) = \log \prod_{k=1}^{n} \sum_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l = \sum_{k=1}^{n} \log \sum_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l. \tag{5.7}$$

Because $\pi_1 + \pi_2 + \cdots + \pi_m = 1$, π_m can be replaced by $1 - \sum_{l=1}^{m-1} \pi_l$ and the *incomplete* log-likelihood function can also be written as:

$$L(\mathcal{Q}) = \sum_{k=1}^{n} \log \left\{ \sum_{l=1}^{m-1} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l + p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}_m\big) \left(1 - \sum_{l=1}^{m-1} \pi_l\right) \right\}.$$

A *missing* (unobserved) variable $\mathbf{z}_k = [z_{k1}, \ldots, z_{km}]$, for $k = 1, \ldots, n$, is introduced; this variable is an m-dimensional binary vector whose lth component equals 1 if the response variable $\mathbf{x}[k]$ was drawn from the lth mixture component and 0 otherwise. Hence, a *complete* log-likelihood function is generated as follows:

$$L^{(c)}\big(\mathcal{Q}, \{\mathbf{z}_k\}_{k=1}^n\big) = \sum_{k=1}^{n} \log \prod_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)^{z_{kl}} \pi_l^{z_{kl}} \tag{5.8}$$

$$= \underline{\sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)} + \sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} \log \pi_l, \tag{5.9}$$

where the underlined term is independent of $\pi_l, l = 1, \ldots, m$. Because z_{kl} equals 1 exactly for one l if k is fixed, reflecting the assumption that each $\mathbf{x}[k]$ is drawn from exactly one mixture component, the inner sum in equation (5.7) has in fact for each k exactly one non-zero term. In equation (5.8) it is exactly that non-zero term which is present in the product, all others have an exponent of $z_{kl} = 0$, and hence do not contribute to the product. Note that maximizing equation (5.9) with respect to π_l yields $\widehat{\pi}_l = \sum_{k=1}^{n} z_{kl}/n$ for $l = 1, \ldots, m$, corresponding to the number of samples $\mathbf{x}[k]$ drawn from the lth mixture, divided by the number of samples overall.

First, the E-step, or Expectation-step, allows one to obtain an estimate of the missing variables $\mathbf{z}_k = [z_{k1}, \ldots, z_{km}]^T, k = 1, \ldots, n$, by replacing them with their expected values given the data set $\big\{\mathbf{x}[k]\big\}_{k=1}^n$:

$$\begin{aligned} \widehat{z}_{kl} = \mathrm{E}\left\{z_{kl}|\mathbf{x}[k], \pi_1, \ldots, \pi_m\right\} &= \Pr\big(z_{kl} = 1|\mathbf{x}[k]\big) \\ &= \frac{\Pr\big(\mathbf{x}[k]|z_{kl} = 1\big)\Pr(z_{kl} = 1)}{\sum_{r=1}^{m} \Pr\big(\mathbf{x}[k]|z_{kr} = 1\big)\Pr(z_{kr} = 1)} \\ &= \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r}, \end{aligned} \tag{5.10}$$

for $l = 1, \ldots, m$, where the notation $\Pr(\cdot)$ expresses the probability of an event. For all l and all k, each data point $\mathbf{x}[k]$ has an estimated probability $\widehat{z}_{kl}$ of belonging to the lth mixture component.

Then, the complete log-likelihood function becomes:

$$L^{(c)}\big(\mathcal{Q}, \{\widehat{\mathbf{z}}_k\}_{k=1}^n\big) = \sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big) + \sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl} \log \pi_l. \tag{5.11}$$

Finally, maximizing equation (5.11) leads to the M-step, or Maximization-step and yields the estimates for the point-mass probabilities:

$$\widehat{\pi}_l = \frac{\sum_{k=1}^{n} \widehat{z}_{kl}}{n} = \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l}{n\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r}. \tag{5.12}$$

Now, the component parameters $\underline{\boldsymbol{\theta}}[l] = \underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}, l = 1, \ldots, m$, are assumed to be unknown and need to be estimated in the M-step, i.e., the parameters $\mathbf{V}$, $\mathbf{b}$, and the point-mass support points $\underline{\mathbf{A}} = \big[\underline{\mathbf{a}}[1]^T, \ldots, \underline{\mathbf{a}}[m]^T\big]^T \in \mathbb{R}^{m,q}$ need to be estimated. Maximizing the complete log-likelihood function (5.11) with respect to these parameters is equivalent to:

$$\begin{aligned}
&\arg\max_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} L^{(c)}\big(\mathcal{Q}, \{\widehat{\mathbf{z}}_k\}_{k=1}^n\big) \\
&= \arg\max_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big) + \underline{\sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl} \log \widehat{\pi}_l} \\
&= \arg\max_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl} \log p\big(\mathbf{x}[k]|\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big)
\end{aligned}$$

since the underlined term does not depend on either $\mathbf{V}$, $\mathbf{b}$, or $\underline{\mathbf{A}}$,

$$= \arg\max_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl} \left\{\big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big)\mathbf{x}[k]^T - G\big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big)\right\}$$

where $G(\cdot)$ is the cumulant generating function associated with the exponential family distribution $p(\cdot)$,

$$\begin{aligned}
&= \arg\min_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl} \left\{G\big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big) - \big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big)\mathbf{x}[k]^T\right\} \\
&= \arg\min_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl} G\big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big) - \sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl}\big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big)\mathbf{x}[k]^T
\end{aligned}$$

$$= \arg\min_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \sum_{l=1}^{m} \left(\sum_{k=1}^{n} \widehat{z}_{kl} \right) G\big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big) - \sum_{l=1}^{m} \big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big) \left\{ \sum_{k=1}^{n} \widehat{z}_{kl}\, \mathbf{x}[k]^T \right\}$$

$$= \arg\min_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \sum_{l=1}^{m} \left(\sum_{k=1}^{n} \widehat{z}_{kl} \right) \cdot \left\{ G\big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big) - \big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big) \frac{\sum_{k=1}^{n} \widehat{z}_{kl}\, \mathbf{x}[k]^T}{\sum_{k=1}^{n} \widehat{z}_{kl}} \right\}$$

$$= \arg\min_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \sum_{l=1}^{m} \widehat{\pi}_l \left\{ G\big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big) - \big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big) \frac{\sum_{k=1}^{n} \widehat{z}_{kl}\, \mathbf{x}[k]^T}{\sum_{k=1}^{n} \widehat{z}_{kl}} \right\},$$

since $(1/n)\sum_{k=1}^{n} \widehat{z}_{kl} = \widehat{\pi}_l$. As in [15, 82], the following notation is introduced:

$$\widetilde{\mathbf{x}}[l] = \frac{\sum_{k=1}^{n} \widehat{z}_{kl}\, \mathbf{x}[k]}{\sum_{k=1}^{n} \widehat{z}_{kl}},$$

the lth mixture component center, for $l = 1, \ldots, m$. Since the vector $\mathbf{x}$ belongs to $\mathbb{R}^d$, similarly, the vector $\widetilde{\mathbf{x}}$ belongs to $\mathbb{R}^d$. The loss function is then defined as

$$L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}) = \sum_{l=1}^{m} \widehat{\pi}_l \left\{ G\big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big) - \big(\underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}\big)\widetilde{\mathbf{x}}[l]^T \right\}. \tag{5.13}$$

Note that the coefficients $\widehat{\pi}_l$, $l = 1, \ldots, m$, are not present in the algorithm proposed in [15, 82]. It is easily noticed how similar the loss function (4.12) in Section 4 and the loss function described in equation (5.13) are: the summation over k becomes a summation over l, the data vector $\mathbf{x}[k]$ becomes the mixture component center $\widetilde{\mathbf{x}}[l]$ and a coefficient weighting the importance of the l^{th} mixture component compared to other components is introduced. Hence, following the derivations in Section 4, the classical Newton-Raphson method is used for the iterative minimization of the loss function (5.13) and the resulting update equations are easily deduced from the update equations (4.14), (4.17) and (4.18) from Section 4.

First, for $l = 1, \ldots, m$, at iteration $(t+1)$,

$$\begin{aligned} \underline{\mathbf{a}}^{(t+1)}[l]^T = \underline{\mathbf{a}}^{(t)}[l]^T - \alpha_{\underline{\mathbf{a}}}^{(t+1)} \Big(\mathbf{V}^{(t)} G''\big(\underline{\mathbf{a}}^{(t)}[l]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)\mathbf{V}^{(t),T} \Big)^{-1} \\ \cdot \Big(\mathbf{V}^{(t)} \big(G'(\underline{\mathbf{a}}^{(t)}[l]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}) - \widetilde{\mathbf{x}}[l]^T \big) \Big). \end{aligned} \tag{5.14}$$

Then, for $j = 1, \ldots, q$:

$$\begin{aligned} \mathbf{v}_j^{(t+1),T} = \mathbf{v}_j^{(t),T} - \alpha_{\mathbf{v}}^{(t+1)} \left(\sum_{l=1}^{m} \widehat{\pi}_l\, \underline{a}_j^{(t+1)}[l]^2 G''\big(\underline{\mathbf{a}}^{(t+1)}[l]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \right)^{-1} \\ \cdot \left(\sum_{l=1}^{m} \widehat{\pi}_l\, \underline{a}_j^{(t+1)}[l] \big\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[l]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \widetilde{\mathbf{x}}[l]^T \big\} \right). \end{aligned} \tag{5.15}$$

And finally,

$$\begin{aligned}\mathbf{b}^{(t+1),T} = \mathbf{b}^{(t),T} - \alpha_{\mathbf{b}}^{(t+1)} \left(\sum_{l=1}^{m} \widehat{\pi}_l \, G''\big(\underline{\mathbf{a}}^{(t+1)}[l]\mathbf{V}^{(t+1)} + \mathbf{b}\big) \right)^{-1} \\ \cdot \left(\sum_{l=1}^{m} \widehat{\pi}_l \left\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[l]\mathbf{V}^{(t+1)} + \mathbf{b}\big) - \widetilde{\mathbf{x}}[l]^T \right\} \right).\end{aligned} \tag{5.16}$$

Table 5.1 summarizes the Semi-Parametric exponential family Principal Component Analysis algorithm.

5.2.1 The mixed data-type case

As previously in Section 4.3 for exponential PCA, the SP-PCA approach is now modified to be able to address mixed data-type cases.

For simplicity of presentation, we consider that the f first attributes are distributed according to the exponential family distribution $p^{(\mathbf{1})}$ and the $(d-f)$ last attributes are distributed according to the exponential family distribution $p^{(\mathbf{2})}$. The following notation is used:

$$\mathbf{x}[k] = \big[x_1[k], \ldots, x_f[k], x_{f+1}[k], \ldots, x_d[k]\big] = \big[\mathbf{x}^{(\mathbf{1})}[k] \big| \mathbf{x}^{(\mathbf{2})}[k]\big],$$

for $k = 1, \ldots, n$, and

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}[1] \\ \mathbf{x}[2] \\ \vdots \\ \mathbf{x}[n] \end{pmatrix} = \left(\begin{array}{ccc|ccc} x_1[1] & \ldots & x_f[1] & x_{f+1}[1] & \ldots & x_d[1] \\ x_1[2] & \ldots & x_f[2] & x_{f+1}[2] & \ldots & x_d[2] \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_1[n] & \ldots & x_f[n] & x_{f+1}[n] & \ldots & x_d[n] \end{array} \right) = \Big(\mathbf{X}^{(\mathbf{1})} \Big| \mathbf{X}^{(\mathbf{2})} \Big).$$

The complete log-likelihood function is expressed as follows:

$$\begin{aligned} L^{(c)}\big(\mathcal{Q}, \{\mathbf{z}_k\}_{k=1}^n\big) &= \log \prod_{k=1}^{n} \prod_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)^{z_{kl}} \pi_l^{z_{kl}} \\ &= \sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big) + \sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} \log \pi_l, \end{aligned}$$

Table 5.1 The Semi-Parametric exponential family Principal Component Analysis algorithm.

Algorithm: Semi-Parametric exponential family PCA [15, 82]

Input: a set of observations $\{\mathbf{x}[k]\}_{k=1}^{n} \subseteq \mathbb{R}^d$, an exponential family distribution $p(\cdot)$ defined by its cumulant generating function $G(\cdot)$, a number of atoms m, $q \ll d$ the dimension of the latent variable lower dimensional subspace.

Output: the NPML estimator that maximizes the complete log-likelihood function $L^{(c)}\big(\mathcal{Q}, \{\mathbf{z}_k\}_{k=1}^{n}\big)$: $\widehat{\mathcal{Q}} = \{\widehat{\underline{\boldsymbol{\theta}}}[l], \widehat{\pi}_l\}_{l=1}^{m}$ with $\widehat{\underline{\boldsymbol{\theta}}}[l] = \widehat{\underline{\mathbf{a}}}[l]\widehat{\mathbf{V}} + \widehat{\mathbf{b}}$ for all l, $\{\widehat{\underline{\mathbf{a}}}[l]\}_{l=1}^{m} \in \mathbb{R}^q$, $\widehat{\mathbf{V}} \in \mathbb{R}^{q\times d}$ and $\widehat{\mathbf{b}} \in \mathbb{R}^d$.

Method:

Initialize $\mathbf{V}$, $\mathbf{b}$ and $\{\underline{\mathbf{a}}[l], \pi_l\}_{l=1}^{m}$ with $\pi_l \geq 0$ for all l and $\sum_{l=1}^{m} \pi_l = 1$; $\underline{\boldsymbol{\theta}}[l] = \underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b} \in \boldsymbol{\Theta}$ for all l;

repeat

{The Expectation Step}

for $k = 1$ to n **do**

for $l = 1$ to m **do**

$\widehat{z}_{kl} \longleftarrow p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l / \sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r$

end for

end for

{The Maximization Step}

for $l = 1$ to m **do**

$\widehat{\pi}_l \longleftarrow (1/n)\sum_{k=1}^{n} \widehat{z}_{kl}$

end for

{The Newton-Raphson iterative algorithm}

for $l = 1$ to m **do**

$\underline{\mathbf{a}}[l] \longleftarrow$ update equation (5.14)

end for

for $j = 1$ to q **do**

$\mathbf{v}_j \longleftarrow$ update equation (5.15)

end for

$\mathbf{b} \longleftarrow$ update equation (5.16)

until *convergence*;

return $\widehat{\mathcal{Q}} = \{\widehat{\underline{\boldsymbol{\theta}}}[l] = \widehat{\underline{\mathbf{a}}}[l]\widehat{\mathbf{V}} + \widehat{\mathbf{b}}, \widehat{\pi}_l\}_{l=1}^{m}$.

where $\underline{\boldsymbol{\theta}}[l] = \underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}$. Then, using the *latent variable assumption*,

$$
\begin{aligned}
&p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big) \\
&\quad = p_1(x_1[k]|\underline{\theta}_1[l]) \cdots p_f(x_f[k]|\underline{\theta}_f[l]) p_{f+1}(x_{f+1}[k]|\underline{\theta}_{f+1}[l]) \cdots p_d(x_d[k]|\underline{\theta}_d[l]) \\
&\quad = p^{(\mathbf{1})}(x_1[k]|\underline{\theta}_1[l]) \cdots p^{(\mathbf{1})}(x_f[k]|\underline{\theta}_f[l]) p^{(\mathbf{2})}(x_{f+1}[k]|\underline{\theta}_{f+1}[l]) \cdots p^{(\mathbf{2})}(x_d[k]|\underline{\theta}_d[l]) \\
&\quad = p^{(\mathbf{1})}\big(\mathbf{x}^{(\mathbf{1})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{1})}[l]\big) p^{(\mathbf{2})}\big(\mathbf{x}^{(\mathbf{2})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{2})}[l]\big),
\end{aligned}
$$

where

$$
\underline{\boldsymbol{\theta}}[l] = \big[\underline{\theta}_1[l], \ldots, \underline{\theta}_f[l], \underline{\theta}_{f+1}[l], \ldots, \underline{\theta}_d[l]\big] = \big[\underline{\boldsymbol{\theta}}^{(\mathbf{1})}[l] \big| \underline{\boldsymbol{\theta}}^{(\mathbf{2})}[l]\big],
$$

for $l = 1, \ldots, m$, and

$$
\underline{\boldsymbol{\Theta}} = \begin{pmatrix} \underline{\boldsymbol{\theta}}[1] \\ \underline{\boldsymbol{\theta}}[2] \\ \vdots \\ \underline{\boldsymbol{\theta}}[m] \end{pmatrix} = \left(\begin{array}{ccc|ccc} \underline{\theta}_1[1] & \ldots & \underline{\theta}_f[1] & \underline{\theta}_{f+1}[1] & \ldots & \underline{\theta}_d[1] \\ \underline{\theta}_1[2] & \ldots & \underline{\theta}_f[2] & \underline{\theta}_{f+1}[2] & \ldots & \underline{\theta}_d[2] \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \underline{\theta}_1[m] & \ldots & \underline{\theta}_f[m] & \underline{\theta}_{f+1}[m] & \ldots & \underline{\theta}_d[m] \end{array}\right) = \left(\underline{\boldsymbol{\Theta}}^{(\mathbf{1})} \,\middle|\, \underline{\boldsymbol{\Theta}}^{(\mathbf{2})}\right).
$$

Then, $\underline{\boldsymbol{\Theta}} = \underline{\mathbf{A}}\mathbf{V} + \mathbf{B}$ results in the following decompositions:

$$
\mathbf{V} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_q \end{pmatrix} = \left(\begin{array}{ccc|ccc} v_{11} & \ldots & v_{1f} & v_{1(f+1)} & \ldots & v_{1d} \\ v_{21} & \ldots & v_{2f} & v_{2(f+1)} & \ldots & v_{2d} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ v_{q1} & \ldots & v_{qf} & v_{q(f+1)} & \ldots & v_{qd} \end{array}\right) = \left(\mathbf{V}^{(\mathbf{1})} \,\middle|\, \mathbf{V}^{(\mathbf{2})}\right),
$$

and

$$
\mathbf{B} = \begin{pmatrix} \mathbf{b} \\ \mathbf{b} \\ \vdots \\ \mathbf{b} \end{pmatrix} = \left(\begin{array}{ccc|ccc} b_1 & \ldots & b_f & b_{(f+1)} & \ldots & b_d \\ b_1 & \ldots & b_f & b_{(f+1)} & \ldots & b_d \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ b_1 & \ldots & b_f & b_{(f+1)} & \ldots & b_d \end{array}\right) = \left(\mathbf{B}^{(\mathbf{1})} \,\middle|\, \mathbf{B}^{(\mathbf{2})}\right),
$$

where $\mathbf{B}^{(\mathbf{1})} = [\mathbf{b}^{(\mathbf{1})}, \ldots, \mathbf{b}^{(\mathbf{1})}]^T$ and $\mathbf{b}^{(\mathbf{1})} = [b_1, \ldots, b_f]$, $\mathbf{B}^{(\mathbf{2})} = [\mathbf{b}^{(\mathbf{2})}, \ldots, \mathbf{b}^{(\mathbf{2})}]^T$ and $\mathbf{b}^{(\mathbf{2})} = [b_{f+1}, \ldots, b_d]$. Hence,

$$
\underline{\boldsymbol{\Theta}} = \begin{pmatrix} \underline{\boldsymbol{\theta}}[1] \\ \underline{\boldsymbol{\theta}}[2] \\ \vdots \\ \underline{\boldsymbol{\theta}}[m] \end{pmatrix} = \underline{\mathbf{A}}\mathbf{V} + \mathbf{B} = \left(\underline{\mathbf{A}}\mathbf{V}^{(\mathbf{1})} + \mathbf{B}^{(\mathbf{1})} \,\middle|\, \underline{\mathbf{A}}\mathbf{V}^{(\mathbf{2})} + \mathbf{B}^{(\mathbf{2})}\right).
$$

Note that there is no such split for $\underline{\mathbf{A}}$.

The complete log-likelihood function becomes:

$$\begin{aligned} L^{(c)}\big(\mathcal{Q}, \{\mathbf{z}_k\}_{k=1}^n\big) &= \sum_{k=1}^{n}\sum_{l=1}^{m} z_{kl} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big) + \sum_{k=1}^{n}\sum_{l=1}^{m} z_{kl} \log \pi_l \\ &= \sum_{k=1}^{n}\sum_{l=1}^{m} z_{kl} \log \left\{ p^{(\mathbf{1})}\big(\mathbf{x}^{(\mathbf{1})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{1})}[l]\big) \cdot p^{(\mathbf{2})}\big(\mathbf{x}^{(\mathbf{2})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{2})}[l]\big) \right\} \\ &\quad + \sum_{k=1}^{n}\sum_{l=1}^{m} z_{kl} \log \pi_l. \end{aligned} \tag{5.17}$$

The E-step remains unchanged:

$$\begin{aligned} \widehat{z}_{kl} = \mathrm{E}\left\{z_{kl}|\mathbf{x}[k], \pi_1, \ldots, \pi_m\right\} &= \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} \\ &= \frac{p^{(\mathbf{1})}\big(\mathbf{x}^{(\mathbf{1})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{1})}[l]\big) \cdot p^{(\mathbf{2})}\big(\mathbf{x}^{(\mathbf{2})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{2})}[l]\big)\big)\pi_l}{\sum_{r=1}^{m} p^{(\mathbf{1})}\big(\mathbf{x}^{(\mathbf{1})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{1})}[r]\big) \cdot p^{(\mathbf{2})}\big(\mathbf{x}^{(\mathbf{2})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{2})}[r]\big)\pi_r}, \end{aligned}$$

for $k = 1, \ldots, n$ and $l = 1, \ldots, m$.

The M-step first yields the estimates for the point-mass probabilities:

$$\begin{aligned} \widehat{\pi}_l = \frac{\sum_{k=1}^{n} \widehat{z}_{kl}}{n} &= \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l}{n\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} \\ &= \sum_{k=1}^{n} \frac{p^{(\mathbf{1})}\big(\mathbf{x}^{(\mathbf{1})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{1})}[l]\big) \cdot p^{(\mathbf{2})}\big(\mathbf{x}^{(\mathbf{2})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{2})}[l]\big)\pi_l}{n\sum_{r=1}^{m} p^{(\mathbf{1})}\big(\mathbf{x}^{(\mathbf{1})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{1})}[r]\big) \cdot p^{(\mathbf{2})}\big(\mathbf{x}^{(\mathbf{2})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{2})}[r]\big)\pi_r}. \end{aligned}$$

However, the second part of the M-step, i.e., the estimation of the parameters $\mathbf{V}$, $\mathbf{b}$, and the point-mass support points $\underline{\mathbf{A}} = \big[\underline{\mathbf{a}}[1]^T, \ldots, \underline{\mathbf{a}}[m]^T\big]^T \in \mathbb{R}^{m,q}$, has to be modified. Maximizing the complete log-likelihood function (5.17) with respect to these parameters is equivalent to:

$$\begin{aligned} &\arg\max_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} L^{(c)}\big(\mathcal{Q}, \{\widehat{\mathbf{z}}_k\}_{k=1}^n\big) \\ &= \arg\max_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl} \log p^{(\mathbf{1})}\big(\mathbf{x}^{(\mathbf{1})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{1})}[l]\big) \cdot p^{(\mathbf{2})}\big(\mathbf{x}^{(\mathbf{2})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{2})}[l]\big) \\ &\quad + \underline{\sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl} \log \pi_l} \\ &\Longleftrightarrow \arg\max_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl} \log p^{(\mathbf{1})}\big(\mathbf{x}^{(\mathbf{1})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{1})}[l]\big) \cdot p^{(\mathbf{2})}\big(\mathbf{x}^{(\mathbf{2})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{2})}[l]\big) \end{aligned}$$

since the underlined term does not depend on either $\underline{\mathbf{A}}$, $\mathbf{V}$, or $\mathbf{b}$

$$\begin{aligned}\Longleftrightarrow \arg\max_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \Bigg\{ &\sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl}\Big\{G^{(\mathbf{1})}\big(\underline{\mathbf{a}}[l]\mathbf{V}^{(\mathbf{1})}+\mathbf{b}^{(\mathbf{1})}\big)-\big(\underline{\mathbf{a}}[l]\mathbf{V}^{(\mathbf{1})}+\mathbf{b}^{(\mathbf{1})}\big)\mathbf{x}^{(\mathbf{1})}[k]^T\Big\}\\ &+\sum_{k=1}^{n}\sum_{l=1}^{m} \widehat{z}_{kl}\Big\{G^{(\mathbf{2})}\big(\underline{\mathbf{a}}[l]\mathbf{V}^{(\mathbf{2})}+\mathbf{b}^{(\mathbf{2})}\big)-\big(\underline{\mathbf{a}}[l]\mathbf{V}^{(\mathbf{2})}+\mathbf{b}^{(\mathbf{2})}\big)\mathbf{x}^{(\mathbf{2})}[k]^T\Big\}\Bigg\}\end{aligned}$$

where $G^{(\mathbf{1})}(\cdot)$, $G^{(\mathbf{2})}(\cdot)$ respectively, is the cumulant generating function associated with the exponential family distribution $p^{(\mathbf{1})}(\cdot)$, $p^{(\mathbf{2})}(\cdot)$ respectively,

$$\begin{aligned}\Longleftrightarrow \arg\min_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} &\sum_{l=1}^{m}\widehat{\pi}_l\left\{G^{(\mathbf{1})}\big(\underline{\mathbf{a}}[l]\mathbf{V}^{(\mathbf{1})}+\mathbf{b}^{(\mathbf{1})}\big)-\big(\underline{\mathbf{a}}[l]\mathbf{V}^{(\mathbf{1})}+\mathbf{b}^{(\mathbf{1})}\big)\frac{\sum_{k=1}^{n}\widehat{z}_{kl}\,\mathbf{x}^{(\mathbf{1})}[k]^T}{\sum_{k=1}^{n}\widehat{z}_{kl}}\right\}\\ &+\sum_{l=1}^{m}\widehat{\pi}_l\left\{G^{(\mathbf{2})}\big(\underline{\mathbf{a}}[l]\mathbf{V}^{(\mathbf{2})}+\mathbf{b}^{(\mathbf{2})}\big)-\big(\underline{\mathbf{a}}[l]\mathbf{V}^{(\mathbf{2})}+\mathbf{b}^{(\mathbf{2})}\big)\frac{\sum_{k=1}^{n}\widehat{z}_{kl}\,\mathbf{x}^{(\mathbf{2})}[k]^T}{\sum_{k=1}^{n}\widehat{z}_{kl}}\right\},\end{aligned}$$

since $1/n\sum_{k=1}^{n}\widehat{z}_{kl}=\widehat{\pi}_l$. As previously, the following notation is introduced:

$$\widetilde{\mathbf{x}}[l]=\frac{\sum_{k=1}^{n}\widehat{z}_{kl}\,\mathbf{x}[k]}{\sum_{k=1}^{n}\widehat{z}_{kl}},$$

for $l=1,\ldots,m$. The loss function is then defined as

$$\begin{aligned}L(\underline{\mathbf{A}},\mathbf{V},\mathbf{b})=&\sum_{l=1}^{m}\widehat{\pi}_l\left\{G^{(\mathbf{1})}\big(\underline{\mathbf{a}}[l]\mathbf{V}^{(\mathbf{1})}+\mathbf{b}^{(\mathbf{1})}\big)-\big(\underline{\mathbf{a}}[l]\mathbf{V}^{(\mathbf{1})}+\mathbf{b}^{(\mathbf{1})}\big)\widetilde{\mathbf{x}}^{(\mathbf{1})}[l]^T\right\}\\ &+\sum_{l=1}^{m}\widehat{\pi}_l\left\{G^{(\mathbf{2})}\big(\underline{\mathbf{a}}[l]\mathbf{V}^{(\mathbf{2})}+\mathbf{b}^{(\mathbf{2})}\big)-\big(\underline{\mathbf{a}}[l]\mathbf{V}^{(\mathbf{2})}+\mathbf{b}^{(\mathbf{2})}\big)\widetilde{\mathbf{x}}^{(\mathbf{2})}[l]^T\right\}.\end{aligned}\tag{5.18}$$

Following the derivations in Section 4.3, the classical Newton-Raphson method is used for the iterative minimization of the loss function (5.18) and the resulting update equations are as follows.

First, for $l=1,\ldots,m$, at iteration $(t+1)$,

$$\begin{aligned}\underline{\mathbf{a}}^{(t+1)}[l]^T=\underline{\mathbf{a}}^{(t)}[l]^T-\alpha_{\underline{\mathbf{a}}}^{(t+1)}\Big\{&\mathbf{V}^{(\mathbf{1})(t)}G^{(\mathbf{1})''}\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(\mathbf{1})(t)}+\mathbf{b}^{(\mathbf{1})(t)}\big)\mathbf{V}^{(\mathbf{1})(t),T}\\ &+\mathbf{V}^{(\mathbf{2})(t)}G^{(\mathbf{2})''}\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(\mathbf{2})(t)}+\mathbf{b}^{(\mathbf{2})(t)}\big)\mathbf{V}^{(\mathbf{2})(t),T}\Big\}^{-1}\\ &\cdot\Big\{\mathbf{V}^{(\mathbf{1})(t)}\big(G^{(\mathbf{1})'}(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(\mathbf{1})(t)}+\mathbf{b}^{(\mathbf{1})(t)})-\widetilde{\mathbf{x}}^{(\mathbf{1})}[l]^T\big)\\ &+\mathbf{V}^{(\mathbf{2})(t)}\big(G^{(\mathbf{2})'}(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(\mathbf{2})(t)}+\mathbf{b}^{(\mathbf{2})(t)})-\widetilde{\mathbf{x}}^{(\mathbf{2})}[l]^T\big)\Big\}.\end{aligned}$$

For the second step, the two sets of row vectors $\left\{\mathbf{v}_j^{(\mathbf{1})}\right\}_{j=1}^{q}$ and $\left\{\mathbf{v}_j^{(\mathbf{2})}\right\}_{j=1}^{q}$ are updated separately. The update equations can then be used for $\left\{\mathbf{v}_j^{(\mathbf{1})}\right\}_{j=1}^{q}$ and $\left\{\mathbf{v}_j^{(\mathbf{2})}\right\}_{j=1}^{q}$ by changing $\mathbf{v}_j$ to $\mathbf{v}_j^{(\mathbf{1})}$, respectively to $\mathbf{v}_j^{(\mathbf{2})}$, $\mathbf{b}$ to $\mathbf{b}^{(\mathbf{1})}$, respectively to $\mathbf{b}^{(\mathbf{2})}$, $G(\cdot), G'(\cdot)$, and $G''(\cdot)$ to $G^{(\mathbf{1})}(\cdot), G^{(\mathbf{1})'}(\cdot)$, and $G^{(\mathbf{1})''}(\cdot)$, respectively to $G^{(\mathbf{2})}(\cdot), G^{(\mathbf{2})'}(\cdot)$, and $G^{(\mathbf{2})''}(\cdot)$. For $j = 1, \ldots, q$:

$$\mathbf{v}_j^{(t+1),T} = \mathbf{v}_j^{(t),T} - \alpha_{\mathbf{v}}^{(t+1)} \left(\sum_{l=1}^{m} \widehat{\pi}_l \, \underline{a}_j^{(t+1)}[l]^2 G''\big(\underline{\mathbf{a}}^{(t+1)}[l]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \right)^{-1} \cdot \left(\sum_{l=1}^{m} \widehat{\pi}_l \, \underline{a}_j^{(t+1)}[l] \big\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[l]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \widetilde{\mathbf{x}}[l]^T \big\} \right).$$

And finally for the last step, the update equations can then be used for $\mathbf{b}^{(\mathbf{1})}$ and $\mathbf{b}^{(\mathbf{2})}$ by changing $\mathbf{b}$ to $\mathbf{b}^{(\mathbf{1})}$, respectively to $\mathbf{b}^{(\mathbf{2})}$, $\mathbf{V}$ to $\mathbf{V}^{(\mathbf{1})}$, respectively to $\mathbf{V}^{(\mathbf{2})}$, $G(\cdot), G'(\cdot)$, and $G''(\cdot)$ to $G^{(\mathbf{1})}(\cdot), G^{(\mathbf{1})'}(\cdot)$, and $G^{(\mathbf{1})''}(\cdot)$, respectively to $G^{(\mathbf{2})}(\cdot), G^{(\mathbf{2})'}(\cdot)$, and $G^{(\mathbf{2})''}(\cdot)$.

$$\mathbf{b}^{(t+1),T} = \mathbf{b}^{(t),T} - \alpha_{\mathbf{b}}^{(t+1)} \left(\sum_{l=1}^{m} \widehat{\pi}_l \, G''\big(\underline{\mathbf{a}}^{(t+1)}[l]\mathbf{V}^{(t+1)} + \mathbf{b}\big) \right)^{-1} \cdot \left(\sum_{l=1}^{m} \widehat{\pi}_l \left\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[l]\mathbf{V}^{(t+1)} + \mathbf{b}\big) - \widetilde{\mathbf{x}}[l]^T \right\} \right).$$

5.3 Exponential PCA approach

The work proposed in Section 4 is based on the generalization of Principal Component Analysis to the exponential family technique presented in [10], often referred to as exponential family Principal Component Analysis or exponential PCA.

As stated earlier, instead of the fastidious estimation of the point-mass probabilities, exponential PCA considers the special classical case for which the number of parameter points equals the number of data samples, i.e., $m = n$. The point-mass probabilities do not need to be estimated and the EM algorithm is unnecessary. Then, each vector $\mathbf{x}$ corresponds to a single vector $\underline{\mathbf{a}}$, i.e., a single vector $\underline{\boldsymbol{\theta}}$, and they all share a common index $k = 1, \ldots, n$.

Following the notations used previously in the SP-PCA presentation, the set of parameters to be estimated is denoted by $\mathcal{Q} = \{\underline{\boldsymbol{\theta}}[k]\}_{k=1}^{n}$, with linear predictors $\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}$ for $k = 1, \ldots, n$. The Maximum Likelihood estimator is $\widehat{\mathcal{Q}} = \{\widehat{\underline{\boldsymbol{\theta}}}[k] = \widehat{\underline{\mathbf{a}}}[k]\widehat{\mathbf{V}} + \widehat{\mathbf{b}}\}_{k=1}^{m}$. The classical approach can also be seen as an extreme case of the Bayesian approach for which the probability density function $\pi(\boldsymbol{\theta})$ is a delta function (one per data point). Hence, the log-likelihood function is given by:

$$L(\mathcal{Q}) = \sum_{k=1}^{n} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[k]\big)$$

and,

$$\begin{aligned}
\arg\max_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} L(\mathcal{Q}) &= \arg\max_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \sum_{k=1}^{n} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[k]\big) \\
&= \arg\max_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \sum_{k=1}^{n} \Big\{ G\big(\underline{\boldsymbol{\theta}}[k]\big) - \underline{\boldsymbol{\theta}}[k]\mathbf{x}[k]^T \Big\} \\
&= \arg\max_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} \sum_{k=1}^{n} \Big\{ G\big(\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}\big) - \big(\underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b}\big)\mathbf{x}[k]^T \Big\} \\
&= \arg\max_{\underline{\mathbf{A}},\mathbf{V},\mathbf{b}} L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b}),
\end{aligned}$$

where $L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b})$ is defined in equation (3.13) in Section 4. Then, following the derivations in Section 4, the classical Newton-Raphson method is used for the iterative minimization of the loss function $L(\underline{\mathbf{A}}, \mathbf{V}, \mathbf{b})$ and the resulting update equations are copied below from the update equations (4.14), (4.17) and (4.18).

First, for $l = 1, \ldots, m$, at iteration $(t+1)$,

$$\begin{aligned}
\underline{\mathbf{a}}^{(t+1)}[k]^T = \underline{\mathbf{a}}^{(t)}[k]^T &- \alpha_{\underline{\mathbf{a}}}^{(t+1)} \Big(\mathbf{V}^{(t)} G''\big(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big)\mathbf{V}^{(t),T} \Big)^{-1} \\
&\cdot \Big(\mathbf{V}^{(t)} \big(G'(\underline{\mathbf{a}}^{(t)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}) - \mathbf{x}[k]^T \big) \Big).
\end{aligned} \tag{5.19}$$

Then, for $j = 1, \ldots, q$:

$$\begin{aligned}
\mathbf{v}_j^{(t+1),T} = \mathbf{v}_j^{(t),T} &- \alpha_{\mathbf{v}}^{(t+1)} \left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k]^2 G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) \right)^{-1} \\
&\cdot \left(\sum_{k=1}^{n} \underline{a}_j^{(t+1)}[k] \big\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t)} + \mathbf{b}^{(t)}\big) - \mathbf{x}[k]^T \big\} \right).
\end{aligned} \tag{5.20}$$

And, finally,

$$\mathbf{b}^{(t+1),T} = \mathbf{b}^{(t),T} - \alpha_{\mathbf{b}}^{(t+1)} \left(\sum_{k=1}^{n} G''\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) \right)^{-1} \cdot \left(\sum_{k=1}^{n} \big\{ G'\big(\underline{\mathbf{a}}^{(t+1)}[k]\mathbf{V}^{(t+1)} + \mathbf{b}\big) - \mathbf{x}[k]^T \big\} \right). \tag{5.21}$$

It is easily noticed that equations (5.19), (5.20) and (5.21) are almost identical to equations (5.14), (5.15) and (5.16), the difference being that the exponential PCA update equations use $\mathbf{x}$ instead of $\widetilde{\mathbf{x}}$, i.e., data points instead of mixture component centers, and do not include mixture component proportion factors. Each data point $\mathbf{x}$ is its own mixture component center.

Table 5.2 summarizes the exponential family Principal Component Analysis algorithm.

The mixed data-type case was already presented in Section 4.3.

5.4 Bregman soft clustering approach

The Bregman soft clustering approach presented in [14, 16] utilizes an alternative interpretation of the EM algorithm for learning models involving mixtures of exponential family distributions. It is a simple soft clustering algorithm for all Bregman distances, i.e., for all exponential family distributions. It is based on the fact that there exists a bijection between Bregman distances and exponential family distributions. Indeed, the existence of a unique Bregman distance corresponding to every regular exponential family had been previously observed [57, 87], but was formally proven by [16] (further discussed in Appendix A).

Given a data set of observations $\big\{\mathbf{x}[k]\big\}_{k=1}^{n}$ where $\mathbf{x}[k] = \big[x_1[k], \ldots, x_d[k]\big]$, Bregman soft clustering aims at modeling the statistical structure of the data as a mixture of m densities of the same exponential family. The clusters correspond to the components of the mixture model and the soft membership of a data point in each cluster is proportional to the probability of the data point being generated by the corresponding density function. The Bregman soft clustering problem is based

Table 5.2 The exponential family Principal Component Analysis algorithm.

Algorithm: Exponential PCA [10]

Input: a set of observations $\{\mathbf{x}[k]\}_{k=1}^{n} \subseteq \mathbb{R}^d$, an exponential family distribution $p(\cdot)$ defined by its cumulant generating function $G(\cdot)$, a number of atoms n, $q \ll d$ the dimension of the latent variable lower dimensional subspace.

Output: the ML estimator that maximizes the log-likelihood function $L(\mathcal{Q}) = \log \prod_{k=1}^{n} p(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[k])$: $\widehat{\mathcal{Q}} = \{\widehat{\underline{\boldsymbol{\theta}}}[k]\}_{k=1}^{n}$ with $\widehat{\underline{\boldsymbol{\theta}}}[k] = \widehat{\underline{\mathbf{a}}}[k]\widehat{\mathbf{V}} + \widehat{\mathbf{b}}$ for all k, $\{\widehat{\underline{\mathbf{a}}}[k]\}_{k=1}^{n} \in \mathbb{R}^q$, $\widehat{\mathbf{V}} \in \mathbb{R}^{q \times d}$ and $\widehat{\mathbf{b}} \in \mathbb{R}^d$.

Method:

Initialize $\mathbf{V}$, $\mathbf{b}$ and $\{\underline{\mathbf{a}}[k]\}_{k=1}^{n}$; $\underline{\boldsymbol{\theta}}[k] = \underline{\mathbf{a}}[k]\mathbf{V} + \mathbf{b} \in \boldsymbol{\Theta}$ for all k;

repeat

{The Newton-Raphson iterative algorithm}

for $k = 1$ to n **do**

$\underline{\mathbf{a}}[k] \longleftarrow$ update equation (5.19)

end for

for $j = 1$ to q **do**

$\mathbf{v}_j \longleftarrow$ update equation (5.20)

end for

$\mathbf{b} \longleftarrow$ update equation (5.21)

until *convergence*;

return $\widehat{\mathcal{Q}} = \{\widehat{\underline{\boldsymbol{\theta}}}[k] = \widehat{\underline{\mathbf{a}}}[k]\widehat{\mathbf{V}} + \widehat{\mathbf{b}}\}_{k=1}^{n}$.

on a Maximum Likelihood (ML) estimation of the cluster parameters $\{\underline{\boldsymbol{\theta}}[l], \pi_l\}_{l=1}^{m}$ satisfying the following mixture structure:

$$p(\mathbf{x}) = \sum_{l=1}^{m} p\big(\mathbf{x}|\underline{\boldsymbol{\theta}}[l]\big)\pi_l, \tag{5.22}$$

where $p(\mathbf{x}|\cdot)$ is an exponential family distribution. The data likelihood function takes the following form:

$$p(\mathbf{X}) = \prod_{k=1}^{n}\sum_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l. \tag{5.23}$$

The bijection between Bregman distances and exponential family distributions states that for any exponential family distribution $p(\mathbf{x}|\cdot)$,

$$p(\mathbf{x}|\underline{\boldsymbol{\theta}}) = \exp\Big\{ - B_F\big(\mathbf{x}\|g(\underline{\boldsymbol{\theta}})\big)\Big\} \cdot \exp\big\{F(\mathbf{x})\big\},$$

where $F(\cdot)$ is the Fenchel conjugate of the cumulant generative function associated with $p(\mathbf{x}|\cdot)$, and $B_F(\cdot\|\cdot)$ is the Bregman distance associated with $p(\mathbf{x}|\cdot)$. Hence, equations (5.22) and (5.23) become:

$$p(\mathbf{x}) = \sum_{l=1}^{m} \exp\Big\{ - B_F\big(\mathbf{x}\|g(\underline{\boldsymbol{\theta}}[l])\big)\Big\} \cdot \exp\big\{F(\mathbf{x})\big\}\pi_l, \tag{5.24}$$

$$p(\mathbf{X}) = \prod_{k=1}^{n}\sum_{l=1}^{m} \exp\Big\{ - B_F\big(\mathbf{x}[k]\|g(\underline{\boldsymbol{\theta}}[l])\big)\Big\} \cdot \exp\big\{F(\mathbf{x}[k])\big\}\pi_l. \tag{5.25}$$

The Expectation-Maximization (EM) algorithm is used for the parameters estimation. The authors in [14, 16] show that the maximization step reduces to:

$$\underline{\boldsymbol{\theta}}[l] = \frac{\sum_{k=1}^{n} p\big(l|\mathbf{x}[k]\big)\mathbf{x}[k]}{\sum_{k=1}^{n} p\big(l|\mathbf{x}[k]\big)},$$

where $p\big(l|\mathbf{x}[k]\big)$ is the posterior probability of cluster l containing the data point $\mathbf{x}[k]$, given the data point $\mathbf{x}[k]$. The update equation for the posterior probabilities are given by

$$p\big(l|\mathbf{x}\big) = \frac{\exp\Big\{ - B_F\big(\mathbf{x}\|g(\underline{\boldsymbol{\theta}}[l])\big)\Big\}\pi_l}{\sum_{r=1}^{m} \exp\Big\{ - B_F\big(\mathbf{x}\|g(\underline{\boldsymbol{\theta}}[r])\big)\Big\}\pi_r}.$$

The Bregman soft clustering algorithm uses an EM approach to a mixture of m exponential family distribution problem. This exactly corresponds to the NPML for exponential family distributions discussed in Appendix C with the notation $z_{kl} = p\big(l|\mathbf{x}[k]\big)$.

We now present this technique without referring to the Bregman distance but by using its corresponding exponential family probability distribution for the sake of comparison with SP-PCA and exponential PCA. The data likelihood function in (5.23) is similar to the data likelihood function in (5.6) without the linear constraint $\underline{\boldsymbol{\theta}}[l] = \underline{\mathbf{a}}[l]\mathbf{V} + \mathbf{b}$ for $l = 1, \ldots, m$. Hence, the Bregman soft clustering problem is similar to the SP-PCA problem without the lower dimensional subspace constraint and a simple EM algorithm is used to estimate the cluster parameters. The E-step and the first part of the M-step yield the same results as for SP-PCA. In the second part of the M-step, the component parameters $\underline{\boldsymbol{\theta}}[l], l = 1, \ldots, m$, are estimated in the following way:

$$\widehat{\underline{\boldsymbol{\theta}}}[l] = \arg\max_{\underline{\boldsymbol{\theta}}[l]} \sum_{k=1}^{n} \sum_{r=1}^{m} \widehat{z}_{kr} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big),$$

with $\log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big) = \underline{\boldsymbol{\theta}}[r]\mathbf{x}[k]^T - G\big(\underline{\boldsymbol{\theta}}[r]\big)$. Using the convexity properties of $G(\cdot)$, it is easily shown that:

$$G'(\widehat{\underline{\boldsymbol{\theta}}}[l]) = \Big(\sum_{k=1}^{n} \widehat{z}_{kl}\mathbf{x}[k]\Big) \Big/ \Big(\sum_{k=1}^{n} \widehat{z}_{kl}\Big)$$

can be solved for $\widehat{\underline{\boldsymbol{\theta}}}[l]$.

Table 5.3 summarizes the Bregman soft clustering algorithm.

5.4.1 The mixed data-type case

We consider again the mixed data-type case. The E-step and the first part of the M-step yield the same results as for SP-PCA. In the second part of the M-step, the component parameters $\underline{\boldsymbol{\theta}}[l], l = 1, \ldots, m$, are estimated in the

Table 5.3 The Bregman soft clustering algorithm.

Algorithm: Bregman Soft Clustering [14, 16]

Input: a set of observations $\{\mathbf{x}[k]\}_{k=1}^{n} \subseteq \mathbb{R}^d$, an exponential family distribution $p(\cdot)$ defined by its cumulant generating function $G(\cdot)$, a number of atoms m.

Output: the NPML estimator that maximizes the complete log-likelihood function $L^{(c)}\big(\mathcal{Q}, \{\mathbf{z}_k\}_{k=1}^{n}\big)$: $\widehat{\mathcal{Q}} = \big\{\widehat{\underline{\boldsymbol{\theta}}}[l], \widehat{\pi}_l\big\}_{l=1}^{m}$.

Method:

Initialize $\big\{\underline{\boldsymbol{\theta}}[l], \pi_l\big\}_{l=1}^{m}$ with $\pi_l \geq 0$ for all l and $\sum_{l=1}^{m} \pi_l = 1$; $\underline{\boldsymbol{\theta}}[l] \in \boldsymbol{\Theta}$ for all l;

repeat

{The Expectation Step}

for $k = 1$ to n **do**

for $l = 1$ to m **do**

$\widehat{z}_{kl} \longleftarrow p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l / \sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r$

end for

end for

{The Maximization Step}

for $l = 1$ to m **do**

$\widehat{\pi}_l \longleftarrow (1/n)\sum_{k=1}^{n} \widehat{z}_{kl}$

$\underline{\boldsymbol{\theta}}[l] \longleftarrow$ solve for $\underline{\boldsymbol{\theta}}[l]$:

$G'(\underline{\boldsymbol{\theta}}[l]) = \sum_{k=1}^{n} \widehat{z}_{kl}\mathbf{x}[k] / \sum_{k=1}^{n} \widehat{z}_{kl}$

end for

until *convergence*;

return $\widehat{\mathcal{Q}} = \big\{\widehat{\underline{\boldsymbol{\theta}}}[l], \widehat{\pi}_l\big\}_{l=1}^{m}$.

following way:

$$\begin{aligned}\widehat{\underline{\boldsymbol{\theta}}}[l] &= \arg\max_{\underline{\boldsymbol{\theta}}[l]} \sum_{k=1}^{n}\sum_{r=1}^{m} \widehat{z}_{kr} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big) \\ &= \arg\max_{\underline{\boldsymbol{\theta}}[l]} \Big\{ \sum_{k=1}^{n}\sum_{r=1}^{m} \widehat{z}_{kr} \log p^{(\mathbf{1})}\big(\mathbf{x}^{(\mathbf{1})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{1})}[r]\big) \\ &\qquad + \sum_{k=1}^{n}\sum_{r=1}^{m} \widehat{z}_{kr} \log p^{(\mathbf{2})}\big(\mathbf{x}^{(\mathbf{2})}[k]|\underline{\boldsymbol{\theta}}^{(\mathbf{2})}[r]\big)\Big\},\end{aligned}$$

with $\log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big) = \underline{\boldsymbol{\theta}}[r]\mathbf{x}[k]^T - G\big(\underline{\boldsymbol{\theta}}[r]\big)$. Using the convexity properties of $G(\cdot)$, it is easily shown that:

$$G'(\widehat{\underline{\boldsymbol{\theta}}}[l]) = \Big(\sum_{k=1}^{n} \widehat{z}_{kl}\mathbf{x}[k]\Big) \Big/ \Big(\sum_{k=1}^{n} \widehat{z}_{kl}\Big)$$

can be solved for $\widehat{\underline{\boldsymbol{\theta}}}[l]^{(\mathbf{1})}$ and $\widehat{\underline{\boldsymbol{\theta}}}[l]^{(\mathbf{2})}$ by changing $\mathbf{x}$ to $\mathbf{x}^{(\mathbf{1})}$, respectively to $\mathbf{x}^{(\mathbf{2})}$, $G'(\cdot)$ to $G^{(\mathbf{1})'}(\cdot)$, respectively to $G^{(\mathbf{2})'}(\cdot)$.

5.5 A unifying framework

Within the proposed hierarchical Bayes graphical model framework, exponential PCA, SP-PCA and Bregman soft clustering are not separate uncorrelated algorithms but different manifestations of model assumptions and parameter choices.

Figure 5.1 considers the number of atoms as a common characteristic for comparison purposes. The exponential PCA technique corresponds to a classical approach to the GLS estimation problem. The classical approach can be seen as an extreme case of the Bayesian approach for which the probability density function $\pi(\boldsymbol{\theta})$ is a delta function (one per data point) and the total number of distinct natural parameter values m equals the number of data points n, i.e., $m = n$. While the $m < n$ parameters of the Bayesian approach consistent with SP-PCA and the Bregman soft clustering techniques are shared by all the data points, the classical approach assigns one parameter point to each data point (hence

$m = n$). The Bregman soft clustering approach considers an even smaller number of natural parameters or atoms than SP-PCA. Since its primary goal is clustering, the atoms play the role of cluster centers in parameter space and their total number is generally small. Furthermore, both exponential PCA and SP-PCA impose a low-dimensional (unknown) latent variable subspace in their structure. However, Bregman soft clustering does not impose this lower dimensional constraint and hence can be seen as a degenerate case.

It becomes clear while looking at Table 5.1, Table 5.2 and Table 5.3 shown previously that both SP-PCA and Bregman soft clustering utilize the EM algorithm for estimation purposes whereas exponential PCA does not. Indeed, because exponential PCA assumes a classical approach, no point-mass probabilities need to be estimated.

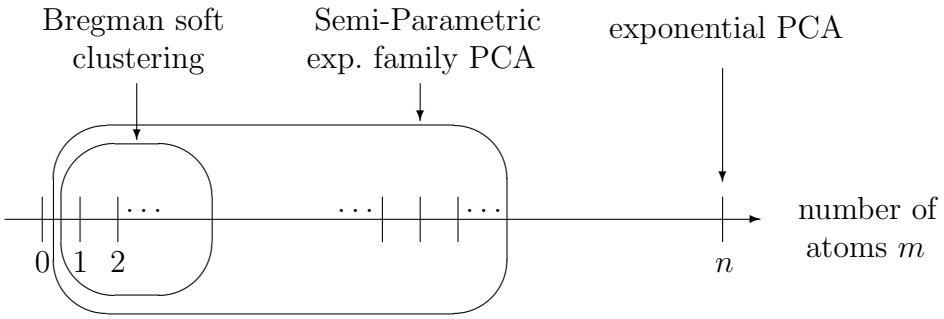

Figure 5.1 General point of view based on the number of atoms used in the GLS estimation.

Figure 5.2 presents connections between the techniques of interest in terms of an algorithmic perspective. As explained previously, Semi-Parametric exponential family PCA and Bregman soft clustering utilize the EM algorithm for estimation purposes whereas exponential PCA does not. The dotted line between exponential PCA and NPML suggests that exponential PCA can be seen as an extreme case of the general Bayesian approach for which the number of natural

parameters equals the number of data points, $m = n$. Both exponential PCA and SP-PCA impose a lower dimensional latent variable subspace in their structure, hence the need for the Newton-Raphson iterative algorithm (expressed as NR on the figure). The two arrows going back and forth between NPML and Finite Mixture Models suggest that the two methods are similar. Vertex Direction Method (VDM) and Vertex Exchange Method (VEM) are alternative suitable algorithms for constructing the NPML estimates [34].

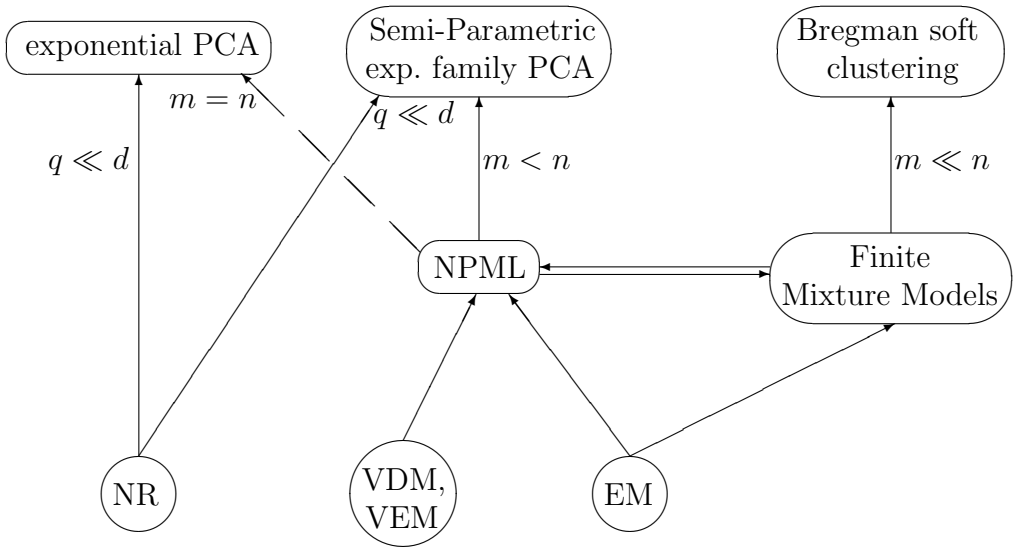

Figure 5.2 Algorithmic connections between Bregman soft clustering, exponential PCA and Semi-Parametric exponential family PCA.

Figure 5.3 offers a detailed diagram of the successive steps of Bregman soft clustering, SP-PCA and exponential PCA. It illustrates how different assumptions and parameter choices generate different approaches for a common problem, i.e., different algorithms.

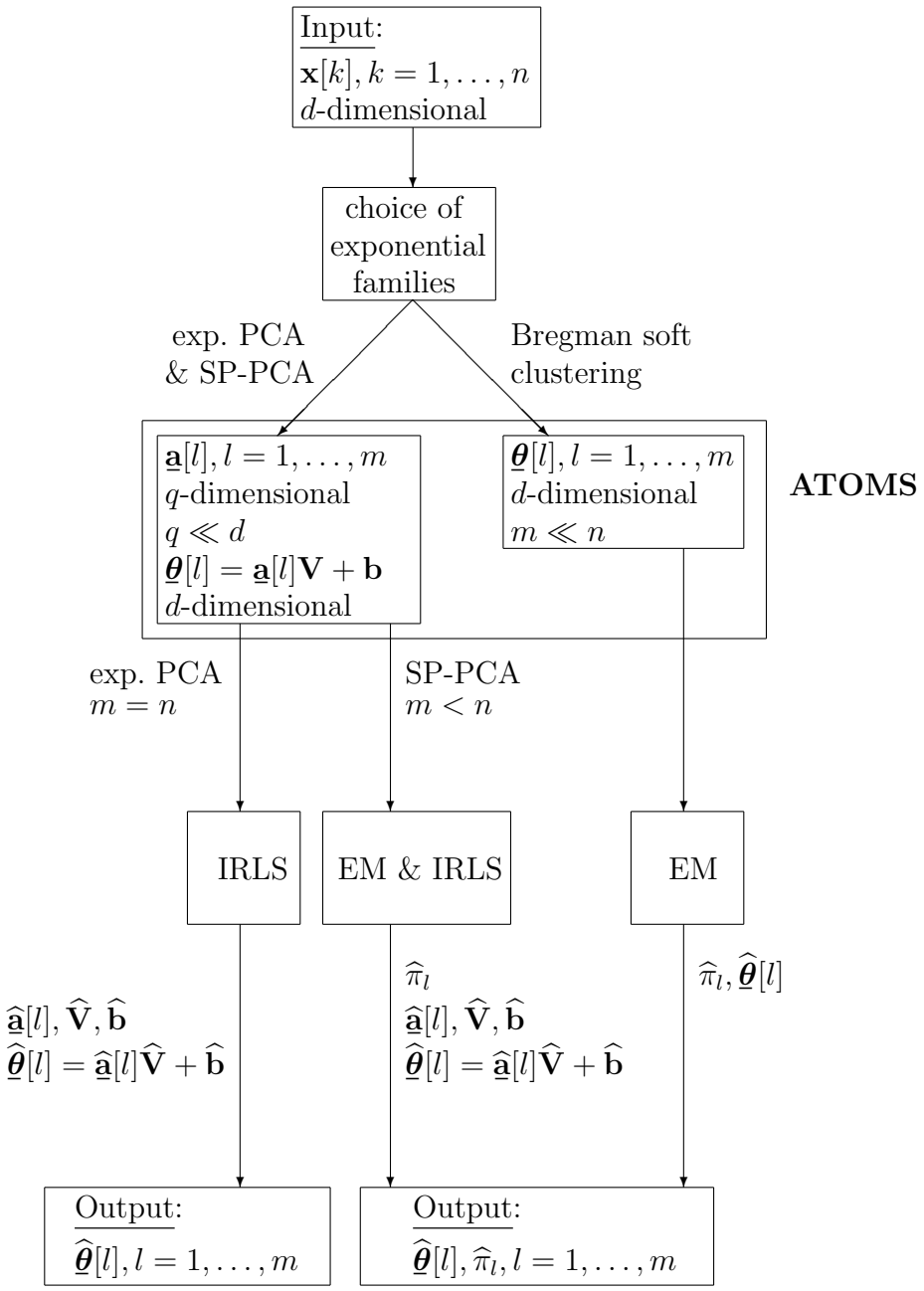

Figure 5.3 Detailed diagram of the successive steps comparing Bregman soft clustering, SP-PCA and exponential PCA approaches.

5.6 Application: experimental clustering results on synthetic data

This application compares the relative performances of exponential PCA, SP-PCA and Bregman soft clustering in a mixed data set clustering problem with two data types and demonstrates how exponential PCA with the addition of a non-parametric estimation of the point-mass probabilities can exceed SP-PCA in performance.

We first consider a synthetic $d = 3$-dimensional data set with a lower dimensional subspace of dimension $q = 1$. The first data feature is Poisson distributed, the second and third features are Gaussian distributed. The data has $n = 500$ points and is composed of two mixture components with parameters $\underline{\boldsymbol{\theta}}[1]$ and $\underline{\boldsymbol{\theta}}[2]$ constrained to the lower dimensional subspace.

We first use exponential PCA. However, exponential PCA does not estimate point-mass probabilities. We use a non-parametric density estimation technique based on a kernel smoothing method to estimate the point-mass probabilities using the support points values $\underline{\mathbf{a}}[k]$, $k = 1, \ldots, n$, obtained by exponential PCA. Figure 5.4 shows that the non-parametric density estimation exhibits a definite two-component shape. The dotted lines represent the correct values $\underline{\mathbf{a}}[1]$ and $\underline{\mathbf{a}}[2]$. We can then estimate the values of $\underline{\mathbf{a}}[1]$ and $\underline{\mathbf{a}}[2]$ as well as their mixing distributions π_1 and π_2 using a simple `k-means` algorithm, with the $\pi_1 + \pi_2 = 1$ assumption.

Figure 5.5 presents the histogram of the estimated point-mass probabilities obtained with SP-PCA, $m = 2$.

Table 5.4 shows detailed results for this synthetic data setting ("modified" means the extension to mixed data sets of the algorithm): the mixing distributions or point-mass probabilities π_1 and π_2, the latent variable or point of support values $\underline{\mathbf{a}}[1]$ and $\underline{\mathbf{a}}[2]$, the parameter values $\underline{\boldsymbol{\theta}}[1]$ and $\underline{\boldsymbol{\theta}}[2]$ as well as the sine of the angle between the estimated lower dimensional subspace and the correct subspace.

Bregman soft clustering does not have the lower dimensional subspace constraint, and hence does not exhibit a sine or the latent variables values in Table 5.4. The estimation quality of the $\underline{\boldsymbol{\theta}}[1]$, $\underline{\boldsymbol{\theta}}[2]$ and π_1, π_2 values defines the clustering performance. For this simple Poisson-Gaussian mixed data setting, both exponential PCA and Bregman soft clustering seem to perform better than SP-PCA: the SP-PCA obtained parameter values for $\underline{\boldsymbol{\theta}}[2]$ are far from the original values, contrary to exponential PCA and Bregman soft clustering.

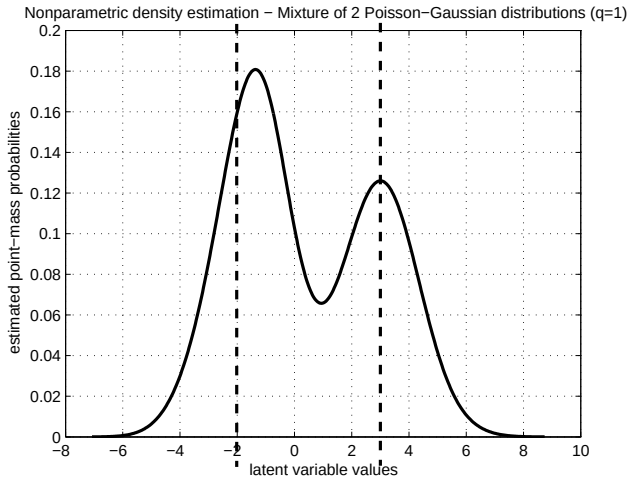

Figure 5.4 Non-parametric estimation of the point-mass probabilities obtained with exponential PCA (dotted: correct cluster centers).

Results for a second experiment are shown in Table 5.5 for a Binomial-Gaussian mixed data set created in a similar fashion as the Poisson-Gaussian mixed data set (the parameter N is set to 10 for the Binomial component). Again, exponential PCA exceeds SP-PCA in clustering performance.

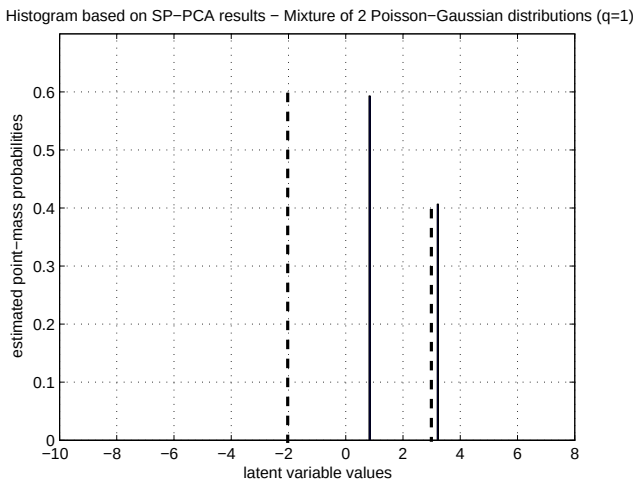

Figure 5.5 Histogram of the estimated point-mass probabilities obtained with SP-PCA (dotted: correct cluster values).

Table 5.4 Clustering results for a Poisson-Gaussian mixed data set.

	π_1; π_2	$\underline{\mathbf{a}}[1]$; $\underline{\mathbf{a}}[2]$	$\underline{\boldsymbol{\theta}}[1]$; $\underline{\boldsymbol{\theta}}[2]$	sin
correct	0.4	3	[1.9404, 1.6148, 1.6210]	
model values	0.6	-2	[-1.2936, -1.0765, -1.0806]	
modified	0.4107	3.0009	[1.6235, 1.8648, 1.7007]	0.1368
exponential PCA	0.5893	-1.3725	[-0.7425, -0.8529, -0.7778]	
modified	0.3724	3.2170	[2.1732, 1.5715, 1.7768]	0.058663
SP-PCA	0.6276	0.8355	[0.5644, 0.4081, 0.4614]	
modified Bregman	0.4069		[1.9317, 1.7162, 1.5585]	
soft clustering	0.5931		[-1.1061, -1.0802, -1.0304]	

Table 5.5 Clustering results for a Binomial-Gaussian mixed data set.

	π_1; π_2	$\underline{\mathbf{a}}[1]$; $\underline{\mathbf{a}}[2]$	$\underline{\boldsymbol{\theta}}[1]$; $\underline{\boldsymbol{\theta}}[2]$	sin
correct	0.4	1	[0.8914, 0.1688, 0.4206]	
model values	0.6	-2	$[-1.7828, -0.3375, -0.8412]$	
modified	0.4475	0.8559	[0.7796, 0.1166, 0.3334]	0.049038
exponential PCA	0.5525	-1.9972	$[-1.8193, -0.2721, -0.7779]$	
modified	0.3978	-0.9548	$[-0.9046, -0.0989, -0.2890]$	0.1455
SP-PCA	0.6022	-3.1821	$[-3.0148, -0.3296, -0.9633]$	
modified Bregman	0.3973		[0.82252, 0.144, 0.41004]	
soft clustering	0.6027		$[-1.8072, -0.3089, -0.9816]$	

Acknowledgement

Chapter 5, in part, is a reprint of the material as it appears in "A unifying viewpoint of some clustering techniques using Bregman divergences and extensions to mixed data sets," C. Levasseur, B. Burdge, K. Kreutz-Delgado and U. F. Mayer, in *Proceedings of the First IEEE International Workshop on Data Mining and Artificial Intelligence (DMAI)*, pp. 56–63, Dec. 2008, "Generalized statistical methods for mixed exponential families, part I: theoretical foundations," C. Levasseur, K. Kreutz-Delgado and U. F. Mayer, submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Sept. 2009 and "Generalized statistical methods for mixed exponential families, part II: applications," C. Levasseur, U. F. Mayer and K. Kreutz-Delgado, submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Sept. 2009. The dissertation author was the primary author of these papers.

6 Conclusions

This dissertation considered the problem of learning the underlying statistical structure of data of mixed types for fitting a generative model, and for both supervised and unsupervised data-driven decision making. A new unified theoretical model called Generalized Linear Statistics was established using properties of exponential family distributions. The primary contributions are summarized below.

6.1 Contributions of this thesis

The proposed statistical modeling approach called *Generalized Linear Statistics* (GLS) is a generalization and amalgamation of techniques from classical linear statistics, Generalized Linear Models (GLMs) and latent variable modeling. This is a *nonlinear* methodology which exploits the split that occurs for exponential family distributions between the *data space* and the *parameter space* as soon as one leaves the domain of purely Gaussian random variables. Nonlinear problems can then be attacked using classical linear and other standard statistical tools applied to data that have been *mapped into the parameter space*, which is assumed to still have a natural, flat Euclidean space structure. This dissertation demonstrated the ability to learn a generative GLS model that captures the statistical structure of the data, using this knowledge to gain insight into the problem domain and to develop effective algorithms capable of data-driven techniques in domains involving mixed data-type, labeled and unlabeled data sets. Specifically, this work considered mixed

data-type records which have both continuous (e.g., Exponential and Gaussian) and discrete (e.g., count and binary) components. The approach employed here allows for the data components to have different parametric forms by using the large range of exponential family distributions. Exponential families have many useful and important mathematical properties which were fruitfully exploited to obtain Maximum Likelihood (ML) estimates of the GLS model parameters.

The specific Generalized Linear Statistics (GLS) framework developed in this dissertation represents a subclass of graphical model techniques and is equivalent to a computationally tractable mixed exponential families data-type hierarchical Bayes graphical model with latent variables constrained to a low-dimensional parameter subspace. The exponential family Principal Component Analysis (exponential PCA) technique of [10], the Semi-Parametric exponential family Principal Component Analysis (SP-PCA) technique of [15] and the Bregman soft clustering method presented in [14] were demonstrated not to be separate unrelated algorithms, but rather different manifestations of model assumptions and parameter choices within the GLS framework. Because of this insight, the three algorithms could be extended to readily derive novel extensions that deal with the important mixed data-type case. Several synthetic data examples of mixed types were considered, demonstrating that exponential PCA, with the addition of a non-parametric estimation tool, rivals SP-PCA and Bregman soft clustering in terms of clustering performance for some data sets.

This work exposed in detail the convex optimization problem related to fitting one extreme case of the GLS model to a set of data. This extreme case of the GLS model is similar to exponential family Principal Component Analysis, proposed in [10], and is characterized by the fact that each data point is mapped to one (generally different) parameter point in parameter space, whereas the general GLS case considers a set of parameter points shared by all the data points. In light of the significant numerical difficulties associated with the cyclic-coordinate descent-like algorithm based on Bregman divergence properties proposed in [10],

especially in the mixed data-type case, this dissertation focused on an algorithm based on Iterative Reweighted Least Squares (IRLS), an approach commonly used in the GLMs literature [4, 38, 39]. Using an IRLS-based learning algorithm makes it possible to tractably attack the more general problem of prediction in a mixed data-type environment. Since the optimal model parameter values for our optimization problem may be non-finite [10], a penalty function was introduced that defined and placed a set of constraints into the loss function via a penalty parameter in a way so that any divergence to infinity is avoided. Additionally, for several exponential family distributions with natural restrictions on their parameter, a positivity constraint on the natural parameter values was introduced. Synthetic data examples for several exponential family distributions in both mixed and non-mixed data-type cases were presented and generative models were fit to the data. Furthermore, an unsupervised minority class detection technique to be performed in the parameter space, rather than in the data space, as in more classical approaches, was proposed. A synthetic data example was created, demonstrating that there are domains for which classical linear techniques used in the data space, such as PCA, perform significantly worse than the new proposed parameter space technique.

Once the generative GLS model was learned, the knowledge gained about the statistical structure of data was used to perform effective classification on data sets from the University of California, Irvine machine learning repository. The text categorization and classification problems attacked in Appendix D illustrated the benefits of making decisions in parameter space rather than in data space, as with more classical approaches, and demonstrated the utility of the GLS approach for experiments on real data sets in both supervised and unsupervised settings. Support Vector Machines (SVMs) also make decisions in a non-data space. However, the SVMs technique does not provide any better understanding of the data, despite often promising the highest degree of accuracy. An advantage of learning a generative model of the data, as with GLS, is that generating synthetic data for the

purposes of developing and training classifiers with the same statistical structure as the original data becomes possible. This is particularly useful in cases where data are very difficult or expensive to obtain, or when the original data are proprietary and cannot be directly used for publication purposes in open literature.

6.2 Future work

While this dissertation established a complete well-rounded theory, several interesting and important associated problems remain open, some of which are listed below.

- Bregman distances and convexity properties: in addition to Lafferty et al [88, 89], Bauschke has intensively studied optimization problems using Bregman distances and their convexity properties [90–92]. In particular, the existence of a convex dual optimization problem associated with the minimization of Bregman distances might be of interest to the Generalized Linear Statistics (GLS) parameters estimation problem.

- Modeling overdispersion: the problem of overdispersion, or more rarely underdispersion, is often mentioned when the variance observed on fitting the model is greater, or more rarely smaller, than anticipated [27,93,94], i.e., the data samples are strongly heterogenous. This phenomenon can be accounted for in Generalized Linear Statistics (GLS) by adding a *dispersion parameter* ϕ. The definition of an exponential family then becomes $p(x|\theta,\phi) = \exp\left\{\left(\theta x - G(\theta)\right)/\phi\right\} h(x)$. The dispersion parameter can be estimated by maximum likelihood estimation and regularly updated within the iterative algorithm proposed for GLS. For a Gaussian distribution, the dispersion parameter corresponds to the variance, i.e., $\phi = \sigma^2$. The dispersion parameter can be taken as a correction factor and as such, a dispersion parameter with a value greater than 1 for a unit-variance Gaussian model assumption would mean that overdispersion is observed, whereas a dispersion parameter value

of 1 would mean that the model is correct.

- Sparse representation in parameter space: sparse representation of signals have caught researchers' attention in recent years. Solving the sparse representation problem corresponds to finding the most compact representation of a signal in terms of a linear combination of atoms in an overcomplete dictionary. In other words, the chosen representation should be characterized by a high number of zero-valued elements, i.e., the weights in the linear combination are mostly zero-valued. The work proposed in [95–97] is based on the use of FOCUSS (FOCal Underdetermined System Solver), an algorithm designed to obtain suboptimal sparse solutions to the underdetermined linear inverse problem $\boldsymbol{\theta} = \mathbf{aV}$ for known $(q \times d)$-matrix $\mathbf{V}$ with $d \leq q$, typically $d \ll q$. Instead of a wide $\mathbf{V}$ matrix as in GLS ($q \ll d$), the matrix $\mathbf{V}$ is now considered tall. We can imagine a generalization of FOCUSS from $\boldsymbol{\mu} \triangleq E[\mathbf{x}|\boldsymbol{\theta}] = \boldsymbol{\theta} = \mathbf{aV}$ to $\boldsymbol{\mu} \triangleq E[\mathbf{x}|\boldsymbol{\theta}] = g(\boldsymbol{\theta})$ & $\boldsymbol{\theta} = \mathbf{aV}$ with $g(\cdot)$ the link function of any exponential family distribution. Along these lines of inquiry, there already exists a sparse alternative to Principal Component Analysis (PCA) [98, 99]. Just as PCA was generalized to the exponential family [10], it would be interesting to look at the problem of generalizing existing sparse PCA techniques to provide the option of a sparse PCA representation in parameter space.

A Exponential families

A.1 Motivation in a learning environment

To model a distribution over some data set $\mathcal{X} \subset \mathbb{R}^d$, a prior distribution π can be set over $\mathcal{X}$, for example the Uniform distribution. Then, several "features" are measured, $T_1 : \mathcal{X} \to \mathbb{R}, T_2 : \mathcal{X} \to \mathbb{R}, \ldots, T_N : \mathcal{X} \to \mathbb{R}$, for which the average outcome with respect to the Lebesgue measure is computed and referred to as b_i for $i = 1, \ldots, N$, with $|b_i| < \infty$. In order to choose the best distribution p^* to model the data, the following conditions must be satisfied:

1. $E_{p^*}\big[T_i(\mathbf{x})\big] = b_i \, \forall i$, where $\mathbf{x} \in \mathcal{X}$,

2. p^* is as close to the prior distribution π as possible,

i.e., $p^* = \min_p \mathrm{D}(p||\pi)$, and the minimum p^* is taken to satisfy condition 1, where $\mathrm{D}(p||\pi)$ is the Kullback-Leibler divergence between a distribution p and the prior distribution π. The Kullback-Leibler divergence is a measure of relative entropy, or information divergence, between two distributions (further discussed in Appendix A.4). Therefore, the distribution p^* has to be the solution of the following constraint optimization problem:

$$\min_p \mathrm{D}(p||\pi) \,\text{s.t.}\, E_p\big[T_i(\mathbf{x})\big] = b_i, \; i = 1, \ldots, N. \tag{A.1}$$

A unique solution to the minimization problem (A.1) exists and is a distribution of the form [100]:

$$p^*(\mathbf{x}) = \frac{1}{z_{\boldsymbol{\eta}}} \exp\left(\sum_{i=1}^{N} \eta_i T_i(\mathbf{x})\right) \pi(\mathbf{x}),$$

where $\exp(y) = \mathrm{e}^y$, $z_{\boldsymbol{\eta}}$ is a normalizer and $\{\eta_i\}_{i=1}^N \subseteq \mathbb{R}$ a set of coefficients. As $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_N]$ varies over $\mathbb{R}^N$, the set $\left\{ \frac{1}{z_{\boldsymbol{\eta}}} \exp\left(\sum_{i=1}^N \eta_i T_i(\mathbf{x}) \right) \pi(\mathbf{x}) \right\}$ forms the exponential family of distributions generated by π. An appropriate choice of the prior distribution π defines the Gaussian distribution, or the Poisson distribution, or any other exponential family distribution, as the best model for the data set $\mathcal{X}$. Details on these choices follow in the next section.

A.2 Standard exponential families

A.2.1 Probability and measure

A probability space is a triple $(\mathcal{X}, \mathcal{F}, P)$ where:

(i) the sample space $\mathcal{X}$ is a nonempty set whose elements are known as *outcomes*,

(ii) $\mathcal{F}$ is a nonempty σ-algebra of subsets of $\mathcal{X}$; its elements are measurable sets $A \subseteq \mathcal{X}$ called *events*, which are sets of outcomes for which one can ask a probability; it is closed under (1) complement, if $A \in \mathcal{F}$ then $\mathcal{X} \setminus A \in \mathcal{F}$, and (2) countable union, if $A_i \in \mathcal{F}$ then $\bigcup\limits_i A_i \in \mathcal{F}$,

(iii) the probability measure P is a function from $\mathcal{F}$ to the real numbers that assigns to each event $A \in \mathcal{F}$ a probability between 0 and 1, $P : \mathcal{F} \to [0, 1]$.

A measure generalizes the concepts of the length of a line, the area of a spread, or the volume of a 3-dimensional object and can be considered a consistent notion of size. A measure ν on a σ-algebra $\mathcal{F}$ satisfies the following two properties:

- $\nu(A) \geq \nu(\varnothing) = 0$ for all $A \in \mathcal{F}$,
- countable additivity: if the system of sets A_i is disjoint, then $\nu\left(\bigcup\limits_{i=1}^{\infty} A_i \right) = \sum\limits_{i=1}^{\infty} \nu\big(A_i\big)$.

The counting measure is defined as follows: let $\mathcal{X}$ be any (possibly infinite) countable set, then $\mathcal{F} = 2^{\mathcal{X}}$, the powerset of $\mathcal{X}$, is the class of all subsets of $\mathcal{X}$ and $\nu(A)$ is

the number of points of A if A is finite, otherwise $\nu(A) = \infty$, and $A \in \mathcal{F}$. Another well-known measure is the Lebesgue measure. The Borel subsets $\mathcal{B}$ of $\mathbb{R}$ are defined as the smallest σ-algebra containing all the (closed) intervals. Similarly, the Borel subsets of $\mathbb{R}^d$ are defined as the smallest σ-algebra containing all closed rectangles $\big\{(x_1, \ldots, x_d) : a_i \leq x_i \leq b_i, i = 1, \ldots, d\big\}$, $-\infty < a_i < b_i < \infty$. The Lebesgue measure is defined on a larger family $\mathcal{F}$ that adds to the Borel subsets all sets A for which there is a set $B \in \mathcal{B}$ with $\nu(B) = 0$ and $A \subset B$ [101]. Then, the Lebesgue measure in $\mathbb{R}$ is the measure of the interval $[a, b]$, i.e., $\nu\big((a, b)\big) = b - a$, and the Lebesgue measure in $\mathbb{R}^d$ is $\nu\big((a_1, b_1) \times (a_2, b_2) \times \cdots \times (a_d, b_d)\big) = \prod_{i=1}^{d}(b_i - a_i)$, i.e., the volume of the rectangle. These are not probability measures since $\nu(\mathcal{X}) = \infty$ in both cases, but they are σ-finite, i.e., $\mathcal{X}$ is the countable union of measurable sets of finite measure.

A.2.2 Standard exponential families definition

Definition 1 ([54, 55])**.** *Let ρ be a σ-finite measure on the Borel subsets of $\mathbb{R}^d$. Then, ρ generates an exponential family in the following way:*

(i) Let $\mathcal{N} = \big\{\boldsymbol{\theta} \in \mathbb{R}^d : \int_{\mathcal{X}} e^{\boldsymbol{\theta}\cdot\mathbf{x}}\rho(\mathrm{d}\mathbf{x}) < \infty\big\}$, the natural parameter space*,*

(ii) $G(\boldsymbol{\theta}) = \log \int_{\mathcal{X}} e^{\boldsymbol{\theta}\cdot\mathbf{x}}\rho(\mathrm{d}\mathbf{x})$, the log-partition function, or the cumulant generating function,

(iii) $p(\mathbf{x}|\boldsymbol{\theta}) = e^{\boldsymbol{\theta}\cdot\mathbf{x} - G(\boldsymbol{\theta})}$ for $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\theta} \in \mathcal{N}$, the probability densities with respect to ρ.

(iv) For any $\boldsymbol{\Theta} \subseteq \mathcal{N}$, called the restricted parameter space*, the family of probability densities $\{p(\cdot|\boldsymbol{\theta}) : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ is called a d-dimensional* standard exponential family *(of probability densities).*

This definition of standard exponential families requires the characterization of an elaborate measure ρ for each family. However, it is usually possible to assume the measure to be absolutely continuous with respect to either the Lebesgue

measure or the counting measure ν:

$$\rho \ll \nu : \exists\, h \,\text{s.t.}\, \int_A f(\mathbf{x})\rho(\mathrm{d}\mathbf{x}) = \int_A f(\mathbf{x})h(\mathbf{x})\nu(\mathrm{d}\mathbf{x})\; \forall f,\, \forall A \in \mathcal{F}.$$

The function h is called the Radon-Nikodym derivative of ρ with respect to ν. Then the standard exponential families can be defined as follows:

Definition 2 ([76])**.** *Let ν be either the Lebesgue measure or the counting measure on the Borel subsets of $\mathbb{R}^d$ and let $h(\cdot)$ be a function from $\mathcal{X} \subseteq \mathbb{R}^d$ to $\mathbb{R}$. Then, ν and $h(\cdot)$ generate an exponential family in the following way:*

(i) Let $\mathcal{N} = \left\{\boldsymbol{\theta} \in \mathbb{R}^d : \int_{\mathcal{X}} e^{\boldsymbol{\theta}\cdot\mathbf{x}} h(\mathbf{x})\nu(\mathrm{d}\mathbf{x}) < \infty\right\}$, the natural parameter space*,*

(ii) $G(\boldsymbol{\theta}) = \log \int_{\mathcal{X}} e^{\boldsymbol{\theta}\cdot\mathbf{x}} h(\mathbf{x})\nu(\mathrm{d}\mathbf{x})$, the log-partition function, or the cumulant generating function,

(iii) $p(\mathbf{x}|\boldsymbol{\theta}) = e^{\boldsymbol{\theta}\cdot\mathbf{x} - G(\boldsymbol{\theta})} h(\mathbf{x})$ for $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\theta} \in \mathcal{N}$, the probability densities with respect to ν.

(iv) For any $\boldsymbol{\Theta} \subseteq \mathcal{N}$, called the restricted parameter space*, the family of probability densities $\{p(\cdot|\boldsymbol{\theta}) : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ is called a d-dimensional* standard exponential family *(of probability densities).*

The factor $h(\mathbf{x})$ does not depend on the parameter $\boldsymbol{\theta}$ and absorbing it into the measure ν transforms Definition 2 into Definition 1. Continuous exponential family probability densities are defined with respect to the Lebesgue measure whereas discrete probability densities are defined with respect to the counting measure on $\mathbb{N}_0$ or a subset thereof.

Other d-dimensional exponential family definitions present in the literature can be reduced by sufficiency, reparameterization, and proper choice of $h(\cdot)$ to the definition of a d-dimensional standard exponential family [55]. The term *canonical form* is also frequently used to refer to standard exponential family distributions [76]. Well-known exponential families include the Gaussian distribution, the Bernoulli distribution, the Binomial distribution, the Chi-square distribution,

the Exponential distribution, the Inverse Gaussian distribution and the Poisson distribution.

The parameter $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is sometimes referred to as a *canonical parameter*, and $\mathbf{x} \in \mathcal{X}$ is sometimes called a *canonical observation*. The exponential family generated by $\boldsymbol{\Theta}$ is called *full* if $\boldsymbol{\Theta} = \mathcal{N}$. The family is called *regular* if it is full and if $\mathcal{N}$ is open, i.e., if $\mathcal{N} = \mathcal{N}^\circ$, where $\mathcal{N}^\circ$ denotes the interior of $\mathcal{N}$, defined as $\operatorname{int}\mathcal{N} = \{\cup Q : Q \subset \mathcal{N}, Q \text{ is open}\}$. Let $\mathcal{D}$ be the convex support of h. A d-dimensional standard exponential family is called *minimal* is $\dim\mathcal{N} = \dim\mathcal{D} = d$.

The Multinomial distribution with parameters $(N, \pi_1, \ldots, \pi_k)$, for example, considers N balls thrown into k bins, with π_i being the probability that any given ball falls into bin i for all i, and with the constraint $\pi_1 + \cdots + \pi_k = 1$. The observation vector $\mathbf{x} = [x_1, \ldots, x_k] \in \mathbb{Z}^k_{\geq 0}$ describes the number of balls in each bin, with the constraint $x_1 + \cdots + x_k = N$. The distribution takes the following form:

$$p(\mathbf{x}|\pi_1, \ldots, \pi_k) = \binom{N}{x_1, \ldots, x_k} \pi_1^{x_1} \cdot \ldots \cdot \pi_k^{x_k}.$$

Following Definition 2, this Multinomial family is characterized as follows with parameter $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_k]$:

$$p(\mathbf{x}|\pi_1, \ldots, \pi_k) = \binom{N}{x_1, \ldots, x_k} \exp\{x_1 \log(\pi_1) + \ldots + x_k \log(\pi_k)\}.$$

Hence,

$$\begin{aligned}
h(\mathbf{x}) &= \binom{N}{x_1, \ldots, x_k}, \\
\theta_i &= \log \pi_i, i = 1, \ldots, k \quad \text{with} \quad \boldsymbol{\Theta} = \Big\{[\log \pi_i]_{i=1}^k : 0 < \pi_i, \sum_{i=1}^k \pi_i = 1\Big\}, \\
G(\boldsymbol{\theta}) &= \log \sum_{x_1 + \cdots + x_k = N} \binom{N}{x_1, \ldots, x_k} \mathrm{e}^{x_1\theta_1} \cdot \ldots \cdot \mathrm{e}^{x_k\theta_k} \\
&= \log(\mathrm{e}^{\theta_1} + \cdots + \mathrm{e}^{\theta_k})^N = N \log\Big(\sum_{i=1}^k \mathrm{e}^{\theta_i}\Big), \\
\mathcal{N} &= \mathbb{R}^k.
\end{aligned}$$

This exponential family is not full because $\boldsymbol{\Theta} \neq \mathcal{N} = \mathbb{R}^k$. However, the family is not minimal either since $\dim \mathcal{D} = k-1 < k$ because of the constraint $x_1+\cdots+x_k = N$. To reduce this family to a minimal family, let $\mathbf{x}^\star = [x_1^\star, \ldots, x_{k-1}^\star] \in \mathbb{Z}_{\geq 0}^{k-1}$ be defined as $x_1^\star = x_1, \ldots, x_{k-1}^\star = x_{k-1}$. It is essentially equivalent to $\mathbf{x}$ since $x_k = N - \sum_{i=1}^{k-1} x_i^\star$. Let $\theta_i^\star = \theta_i - \theta_k$, and let

$$h^\star(\mathbf{x}^\star) = \binom{N}{x_1^\star, \ldots, x_{k-1}^\star, N - \sum_{i=1}^{k-1} x_i^\star}.$$

Then, the density of $\mathbf{x}^\star$ with parameter $\boldsymbol{\theta}^\star = [\theta_1^\star, \ldots, \theta_k^\star]$ is

$$p^\star(\mathbf{x}^\star|\boldsymbol{\theta}^\star) = \mathrm{e}^{\boldsymbol{\theta}^\star \cdot \mathbf{x}^\star - G^\star(\boldsymbol{\theta}^\star)} h^\star(\mathbf{x}^\star) \quad \text{with} \quad G^\star(\boldsymbol{\theta}^\star) = N \log\Big(1 + \sum_{i=1}^{k-1} \mathrm{e}^{\theta_i^\star}\Big).$$

This is a full minimal standard exponential family with $\mathcal{N} = \mathbb{R}^{k-1}$. Note that

$$\pi_i = \mathrm{e}^{\theta_i^\star} / \Big(1 + \sum_{j=1}^{k-1} \mathrm{e}^{\theta_j^\star}\Big),\ i = 1, \ldots, k-1,$$
$$\pi_k = 1 / \Big(1 + \sum_{j=1}^{k-1} \mathrm{e}^{\theta_j^\star}\Big).$$

It is well-known [54] that the parameter space $\mathcal{N}$ is a convex set for the exponential families listed in the previous paragraph and that $G(\boldsymbol{\theta})$ is a convex function on $\mathcal{N}$ (strictly convex if the family is minimal): $\forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathcal{N}$ and $\forall \alpha \in (0,1)$,

$$\begin{aligned} G\big(\alpha\boldsymbol{\theta}_1 + (1-\alpha)\boldsymbol{\theta}_2\big) &= \log \int \mathrm{e}^{(\alpha\boldsymbol{\theta}_1 + (1-\alpha)\boldsymbol{\theta}_2)\cdot\mathbf{x}} \nu(\mathrm{d}\mathbf{x}) \\ &= \log \int \mathrm{e}^{\alpha\boldsymbol{\theta}_1\cdot\mathbf{x}} \cdot \mathrm{e}^{(1-\alpha)\boldsymbol{\theta}_2\cdot\mathbf{x}} \nu(\mathrm{d}\mathbf{x}) \\ &\leq \log \Big[\int \mathrm{e}^{\boldsymbol{\theta}_1\cdot\mathbf{x}} \nu(\mathrm{d}\mathbf{x})\Big]^\alpha \Big[\int \mathrm{e}^{\boldsymbol{\theta}_2\cdot\mathbf{x}} \nu(\mathrm{d}\mathbf{x})\Big]^{1-\alpha} \\ &= \alpha G(\boldsymbol{\theta}_1) + (1-\alpha) G(\boldsymbol{\theta}_2), \end{aligned}$$

using Hölder's inequality.

The function $G(\cdot)$ plays a fundamental role in characterizing members of an exponential family [57]. Let $E_{\boldsymbol{\theta}}[\cdot]$ be the expectation with respect to the

distribution $p(\cdot|\boldsymbol{\theta})$. First, note that the moment-generating function of $p(\cdot|\boldsymbol{\theta})$ can be written as:

$$\begin{aligned} E_{\boldsymbol{\theta}}\left[\mathrm{e}^{\mathbf{t}\cdot\mathbf{x}}\right] &= \int_{\mathcal{X}} \mathrm{e}^{\mathbf{t}\cdot\mathbf{x}} \mathrm{e}^{\boldsymbol{\theta}\cdot\mathbf{x}-G(\boldsymbol{\theta})} h(\mathbf{x})\nu(\mathrm{d}\mathbf{x}) \\ &= \mathrm{e}^{-G(\boldsymbol{\theta})} \int_{\mathcal{X}} \mathrm{e}^{(\mathbf{t}+\boldsymbol{\theta})\cdot\mathbf{x}} h(\mathbf{x})\nu(\mathrm{d}\mathbf{x}) \\ &= \mathrm{e}^{-G(\boldsymbol{\theta})} \mathrm{e}^{G(\mathbf{t}+\boldsymbol{\theta})} = \mathrm{e}^{G(\mathbf{t}+\boldsymbol{\theta})-G(\boldsymbol{\theta})}, \end{aligned}$$

so that the function $G(\cdot)$ fully characterizes the moment-generating function of $p(\cdot|\boldsymbol{\theta})$.

Second, the function $G(\cdot)$ is differentiable and its gradient function $g(\cdot)$ is equal to the mean of $\mathbf{x}$, i.e., $E_{\boldsymbol{\theta}}[\mathbf{x}] = \nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta})$ [76]:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \log \int_{\mathcal{X}} \mathrm{e}^{\boldsymbol{\theta}\cdot\mathbf{x}} h(\mathbf{x})\nu(\mathrm{d}\mathbf{x}) \\ &= \frac{\int_{\mathcal{X}} \nabla_{\boldsymbol{\theta}} \mathrm{e}^{\boldsymbol{\theta}\cdot\mathbf{x}} h(\mathbf{x})\nu(\mathrm{d}\mathbf{x})}{\int_{\mathcal{X}} \mathrm{e}^{\boldsymbol{\theta}\cdot\mathbf{x}} h(\mathbf{x})\nu(\mathrm{d}\mathbf{x})} = \frac{\int_{\mathcal{X}} \mathbf{x}\mathrm{e}^{\boldsymbol{\theta}\cdot\mathbf{x}} h(\mathbf{x})\nu(\mathrm{d}\mathbf{x})}{\int_{\mathcal{X}} \mathrm{e}^{\boldsymbol{\theta}\cdot\mathbf{x}} h(\mathbf{x})\nu(\mathrm{d}\mathbf{x})} \\ &= \int_{\mathcal{X}} \mathbf{x}\mathrm{e}^{\boldsymbol{\theta}\cdot\mathbf{x}} h(\mathbf{x})\nu(\mathrm{d}\mathbf{x}) \cdot \mathrm{e}^{-G(\boldsymbol{\theta})} = \int_{\mathcal{X}} \mathbf{x}\mathrm{e}^{\boldsymbol{\theta}\cdot\mathbf{x}-G(\boldsymbol{\theta})} h(\mathbf{x})\nu(\mathrm{d}\mathbf{x}) \\ &= \int_{\mathcal{X}} \mathbf{x} p(\mathbf{x}|\boldsymbol{\theta})\nu(\mathrm{d}\mathbf{x}) = E_{\boldsymbol{\theta}}[\mathbf{x}]. \end{aligned}$$

The derivatives of the moment-generating function evaluated at 0 produce the moments, hence the previous result can also be obtained as follows:

$$\begin{aligned} E_{\boldsymbol{\theta}}[\mathbf{x}] &= \left.\frac{\partial E_{\boldsymbol{\theta}}\left[\mathrm{e}^{\mathbf{t}\cdot\mathbf{x}}\right]}{\partial \mathbf{t}}\right|_{\mathbf{t}=\mathbf{0}} = \left.\frac{\partial}{\partial \mathbf{t}} \mathrm{e}^{G(\mathbf{t}+\boldsymbol{\theta})-G(\boldsymbol{\theta})}\right|_{\mathbf{t}=\mathbf{0}} \\ &= \left.\frac{\partial G(\mathbf{t}+\boldsymbol{\theta})}{\partial \mathbf{t}} \mathrm{e}^{G(\mathbf{t}+\boldsymbol{\theta})-G(\boldsymbol{\theta})}\right|_{\mathbf{t}=\mathbf{0}} = \nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta}). \end{aligned}$$

Similarly, it can be shown that its Hessian $\nabla^2_{\boldsymbol{\theta}} G(\boldsymbol{\theta})$ is the covariance matrix of $\mathbf{x}$.

Since $G(\boldsymbol{\theta})$ is convex, its Hessian is (symmetric and) positive semidefinite, and strictly positive definite if the family is minimal.

A.2.3 Two parameter spaces

For minimal standard exponential families, the cumulant generative function $G(\cdot)$ is strictly convex as discussed earlier. Therefore, for any $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathcal{N}$, its

directional derivative function $\nabla G(\cdot) \cdot (\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1)/\|\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1\|$ strictly increases on the line connecting $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, i.e., $g(\boldsymbol{\theta}_1) = \nabla G(\boldsymbol{\theta}_1) \neq \nabla G(\boldsymbol{\theta}_2) = g(\boldsymbol{\theta}_2)$. Hence, the function $g(\cdot)$ is bijective and invertible.

Let $\boldsymbol{\mu} = g(\boldsymbol{\theta})$ where $\boldsymbol{\mu}$ is called the *expectation parameter*. Sometimes it is more convenient to parameterize a distribution in the exponential family by using its expectation parameter $\boldsymbol{\mu}$ instead of its natural parameter $\boldsymbol{\theta}$. This pair of parameterizations has a dual relationship. To understand this duality, the key is that convex functions come in pairs [84, 85]: the (convex) *conjugate* of the convex function $G : \mathbb{R}^d \to \mathbb{R}$ is defined as $G^* : \mathbb{R}^d \to \mathbb{R}$ and given by

$$G^*(\boldsymbol{\eta}) = \sup_{\boldsymbol{\theta} \in \mathbf{dom}\, G} \big(\boldsymbol{\theta} \cdot \boldsymbol{\eta} - G(\boldsymbol{\theta})\big).$$

The domain of the conjugate function consists of $\boldsymbol{\eta} \in \mathbb{R}^d$ for which the supremum is finite, i.e., for which the difference $\boldsymbol{\theta} \cdot \boldsymbol{\eta} - G(\boldsymbol{\theta})$ is bounded above on the domain of the function G. Clearly, G^* is a convex function, since it is the pointwise supremum of a family of convex (indeed, affine) functions of $\boldsymbol{\eta}$. It can be shown that if G is convex and closed, i.e., its epigraph is a closed set, then $G^{**} = G$. Moreover, the conjugate of a differentiable function G is also called the Legendre transform of G. To distinguish the general definition from the differentiable case, the term *Fenchel conjugate* is sometimes used instead of conjugate. Additionally, it is easily shown that for G convex and differentiable with $\mathbf{dom}\, G = \mathbb{R}^d$, $G^*(\boldsymbol{\eta})$ at the point $\boldsymbol{\eta} = \nabla G(\boldsymbol{\theta})$, or equivalently at $\boldsymbol{\theta} = (\nabla G)^{-1}(\boldsymbol{\eta})$, is given by

$$G^*(\boldsymbol{\eta}) = \sup_{\boldsymbol{\theta} \in \mathbf{dom}\, G} \big(\boldsymbol{\theta} \cdot \boldsymbol{\eta} - G(\boldsymbol{\theta})\big) = (\nabla G)^{-1}(\boldsymbol{\eta}) \cdot \boldsymbol{\eta} - G\big((\nabla G)^{-1}(\boldsymbol{\eta})\big).$$

Then, applying this definition to a standard exponential family $\{p(\cdot|\boldsymbol{\theta}) : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, the Fenchel conjugate $F(\cdot)$ of the cumulant generating function $G(\cdot)$ satifies the following:

$$F(\boldsymbol{\mu}) = G^*(\boldsymbol{\mu}) = \boldsymbol{\theta} \cdot \boldsymbol{\mu} - G(\boldsymbol{\theta}) \text{ for } \boldsymbol{\mu} = \nabla G(\boldsymbol{\theta}). \tag{A.2}$$

Let $f(\boldsymbol{\mu}) \triangleq \nabla_{\boldsymbol{\mu}} F(\boldsymbol{\mu})$ denote the gradient of $F(\boldsymbol{\mu})$:

$$\begin{aligned} f(\boldsymbol{\mu}) = \nabla_{\boldsymbol{\mu}} F(\boldsymbol{\mu}) &= \nabla_{\boldsymbol{\mu}} \{\boldsymbol{\theta} \cdot \boldsymbol{\mu} - G(\boldsymbol{\theta})\} \text{ for } \boldsymbol{\mu} = \nabla G(\boldsymbol{\theta}) \\ &= \nabla_{\boldsymbol{\mu}} \boldsymbol{\theta} \cdot \boldsymbol{\mu} + \boldsymbol{\theta} \cdot \nabla_{\boldsymbol{\mu}} \boldsymbol{\mu} - \nabla_{\boldsymbol{\mu}} G(\boldsymbol{\theta}) \\ &= \nabla_{\boldsymbol{\mu}} \boldsymbol{\theta} \cdot \boldsymbol{\mu} + \boldsymbol{\theta} - \nabla_{\boldsymbol{\mu}} \boldsymbol{\theta} \cdot g(\boldsymbol{\theta}) \\ &= \nabla_{\boldsymbol{\mu}} \boldsymbol{\theta} \cdot \boldsymbol{\mu} + \boldsymbol{\theta} - \nabla_{\boldsymbol{\mu}} \boldsymbol{\theta} \cdot \boldsymbol{\mu} = \boldsymbol{\theta}. \end{aligned}$$

Then, the two parameterizations $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ are related by the following transformations:

$$\boldsymbol{\mu} = g(\boldsymbol{\theta}) \quad \text{and} \quad f(\boldsymbol{\mu}) = \boldsymbol{\theta}. \tag{A.3}$$

It follows from (A.3) that the Hessian of $F(\boldsymbol{\mu})$ is the inverse of the Fisher Information matrix, i.e., $\nabla^2_{\boldsymbol{\mu}} F(\boldsymbol{\mu}) = \big(\nabla^2_{\boldsymbol{\theta}} G(\boldsymbol{\theta})\big)^{-1}$. Using (A.3) to replace $\boldsymbol{\mu}$ in (A.2) brings:

$$F\big(g(\boldsymbol{\theta})\big) = \boldsymbol{\theta} \cdot g(\boldsymbol{\theta}) - G(\boldsymbol{\theta}). \tag{A.4}$$

To conclude discussion of the duality property, there exist two spaces, the expectation parameter space (or data space) and the natural parameter space, and two functions that enable switching from one space to the other, $g(\cdot)$ and $f(\cdot)$, as shown in Figure A.1. As shown above, $g(\cdot)$ is the inverse of $f(\cdot)$, $f(\boldsymbol{\mu}) = g^{-1}(\boldsymbol{\mu})$. The function $g(\cdot)$ is referred to as the link function. The Generalized Linear Models (GLMs) literature often denotes $f(\cdot)$ as the link function [3].

A.2.4 Log-likelihood, score function and information matrix

The likelihood function of a random sample $\big\{\mathbf{x}[1], \ldots, \mathbf{x}[n]\big\}$ independently and identically drawn from an exponential family distribution with unknown parameter $\boldsymbol{\theta}$ is given by

$$L(\boldsymbol{\theta}) = \prod_{k=1}^{n} p\big(\mathbf{x}[k]|\boldsymbol{\theta}\big)$$

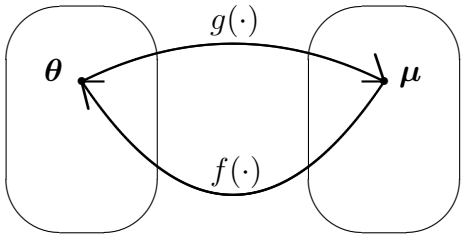

Figure A.1 Exponential family two parameter spaces, the link function $g(\cdot)$ and its inverse $f(\cdot)$.

and the log-likelihood function takes the form

$$\begin{aligned} l(\boldsymbol{\theta}) &= \log L(\boldsymbol{\theta}) \\ &= \sum_{k=1}^{n} \log p\big(\mathbf{x}[k]|\boldsymbol{\theta}\big) = \sum_{k=1}^{n} \big\{\boldsymbol{\theta} \cdot \mathbf{x}[k] - G(\boldsymbol{\theta})\big\} + \sum_{k=1}^{n} \log h\big(\mathbf{x}[k]\big). \end{aligned}$$

The vector of the first derivative of $l(\cdot)$ with respect to the row vector $\boldsymbol{\theta}$ is called the *score function*. It is defined as

$$s(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} l(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = \frac{1}{L(\boldsymbol{\theta})} \frac{\partial}{\partial \boldsymbol{\theta}} L(\boldsymbol{\theta}), \tag{A.5}$$

where $\nabla_{\boldsymbol{\theta}} l(\cdot)$ is the gradient of the log-likelihood function with respect to $\boldsymbol{\theta}$. Here, the following convention for derivatives with respect to a row vector is used: for the $(1 \times d)$ vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_d]$ and the scalar-valued function $l(\boldsymbol{\theta})$, the score function $s(\boldsymbol{\theta}) = \partial l(\boldsymbol{\theta})/\partial \boldsymbol{\theta} = \big[\partial l(\boldsymbol{\theta})/\partial \theta_1, \ldots, \partial l(\boldsymbol{\theta})/\partial \theta_d\big]^T$ is a $(d \times 1)$ vector. Then,

$$\begin{aligned} E\left[s(\boldsymbol{\theta})\right] &= E\left[\frac{\partial l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right] = E\left[\frac{\partial}{\partial \boldsymbol{\theta}}\Big\{\sum_{k=1}^{n} \big\{\boldsymbol{\theta} \cdot \mathbf{x}[k] - G(\boldsymbol{\theta})\big\} + \sum_{k=1}^{n} \log h\big(\mathbf{x}[k]\big)\Big\}\right] \\ &= \sum_{k=1}^{n} \left\{E\big[\mathbf{x}[k]\big] - \frac{\partial G(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right\} = \sum_{k=1}^{n} \{\boldsymbol{\mu} - g(\boldsymbol{\theta})\} = 0. \end{aligned}$$

The following matrix,

$$E\left[\frac{-\partial^2 l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right] = E\big[-\nabla^2_{\boldsymbol{\theta}} l(\boldsymbol{\theta})\big], \tag{A.6}$$

is called the *Fisher-information matrix*, where $\nabla^2_{\boldsymbol{\theta}} l(\cdot)$ is the Hessian matrix of the log-likelihood function with respect to $\boldsymbol{\theta}$ [7, 76]. It represents the information that the random sample $\{\mathbf{x}[1], \ldots, \mathbf{x}[n]\}$ contains about the parameter $\boldsymbol{\theta}$. More specifically, $E\big[-\nabla^2_{\boldsymbol{\theta}} l(\boldsymbol{\theta})\big]$ is a $(d \times d)$ matrix and can be expressed as

$$\begin{aligned} E\Big[-\nabla^2_{\boldsymbol{\theta}} l(\boldsymbol{\theta})\Big] &= E\left[-\frac{\partial^2}{\partial \boldsymbol{\theta}^2}\Big\{\sum_{k=1}^{n}\big\{\boldsymbol{\theta}\cdot\mathbf{x}[k] - G(\boldsymbol{\theta})\big\} + \sum_{k=1}^{n}\log h\big(\mathbf{x}[k]\big)\Big\}\right] \\ &= E\Big[\sum_{k=1}^{n}\nabla^2_{\boldsymbol{\theta}} G(\boldsymbol{\theta})\Big] = n\nabla^2_{\boldsymbol{\theta}} G(\boldsymbol{\theta}), \end{aligned}$$

where $\nabla^2_{\boldsymbol{\theta}} G(\boldsymbol{\theta})$ is the covariance matrix of $\mathbf{x}$.

A.3 Bregman distance

A.3.1 Definition

Bregman distances were introduced by L. M. Bregman [102]. The Bregman distance between two points in $\mathbb{R}^d$ is a nonnegative real number which can be interpreted as a measure of distance [87]. However, a Bregman distance is usually not symmetric and the triangular inequality may not hold. Hence Bregman distances are also referred to as Bregman divergences. The Bregman distance is formally defined as follows: for an arbitrary real-valued convex and differentiable function $G(\boldsymbol{\theta})$ on the parameter space $\boldsymbol{\Theta} \subseteq \mathbb{R}^d$, the Bregman distance between two parameters $\boldsymbol{\theta}$ and $\widetilde{\boldsymbol{\theta}}$ in $\boldsymbol{\Theta}$ is defined as [57]

$$B_G(\widetilde{\boldsymbol{\theta}} \| \boldsymbol{\theta}) \triangleq G(\widetilde{\boldsymbol{\theta}}) - G(\boldsymbol{\theta}) - (\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}) \cdot \nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta}).$$

Here, $\nabla_{\boldsymbol{\theta}}$ denotes the gradient with respect to $\boldsymbol{\theta}$ and "$\cdot$" denotes the dot product between vectors. As discussed in [57], the Bregman distance $B_G(\widetilde{\boldsymbol{\theta}} \| \boldsymbol{\theta})$ equals $G(\widetilde{\boldsymbol{\theta}})$ minus the linearization of $G(\widetilde{\boldsymbol{\theta}})$ around $\boldsymbol{\theta}$. Equivalently, $B_G(\widetilde{\boldsymbol{\theta}} \| \boldsymbol{\theta})$ is the error in the approximation of $G(\widetilde{\boldsymbol{\theta}})$ by a first-order Taylor expansion around $\boldsymbol{\theta}$. Since $G(\boldsymbol{\theta})$ is convex, $B_G(\widetilde{\boldsymbol{\theta}} \| \boldsymbol{\theta}) \geq 0$, and, if $G(\boldsymbol{\theta})$ is strictly convex, with equality if and only if $\widetilde{\boldsymbol{\theta}} = \boldsymbol{\theta}$.

For a 1-dimensional parameter $\theta \in \Theta \subseteq \mathbb{R}$, the Bregman distance takes the following form:

$$B_G(\widetilde{\theta}\|\theta) = G(\widetilde{\theta}) - G(\theta) - (\widetilde{\theta} - \theta)g(\theta),$$

where $g(\theta) = G'(\theta)$. As previously described in [88], a graphical representation of the Bregman distance as a measure of the convexity of G is shown in Figure A.2.

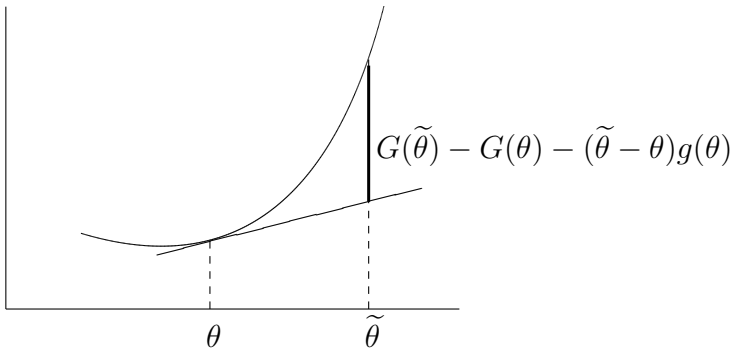

Figure A.2 For 1-dimensional parameters θ and $\widetilde{\theta}$, the Bregman distance is an indication of the increase in $G(\widetilde{\theta})$ over $G(\theta)$ above linear growth with slope $g(\theta)$.

The Bregman distance encompasses a large class of distance/divergence functions. For example, let the parameter space be $\boldsymbol{\Theta} = \mathbb{R}^d$ and the function $G(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta} \cdot \boldsymbol{\theta}$, which is convex and differentiable on $\boldsymbol{\Theta}$. In this case, the Bregman distance becomes the squared Euclidean distance:

$$B_G(\widetilde{\boldsymbol{\theta}}\|\boldsymbol{\theta}) = \frac{1}{2}\widetilde{\boldsymbol{\theta}} \cdot \widetilde{\boldsymbol{\theta}} - \frac{1}{2}\boldsymbol{\theta} \cdot \boldsymbol{\theta} - (\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}) \cdot \boldsymbol{\theta} = \frac{1}{2}\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}\|^2.$$

Another example is the Itakura-Saito distance often used in speech processing, where $G(\boldsymbol{\theta}) = -\sum_{i=1}^{d} \log(\boldsymbol{\theta}_i)$ on the parameter space $\boldsymbol{\Theta} = \mathbb{R}^d_{\geq 0}$. Noting that $G(\cdot)$ is convex and differentiable on $\boldsymbol{\Theta}$, the Bregman distance becomes

$$\begin{aligned} B_G(\widetilde{\boldsymbol{\theta}}\|\boldsymbol{\theta}) &= -\sum_{i=1}^{d} \log(\widetilde{\boldsymbol{\theta}}_i) + \sum_{i=1}^{d} \log(\boldsymbol{\theta}_i) + \sum_{i=1}^{d} \frac{1}{\boldsymbol{\theta}_i}(\widetilde{\boldsymbol{\theta}}_i - \boldsymbol{\theta}_i) \\ &= \sum_{i=1}^{d} \left(\frac{\widetilde{\boldsymbol{\theta}}_i}{\boldsymbol{\theta}_i} - \log \frac{\widetilde{\boldsymbol{\theta}}_i}{\boldsymbol{\theta}_i} - 1 \right). \end{aligned}$$

Table A.1 presents an example of different distances that in fact are a special case

of a Bregman distance (for the Mahalanobis distance, $\mathbf{W}$ is the inverse covariance matrix of $\boldsymbol{\theta}$).

Table A.1 Example of Bregman distances.

domain	$G(\boldsymbol{\theta})$	$B_G(\widetilde{\boldsymbol{\theta}}\|\boldsymbol{\theta})$	Bregman distance
$\mathbb{R}$	θ^2	$(\widetilde{\theta}-\theta)^2$	squared Euclidean dist.
$\mathbb{R}^d$	$\|\boldsymbol{\theta}\|^2$	$\|\widetilde{\boldsymbol{\theta}}-\boldsymbol{\theta}\|^2$	squared Euclidean dist.
$\mathbb{R}^d$	$\boldsymbol{\theta}\mathbf{W}\boldsymbol{\theta}^T$	$(\widetilde{\boldsymbol{\theta}}-\boldsymbol{\theta})\mathbf{W}(\widetilde{\boldsymbol{\theta}}-\boldsymbol{\theta})^T$	Mahalanobis distance
$\mathbb{R}^d_{\geq 0}$	$-\sum_{i=1}^d \log\theta_i$	$\sum_{i=1}^d \left\{\widetilde{\theta}_i/\theta_i - \log\widetilde{\theta}_i/\theta_i - 1\right\}$	Itakuro-Saito distance

Considering a standard exponential family $\{p(\cdot|\boldsymbol{\theta}) : \boldsymbol{\theta}\in\boldsymbol{\Theta}\}$, application of identity (A.4) implies that the negative log-likelihood function of the density $p(\mathbf{x}|\boldsymbol{\theta})$ can be expressed through a Bregman distance [57]:

$$\begin{aligned}
-\log p(\mathbf{x}|\boldsymbol{\theta}) &= -\log\left\{\mathrm{e}^{\boldsymbol{\theta}\cdot\mathbf{x}-G(\boldsymbol{\theta})}h(\mathbf{x})\right\}\\
&= -\boldsymbol{\theta}\cdot\mathbf{x} + G(\boldsymbol{\theta}) + \log h(\mathbf{x})\\
&= -\boldsymbol{\theta}\cdot\mathbf{x} - F\big(g(\boldsymbol{\theta})\big) + \boldsymbol{\theta}\cdot g(\boldsymbol{\theta}) + \log h(\mathbf{x})\\
&= \boldsymbol{\theta}\cdot\big(g(\boldsymbol{\theta})-\mathbf{x}\big) - F\big(g(\boldsymbol{\theta})\big) + \log h(\mathbf{x})\\
&= B_F\big(\mathbf{x}\big\|g(\boldsymbol{\theta})\big) - F(\mathbf{x}). \qquad \text{(A.7)}
\end{aligned}$$

Following the description of the Fenchel conjugate $F(\cdot)$ given in equation (A.2) and using the equivalence $\mathbf{x} = \nabla G(\boldsymbol{\theta}) = g(\boldsymbol{\theta}) \Leftrightarrow \boldsymbol{\theta} = f(\mathbf{x})$, $F(\mathbf{x})$ can be expressed as:

$$\begin{aligned}
F(\mathbf{x}) &= f(\mathbf{x})\cdot\mathbf{x} - G\big(f(\mathbf{x})\big)\\
&= f(\mathbf{x})\cdot\mathbf{x} - \int_{\mathcal{X}} \mathrm{e}^{f(\mathbf{x})\cdot\mathbf{x}}\nu(\mathrm{d}\mathbf{x}).
\end{aligned}$$

Bregman distances are a generalization of the log-likelihood of any member of the exponential family of distributions and the negative log-likelihood of the density of an exponential family can be written as the sum of a uniquely determined

Bregman distance and a term that is independent of the parameter $\boldsymbol{\theta}$. Hence, maximizing the log-likelihood function is equivalent to minimizing a Bregman distance. Though it was formally proven by [16], the existence of a unique Bregman distance corresponding to every regular exponential family had been previously observed [57, 87].

A.3.2 Properties of the Bregman distance

The following properties are true of the Bregman distance [16, 57].

1. **Non-negativity**: If $G(\boldsymbol{\theta})$ is strictly convex and $B_G(\widetilde{\boldsymbol{\theta}}\|\boldsymbol{\theta}) \geq 0$, then equality holds if and only if $\widetilde{\boldsymbol{\theta}} = \boldsymbol{\theta}$.

2. **Convexity**: $B_G(\cdot\|\cdot)$ is always convex in the first argument, but not necessarily in the second argument.

3. **Linearity**: The Bregman distance is a linear operator on the space of generating functions,

$$B_{G+H}(\widetilde{\boldsymbol{\theta}}\|\boldsymbol{\theta}) = B_G(\widetilde{\boldsymbol{\theta}}\|\boldsymbol{\theta}) + B_H(\widetilde{\boldsymbol{\theta}}\|\boldsymbol{\theta}) \text{ and}$$
$$B_{cG}(\widetilde{\boldsymbol{\theta}}\|\boldsymbol{\theta}) = cB_G(\widetilde{\boldsymbol{\theta}}\|\boldsymbol{\theta}) \text{ for } c \geq 0.$$

4. **Non-symmetry**: Bregman distances are usually not symmetric,

$$B_G(\widetilde{\boldsymbol{\theta}}\|\boldsymbol{\theta}) \neq B_G(\boldsymbol{\theta}\|\widetilde{\boldsymbol{\theta}}).$$

5. **Dual distances**: If $G(\boldsymbol{\theta})$ is strictly convex, the Bregman distances have the following duality property:

$$B_G(\widetilde{\boldsymbol{\theta}}\|\boldsymbol{\theta}) = B_F\big(g(\boldsymbol{\theta})\|g(\widetilde{\boldsymbol{\theta}})\big) = B_F(\boldsymbol{\mu}\|\widetilde{\boldsymbol{\mu}}).$$

6. **Generalized Pythagorean Theorem**: For any $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ and $\boldsymbol{\theta}_3$,

$$B_G(\boldsymbol{\theta}_1\|\boldsymbol{\theta}_2) + B_G(\boldsymbol{\theta}_2\|\boldsymbol{\theta}_3) = B_G(\boldsymbol{\theta}_1\|\boldsymbol{\theta}_3) + (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) \cdot (\boldsymbol{\mu}_3 - \boldsymbol{\mu}_2).$$

   The dot product can usually have any sign. When it is negative the above contradicts the triangular inequality.

As a conclusion, there are two ways in which Bregman distances are important. Firstly, they generalize squared Euclidean distance to a class of distances that all share similar properties. Secondly, they bear a strong connection to exponential families of distributions: there is a bijection between exponential families and Bregman distances. Recently, researchers have shown that many important algorithms can be generalized from Euclidean metrics to distances defined by a Bregman distance [10, 14–16, 40, 103, 104].

A.4 Kullback-Leibler divergence

The Kullback-Leibler divergence is also referred to as the Kullback-Leibler information [55, 76], or the Kullback-Leibler relative entropy [37].
The Kullback-Leibler divergence for distributions p and q over a set $\mathcal{X}$ is

$$\mathrm{D}(p||q) = \sum_{\mathbf{x}\in\mathcal{X}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})}, \text{ or } \int_{\mathcal{X}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x}, \text{ respectively.}$$

The Kullback-Leibler divergence $\mathrm{D}(p||q)$ can also be called the entropy distance between p and q with respect to p [55, 76], or the relative entropy of p with respect to q [105].

The Kullback-Leibler divergence can be seen as a measure of distance between distributions, but it is not a metric in the topological sense. In particular, it is generally not symmetric, i.e., $\mathrm{D}(p||q) \neq \mathrm{D}(q||p)$. Also, it is interesting to note that the Kullback-Leibler divergence is unbounded. For example, if $\mathcal{X} = \{0, 1\}$, $p(0) = q(1) = 1$, and $p(1) = q(0) = 0$, then $\mathrm{D}(p||q) = \infty$. In general, $p(\mathbf{x})$ should be zero whenever $q(\mathbf{x})$ is zero to avoid this, i.e., p should be absolutely continuous with respect to q, denoted $p \ll q$ (for all events A, if $q(A) = 0$, then $p(A) = 0$), and $p(\mathbf{x})/q(\mathbf{x})$ for $q(\mathbf{x}) = 0$ is then defined to equal 1. In light of the two previous properties of the Kullback-Leibler divergence, for cases where the choice between $\mathrm{D}(p||q)$ and $\mathrm{D}(q||p)$ is available, it is preferable to have the smoother distribution as the second argument. Moreover, $\mathrm{D}(p||q) \geq 0$ as a consequence of the Jensen's

inequality applied to the convex function $-\log(\cdot)$:

$$\begin{aligned}\mathrm{D}(p||q) &= \sum_{\mathbf{x}\in\mathcal{X}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} = E_p\left[\log \frac{p}{q}\right] = E_p\left[-\log \frac{q}{p}\right] \\ &\geq -\log E_p\left[\frac{q}{p}\right] = -\log \sum_{\mathbf{x}\in\mathcal{X}} p(\mathbf{x}) \frac{q(\mathbf{x})}{p(\mathbf{x})} = \log(1) = 0.\end{aligned}$$

Then, for any two distributions, $\mathrm{D}(p||q)$ exists and satisfies $0 \leq \mathrm{D}(p||q) \leq \infty$, where $\mathrm{D}(p||q) = 0$ if and only if $p = q$. Additionally,

$$\begin{aligned}\mathrm{D}\big(p(\mathbf{x})q(\mathbf{y})||r(\mathbf{x})s(\mathbf{y})\big) &= \sum_{\mathbf{x},\mathbf{y}\in\mathcal{X}} p(\mathbf{x})q(\mathbf{y}) \log \frac{p(\mathbf{x})q(\mathbf{y})}{r(\mathbf{x})s(\mathbf{y})} \\ &= \sum_{\mathbf{x},\mathbf{y}\in\mathcal{X}} p(\mathbf{x})q(\mathbf{y}) \log \frac{p(\mathbf{x})}{r(\mathbf{x})} + \sum_{\mathbf{x},\mathbf{y}\in\mathcal{X}} p(\mathbf{x})q(\mathbf{y}) \log \frac{q(\mathbf{y})}{s(\mathbf{y})} \\ &= \mathrm{D}\big(p(\mathbf{x})||r(\mathbf{x})\big) + \mathrm{D}\big(q(\mathbf{y})||s(\mathbf{y})\big).\end{aligned}$$

The Kullback-Leibler divergence between two Gaussian distributions $p_{\boldsymbol{\mu}}$ and $p_{\boldsymbol{\mu}'}$ with means $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$, respectively, and identity covariance matrices is:

$$\begin{aligned}\mathrm{D}(p_{\boldsymbol{\mu}}||p_{\boldsymbol{\mu}'}) &= \int p_{\boldsymbol{\mu}}(\mathbf{x}) \log \frac{p_{\boldsymbol{\mu}}(\mathbf{x})}{p_{\boldsymbol{\mu}'}(\mathbf{x})} \mathrm{d}\mathbf{x} = \int p_{\boldsymbol{\mu}}(\mathbf{x}) \log \frac{\mathrm{e}^{-\|\mathbf{x}-\boldsymbol{\mu}\|^2/2}}{\mathrm{e}^{-\|\mathbf{x}-\boldsymbol{\mu}'\|^2/2}} \mathrm{d}\mathbf{x} \\ &= \int p_{\boldsymbol{\mu}}(\mathbf{x}) \left\{-\frac{1}{2}\|\mathbf{x}-\boldsymbol{\mu}\|^2 + \frac{1}{2}\|\mathbf{x}-\boldsymbol{\mu}'\|^2\right\} \mathrm{d}\mathbf{x} \\ &= \int p_{\boldsymbol{\mu}}(\mathbf{x}) \left\{\mathbf{x}\cdot\boldsymbol{\mu} - \mathbf{x}\cdot\boldsymbol{\mu}' - \frac{1}{2}\|\boldsymbol{\mu}\|^2 + \frac{1}{2}\|\boldsymbol{\mu}'\|^2\right\} \mathrm{d}\mathbf{x} \\ &= \frac{1}{2}\|\boldsymbol{\mu}\|^2 + \frac{1}{2}\|\boldsymbol{\mu}'\|^2 - \boldsymbol{\mu}\cdot\boldsymbol{\mu}' = \frac{1}{2}\|\boldsymbol{\mu}-\boldsymbol{\mu}'\|^2.\end{aligned}$$

Hence, the l_2^2-norm can be seen as the natural distance associated with Gaussian distributions.

The Kullback-Leibler divergence between two distributions $p_{\boldsymbol{\theta}}$ and $p_{\boldsymbol{\theta}'}$ from any standard exponential family is:

$$\begin{aligned}\mathrm{D}(p_{\boldsymbol{\theta}}||p_{\boldsymbol{\theta}'}) &= \int_{\mathcal{X}} p(\mathbf{x}|\boldsymbol{\theta}) \log \frac{p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x}|\boldsymbol{\theta}')} \mathrm{d}\mathbf{x} \\ &= \int_{\mathcal{X}} p(\mathbf{x}|\boldsymbol{\theta}) \log \frac{\mathrm{e}^{\boldsymbol{\theta}\cdot\mathbf{x}-G(\boldsymbol{\theta})}}{\mathrm{e}^{\boldsymbol{\theta}'\cdot\mathbf{x}-G(\boldsymbol{\theta}')}} \mathrm{d}\mathbf{x} \\ &= \int_{\mathcal{X}} p(\mathbf{x}|\boldsymbol{\theta}) \big\{(\boldsymbol{\theta}-\boldsymbol{\theta}')\cdot\mathbf{x} - G(\boldsymbol{\theta}) + G(\boldsymbol{\theta}')\big\} \mathrm{d}\mathbf{x},\end{aligned}$$

and, using the fact that $E_{\boldsymbol{\theta}}[\mathbf{x}] = \nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta})$,

$$= G(\boldsymbol{\theta}') - G(\boldsymbol{\theta}) - (\boldsymbol{\theta}' - \boldsymbol{\theta}) \cdot \nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta}) = B_G(\boldsymbol{\theta}' \| \boldsymbol{\theta}). \tag{A.8}$$

Hence, the Bregman distance is generated by the Kullback-Leibler divergence between distributions of an exponential family. The Bregman distance can then be seen as the natural measure of distance for a particular exponential family.
Note also that for distributions of a standard exponential family,

$$\log \frac{p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x}|\boldsymbol{\theta}')} = (\boldsymbol{\theta} - \boldsymbol{\theta}') \cdot \mathbf{x} - G(\boldsymbol{\theta}) + G(\boldsymbol{\theta}'). \tag{A.9}$$

Using (A.9) in (A.8) gives:

$$\mathrm{D}(p_{\boldsymbol{\theta}} || p_{\boldsymbol{\theta}'}) = B_G(\boldsymbol{\theta}' \| \boldsymbol{\theta}) = \log \frac{p\big(g(\boldsymbol{\theta})|\boldsymbol{\theta}\big)}{p\big(g(\boldsymbol{\theta})|\boldsymbol{\theta}'\big)}. \tag{A.10}$$

The second part of equation (A.10) shows that the Kullback-Leibler divergence is related to log-likelihood ratio testing. Finally, from equation (A.7) it follows that:

$$\mathrm{D}(p_{\boldsymbol{\theta}} || p_{\boldsymbol{\theta}'}) = B_G(\boldsymbol{\theta}' \| \boldsymbol{\theta}) = B_F\big(g(\boldsymbol{\theta}) \big\| g(\boldsymbol{\theta}')\big), \tag{A.11}$$

where F is the Fenchel conjugate of the cumulant generating function G.

The Kullback-Leibler divergence seems to arise as a natural measure of distance, or a natural measure of similarity, between probability distributions associated with language models [106–109]. For example, how should English words be clustered [106]? Let $\mathcal{M}$ be a set of common nouns and choose a set $\mathcal{V}$ comprised of some common verbs which co-occur with the nouns in $\mathcal{M}$. Then, for each noun $m \in \mathcal{M}$, the probability distribution p_m is defined as follows for $v \in \mathcal{V}$:

$$p_m(v) = \frac{\#\text{times } (m, v) \text{ co-occur}}{\sum_{v' \in \mathcal{V}} \#\text{times } (m, v') \text{ co-occur}},$$

and is measured over some corpus of documents (this assumes the noun m occurs with at least one verb of $\mathcal{V}$). Given these distributions, the goal is to find a partition into a specified number of clusters of these distributions and their associated means. The claim is that for any cluster C of distributions, its mean μ_C minimizes the

Kullback-Leibler divergence among all distributions μ in C, i.e., for any positive distribution μ on $\mathcal{V}$ (in particular, any weighted average of the distributions of C), $\sum_{p\in C} \mathrm{D}(p||\mu) \geq \sum_{p\in C} \mathrm{D}(p||\mu_C)$ (note that μ_C is the mean of the distributions of the cluster C and as such is a distribution itself):

$$\begin{aligned}
\sum_{p\in C} \mathrm{D}(p||\mu) &= \sum_{p\in C} \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\mu(\mathbf{x})} \\
&= \sum_{p\in C} \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\mu_C(\mathbf{x})} \frac{\mu_C(\mathbf{x})}{\mu(\mathbf{x})} \\
&= \sum_{p\in C} \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\mu_C(\mathbf{x})} + \sum_{p\in C} \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{\mu_C(\mathbf{x})}{\mu(\mathbf{x})} \\
&= \sum_{p\in C} \mathrm{D}(p||\mu_C) + \sum_{\mathbf{x}} \underline{\sum_{p\in C} p(\mathbf{x})} \log \frac{\mu_C(\mathbf{x})}{\mu(\mathbf{x})},
\end{aligned}$$

where the underlined term equals $|C|\mu_C(\mathbf{x})$ with $|C|$ the cardinality of the cluster,

$$= \sum_{p\in C} \mathrm{D}(p||\mu_C) + |C|\mathrm{D}(\mu||\mu_C) \geq \sum_{p\in C} \mathrm{D}(p||\mu_C).$$

The Kullback-Leibler divergence reaches its minimum value at the mean of C as does the l_2^2-norm in the classical K-means technique and hence offers a natural measure of similarity between probability distributions for clustering in language modeling applications.

A.5 Examples

The Gaussian, Inverse Gaussian, Exponential, Gamma, Chi-square, Beta, Dirichlet, Weibull, Bernoulli, Binomial, Multinomial, Poisson, Negative Binomial, Geometric distributions are all exponential families. The Cauchy and Uniform families of distributions are not exponential families. Note that the Gamma distribution with parameters b and $c = 1$, as well as the Weibull distribution with parameters b and $c = 1$ correspond to the Exponential distribution with mean b.

This paragraph presents a few useful examples of 1-dimensional exponential families [80] and their associated Bregman distances.

Following Definition 2, the probability density function of a 1-dimensional exponential family is $p(x|\theta) = \exp\{\theta x - G(\theta)\}\, h(x)$ where:

1. the factor $h(x)$ only depends on the data space variable $x \in \mathcal{X}$;

2. the cumulant generating function $G(\theta) = \log \int_{\mathcal{X}} \exp\{\theta x\} h(x) \nu(\mathrm{d}x)$ only depends on the natural parameter θ in $\mathcal{N} = \left\{\theta : \int_{\mathcal{X}} \exp\{\theta x\} h(x) \nu(\mathrm{d}x) < \infty\right\}$, with the Lebesgue measure ν for continuous exponential family probability densities, and the counting measure for discrete exponential family probability densities;

3. the link function $g(\theta)$ is the first derivative of $G(\theta)$; the function $f(\mu)$ is the inverse link function for $\mu = g(\theta)$ and the first derivative of $F(\mu)$;

4. $B_F\big(x \| g(\theta)\big) = F(x) - F\big(g(\theta)\big) - \big(x - g(\theta)\big) f\big(g(\theta)\big)$ denotes the Bregman distance associated with the exponential family of interest.

Below are presented several examples of 1-dimensional exponential family distributions and their various terms corresponding to Definition 2.

(i) Gaussian with unit-variance $\mathcal{N}(\mu, 1)$:

A Gaussian random variable is one of the most important random variables because it is a very good approximation of the sum of a large number of random variables whose distribution is not completely known. Its domain is $\mathcal{X} = \mathbb{R}$. The parameter μ is its average value and $\sigma > 0$ (chosen equal to 1 in this example) denotes the spread and σ^2 its variance. A Gaussian random variable has the following probability density function:

$$\begin{aligned} p(x|\mu) &= \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2}\right\} \\ &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \exp\left(x\mu - \frac{\mu^2}{2}\right), \quad \text{for } x \in \mathbb{R} \text{ and } \mu \in \mathbb{R}. \end{aligned}$$

Identifying the various terms in the definition of the probability density of a

standard exponential family yields:

$$\begin{aligned}
\mathcal{X} &= \mathbb{R}, \\
h(x) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right), \\
\theta &= \mu, \\
G(\theta) &= \log \int_{\mathcal{X}} \exp(\theta x) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \mathrm{d}x \\
&= \log \frac{1}{\sqrt{2\pi}} \int_{\mathcal{X}} \exp\left\{-\frac{(x-\theta)^2}{2}\right\} \exp\left(\frac{\theta^2}{2}\right) \mathrm{d}x \\
&= \log \exp\left(\frac{\theta^2}{2}\right) = \frac{\theta^2}{2}, \\
\mathcal{N} &= \mathbb{R}, \\
g(\theta) &= \theta, \\
f(\mu) &= \mu, \\
F(\mu) &= \frac{\mu^2}{2}, \\
B_F\big(x \big\| g(\theta)\big) &= F(x) - F\big(g(\theta)\big) - \big(x - g(\theta)\big) f\big(g(\theta)\big) \\
&= \frac{x^2}{2} - \frac{g(\theta)^2}{2} - \big(x - g(\theta)\big)\theta = \frac{x^2}{2} - \frac{\theta^2}{2} - (x-\theta)\theta \\
&= \frac{1}{2}(x-\theta)^2.
\end{aligned}$$

The natural parameter coincides with the expectation parameter.

(ii) Exponential(β):

An Exponential random variable measures the time between two successive arrivals in a Poisson process. It only takes positive values, $\mathcal{X} = \mathbb{R}_{\geq 0}$, and is parameterized by a positive real number $\beta > 0$. Its probability density function is:

$$p(x|\beta) = \beta \exp(-\beta x) = \exp(-\beta x + \log \beta), \quad \text{for } x \in \mathbb{R}_{\geq 0} \text{ and } \beta \in \mathbb{R}_{\geq 0}.$$

Identifying the various terms in the definition of the probability density of a

standard exponential family yields:

$$\begin{aligned}
\mathcal{X} &= \mathbb{R}_{\geq 0}, \\
h(x) &= 1, \\
\theta &= -\beta < 0, \\
G(\theta) &= \log \int_{\mathcal{X}} \exp(\theta x) \mathrm{d}x \\
&= \log \int_0^{+\infty} \exp(\theta x) \mathrm{d}x = \log \frac{-1}{\theta} = -\log(-\theta), \\
\mathcal{N} &= \mathbb{R}_-, \\
g(\theta) &= -\frac{1}{\theta}, \\
f(\mu) &= -\frac{1}{\mu}, \\
F(\mu) &= -\log \mu - 1, \\
B_F\big(x \big\| g(\theta)\big) &= F(x) - F\big(g(\theta)\big) - \big(x - g(\theta)\big) f\big(g(\theta)\big) \\
&= -\log x - 1 + \log g(\theta) + 1 - \big(x - g(\theta)\big)\theta \\
&= -\log \frac{x}{g(\theta)} - \big(x - g(\theta)\big)\theta = -\log(-x\theta) - x\theta - 1.
\end{aligned}$$

(iii) Bernoulli(p):

The Bernoulli random variable may be the simplest random variable. It takes only two values, $\mathcal{X} = \{0, 1\}$. It may be used to characterize the probability that a (possibly biased) coin falls on heads, taking the value 1, or tails, taking the value 0. It is sometimes interpreted as an experiment or trial with two possible outcomes: failure, wherein the random variable takes the value 0, or success, wherein the random variable takes the value 1, with probabilities $1 - p$ and p, respectively. A Bernoulli random variable has the following probability mass function:

$$\begin{aligned}
p(x|p) &= p^x (1-p)^{(1-x)} \\
&= \exp\left(x \log \frac{p}{1-p} + \log(1-p) \right), \quad \text{for } x \in \mathcal{X} = \{0, 1\},\ 0 \leq p \leq 1.
\end{aligned}$$

Identifying the various terms in the definition of the probability density of a standard exponential family yields:

$$\begin{aligned}
\mathcal{X} &= \{0,1\},\\
h(x) &= 1,\\
\theta &= \log\frac{p}{1-p},\\
G(\theta) &= \log\int_{\mathcal{X}}\exp(\theta x)\mathrm{d}x\\
&= \log\sum_{x=0}^{x=1}\exp(\theta x) = \log\big(1+\exp(\theta)\big)\\
\mathcal{N} &= \mathbb{R},\\
g(\theta) &= \frac{\exp(\theta)}{1+\exp(\theta)},\\
f(\mu) &= \log\frac{\mu}{1-\mu},\\
F(\mu) &= \mu\log\mu + (1-\mu)\log(1-\mu),\\
B_F\big(x\big\|g(\theta)\big) &= F(x) - F\big(g(\theta)\big) - \big(x-g(\theta)\big)f\big(g(\theta)\big)\\
&= x\log x + (1-x)\log(1-x) - g(\theta)\log g(\theta)\\
&\quad + \big(1-g(\theta)\big)\log\big(1-g(\theta)\big) - \big(x-g(\theta)\big)\theta\\
&= x\log\frac{x}{g(\theta)} + (1-x)\log\frac{1-x}{1-g(\theta)}\\
&= \log\big(1+\exp\{-(2x-1)\theta\}\big).
\end{aligned}$$

(iv) Binomial(p, N):

The Binomial random variable counts the number of successes in N independent Bernoulli experiments with parameter p. If the ordering was maintained, then the required probability for x successes would be $p^x(1-p)^{(N-x)}$. Since the ordering is disregarded, all possible outcomes leading to x successes must be counted. The probability mass function is therefore:

$$p(x|p) = \frac{N!}{x!(N-x)!}p^x(1-p)^{(N-x)}, \quad \text{for } x\in\mathcal{X}=\{0,1,2,\ldots,N\},\ 0\le p\le 1.$$

Identifying the various terms in the definition of the probability density of a standard exponential family yields:

$$
\begin{aligned}
\mathcal{X} &= \{0, 1, 2, \ldots, N\}, \\
h(x) &= \frac{N!}{x!(N-x)!}, \\
\theta &= \log \frac{p}{1-p}, \\
G(\theta) &= \log \int_{\mathcal{X}} \frac{N!}{x!(N-x)!} \exp(\theta x) \mathrm{d}x \\
&= \log \sum_{x=0}^{x=N} \frac{N!}{x!(N-x)!} \exp(\theta x) \\
&= \log \left(1^N + \frac{N!}{1!(N-1)!} 1^{N-1} \mathrm{e}^{\theta} + \cdots + \mathrm{e}^{N\theta} \right) \\
&= \log \big(1 + \exp(\theta)\big)^N = N \log \big(1 + \exp(\theta)\big), \\
\mathcal{N} &= \mathbb{R}, \\
g(\theta) &= N \frac{\exp(\theta)}{1 + \exp(\theta)}, \\
f(\mu) &= \log \frac{\mu}{N - \mu}, \\
F(\mu) &= \mu \log \frac{\mu}{N} + (N - \mu) \log \frac{N - \mu}{N}, \\
B_F\big(x \big\| g(\theta)\big) &= F(x) - F\big(g(\theta)\big) - \big(x - g(\theta)\big) f\big(g(\theta)\big) \\
&= x \log \frac{x}{g(\theta)} + (N - x) \log \frac{N - x}{N - g(\theta)} \\
&= N \log \frac{1 + \exp(\theta)}{\exp(\theta)} + (N - x)\theta \\
&\quad + x \log \frac{x}{N} + (N - x) \log \frac{N - x}{N}.
\end{aligned}
$$

(v) Poisson(λ):

The Poisson random variable is often used to count the number of occurrences of an event at a particular region in a fixed time period (e.g., cars arriving at an intersection or phone calls arriving at a switchboard). The parameter

λ is the average number of events occurring. Its probability mass function is

$$\begin{aligned} p(x|\lambda) &= \frac{\lambda^x \exp(-\lambda)}{x!} \\ &= \frac{\exp(x \log \lambda - \lambda)}{x!}, \quad \text{for } x \in \mathcal{X} = \{0, 1, 2, \dots\} \text{ and } \lambda > 0. \end{aligned}$$

Identifying the various terms in the definition of the probability density of a standard exponential family yields:

$$\begin{aligned} \mathcal{X} &= \{0, 1, 2, \dots\}, \\ h(x) &= \frac{1}{x!}, \\ \theta &= \log \lambda, \\ G(\theta) &= \log \int_{\mathcal{X}} \frac{1}{x!} \exp(\theta x) \mathrm{d}x = \log \sum_{x=0}^{\infty} \frac{1}{x!} \exp(\theta x) = \exp(\theta), \\ \mathcal{N} &= \mathbb{R}, \\ g(\theta) &= \exp(\theta), \\ f(\mu) &= \log \mu, \\ F(\mu) &= \mu \log \mu - \mu, \\ B_F\big(x \big\| g(\theta)\big) &= F(x) - F\big(g(\theta)\big) - \big(x - g(\theta)\big) f\big(g(\theta)\big) \\ &= x \log x - x - g(\theta) \log g(\theta) + g(\theta) - \big(x - g(\theta)\big)\theta \\ &= x \log \frac{x}{g(\theta)} - x - g(\theta) = x \log x - x\theta + \exp(\theta) - x. \end{aligned}$$

Table A.2, Table A.3 and Table A.4 summarize the characteristics of several continuous 1-dimensional exponential families whereas Table A.5 and Table A.6 present the characteristics of several discrete 1-dimensional exponential families.

Table A.2 Characteristics of several continuous 1-dimensional exponential families: Gaussian and Exponential.

	Gaussian $\mathcal{N}(\mu, 1)$	Gaussian $\mathcal{N}(\mu, \sigma^2)$	Exponential, $\beta > 0$
$\mathcal{X}$	$\mathbb{R}$	$\mathbb{R}$	$\mathbb{R}_{\geq 0}$
$P(x;\theta)$	$\frac{1}{\sqrt{2\pi}} \exp\{-\frac{(x-\mu)^2}{2}\}$	$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$	$\beta \mathrm{e}^{-\beta x}$
mean	μ	μ	$1/\beta$
variance	1	σ^2	$1/\beta^2$
θ	μ	μ/σ^2	$-\beta$
$G(\theta)$	$\theta^2/2$	$\theta^2\sigma^2/2$	$-\log(-\theta)$
$g(\theta) = E[x\|\theta]$	$\theta = \mu$	$\sigma^2\theta = \mu$	$-1/\theta = 1/\beta$
$g'(\theta) = \frac{\mathrm{d}g(\theta)}{\mathrm{d}\theta}$	1	σ^2	$1/\theta^2$
$f(x) = g^{-1}(x)$	x	x/σ^2	$-1/x$
$F(x)$	$x^2/2$	$x^2/(2\sigma^2)$	$-\log(x) - 1$
$B_F(\varphi \Vert \psi)$	$(\varphi - \psi)^2/2$	$(\varphi - \psi)^2/(2\sigma^2)$	$-\log(\frac{\varphi}{\psi}) + \frac{\varphi}{\psi} - 1$
$B_F(x \Vert g(\theta))$	$(x - \theta)^2/2$	$(x - \sigma^2\theta)^2/(2\sigma^2)$	$-\log(-x\theta) - x\theta - 1$

Table A.3 Characteristics of several continuous 1-dimensional exponential families: Chi-square and Inverse Gaussian.

	Chi-square, ν i.i.d. $\mathcal{N}(0, \sigma^2)$	Inv. Gaussian, $\mu > 0$, $\beta > 0$ fixed
$\mathcal{X}$	$\mathbb{R}_{\geq 0}$	$\mathbb{R}_{\geq 0}$
$P(x;\theta)$	$\frac{x^{\nu/2-1}}{2^{\nu/2}\sigma^\nu \Gamma(\frac{\nu}{2})} \exp(-\frac{x}{2\sigma^2})$	$\sqrt{\frac{\beta}{2\pi x^3}} \exp\{-\frac{\beta(x-\mu)^2}{2\mu^2 x}\}$
mean	ν	μ
variance	2ν	μ^3/β
θ	$-1/(2\sigma^2)$	$-\beta/(2\mu^2)$
$G(\theta)$	$(\nu/2)\log(-1/\theta)$	$-\beta/\mu$
$g(\theta) = E[x\|\theta]$	$-\nu/(2\theta) = \nu\sigma^2$	$\sqrt{\beta/(-2\theta)} = \mu$
$g'(\theta) = \frac{\mathrm{d}g(\theta)}{\mathrm{d}\theta}$	$\nu/(2\theta^2)$	$\sqrt{\beta/(-4\theta^3)}$
$f(x) = g^{-1}(x)$	$-\nu/(2x)$	$\beta/(2x^2)$
$F(x)$	$-\frac{\nu}{2}[\log(\frac{x}{\nu}) + 1]$	$\beta/(2x)$
$B_F(\varphi \Vert \psi)$	$-\frac{\nu}{2}[\log(\varphi/\psi) - \varphi/\psi + 1]$	$\beta[1/(2\varphi) - 1/\psi + \varphi/(2\psi^2)]$
$B_F(x \Vert g(\theta))$	$-\frac{\nu}{2}[\log(-2x\theta/\nu) - 2x\theta/\nu + 1]$	$\beta[1/(2x) - \sqrt{-2\theta/\beta} - x\theta/\beta]$

Table A.4 Characteristics of several continuous 1-dimensional exponential families: Gamma and Weibull.

	Gamma, $b > 0$, $c > 0$ fixed	Weibull, $b > 0$, $c > 0$ fixed
$\mathcal{X}$	$\mathbb{R}_{\geq 0}$	$\mathbb{R}_{\geq 0}$
$P(x;\theta)$	$(x/b)^{c-1}[\exp(-x/b)]/b\Gamma(c)$	$\frac{cx^{c-1}}{b}\exp\{-(\frac{x}{b})^c\}$
mean	bc	$b\Gamma[(c+1)/c]$
variance	b^2c	$b^2(\Gamma[(c+2)/c] - \{\Gamma[(c+1)/c]\}^2)$
θ	$-1/b < 0$	
$G(\theta)$	$\log[(-1/\theta)^c]$	
$g(\theta) = E[x\|\theta]$	$-c/\theta$	
$g'(\theta) = \frac{\mathrm{d}g(\theta)}{\mathrm{d}\theta}$	c/θ^2	
$f(x) = g^{-1}(x)$	$-c/x$	
$F(x)$	$-\log[(x/c)^c] - c$	
$B_F(\varphi\|\psi)$	$-\log[(\frac{\varphi}{\psi})^c] + c\frac{\varphi}{\psi} - c$	
$B_F(x\|g(\theta))$	$-\log[(-x\theta/c)^c] - x\theta - c$	

Table A.5 Characteristics of several discrete 1-dimensional exponential families: Bernoulli and Poisson.

	Bernoulli, $0 < p < 1$	Poisson, $\lambda > 0$
$\mathcal{X}$	{0,1}	$\{0, 1, 2, \dots\}$
$P(x;\theta)$	$p^x(1-p)^{(1-x)}$	$\frac{\lambda^x \mathrm{e}^{-\lambda}}{x!}$
mean	p	λ
variance	$p(1-p)$	$\lambda + \lambda^2$
θ	$\log\left(\frac{p}{1-p}\right)$	$\log\lambda$
$G(\theta)$	$\log(1+\mathrm{e}^\theta)$	e^θ
$g(\theta) = E[x\|\theta]$	$\frac{\mathrm{e}^\theta}{1+\mathrm{e}^\theta} = p$	$\mathrm{e}^\theta = \lambda$
$g'(\theta) = \frac{\mathrm{d}g(\theta)}{\mathrm{d}\theta}$	$\frac{\mathrm{e}^\theta}{(1+\mathrm{e}^\theta)^2}$	e^θ
$f(x) = g^{-1}(x)$	$\log\left(\frac{x}{1-x}\right)$	$\log(x)$
$F(x)$	$x\log(x) + (1-x)\log(1-x)$	$x\log(x) - x$
$B_F(\varphi\|\psi)$	$\varphi\log(\frac{\varphi}{\psi}) + (1-\varphi)\log\left(\frac{1-\varphi}{1-\psi}\right)$	$\varphi\log(\frac{\varphi}{\psi}) + \varphi - \psi$
$B_F(x\|g(\theta))$	$\log\left(1+\exp\{-(2x-1)\theta\}\right)$	$\mathrm{e}^\theta - x\theta + x\log(x) - x$

Table A.6 Characteristics of several discrete 1-dimensional exponential families: Binomial.

	Binomial, $0 < p < 1$ & N fixed
$\mathcal{X}$	$\{0, 1, 2, \ldots, N\}$
$P(x;\theta)$	$\frac{N!}{x!(N-x)!}p^x(1-p)^{N-x}$
mean	Np
variance	$Np(1-p)$
θ	$\log\left(\frac{p}{1-p}\right)$
$G(\theta)$	$N\log(1+\mathrm{e}^{\theta})$
$g(\theta) = E[x\|\theta]$	$N\frac{\mathrm{e}^{\theta}}{1+\mathrm{e}^{\theta}} = Np$
$g'(\theta)$	$N\frac{\mathrm{e}^{\theta}}{(1+\mathrm{e}^{\theta})^2}$
$f(x) = g^{-1}(x)$	$\log\left(\frac{x}{N-x}\right)$
$F(x)$	$x\log\left(\frac{x}{N}\right) + (N-x)\log\left(\frac{N-x}{N}\right)$
$B_F(\varphi\Vert\psi)$	$\varphi\log\left(\frac{\varphi}{\psi}\right) + (N-\varphi)\log\left(\frac{N-\varphi}{N-\psi}\right)$
$B_F(x\Vert g(\theta))$	$N\log\left(\frac{1+\mathrm{e}^{\theta}}{\mathrm{e}^{\theta}}\right) + (N-x)\theta + x\log\left(\frac{x}{N}\right) + (N-x)\log\left(\frac{N-x}{N}\right)$

B The Newton-Raphson minimization technique

One classical minimization technique is the Newton, or Newton-Raphson, method and is based on the Newton-Raphson method for finding the roots of nonlinear equations [6]. In this procedure, a candidate solution is found by applying the Newton-Raphson method to find a zero of the gradient of the loss function. If the loss function is (locally) convex, then the solution will be a (locally) unique minimum.

The Newton-Raphson method results from a Taylor series expansion around some given point. Taylor series expansions utilize the principle that there exists a relationship between the value of a mathematical function (with continuous derivatives over the relevant support) at a given point, x_0, and the function value at another (perhaps close) point, x_1, given by

$$\begin{aligned} f(x_1) = f(x_0) + (x_1 - x_0)f'(x_0) \\ + \frac{1}{2!}(x_1 - x_0)^2 f''(x_0) + \frac{1}{3!}(x_1 - x_0)^3 f'''(x_0) + \cdots , \end{aligned} \tag{B.1}$$

where $f'(\cdot)$ is the first derivative with respect to x, $f''(\cdot)$ is the second derivative with respect to x, and so on. Infinite precision is achieved only with infinite application of the series (as opposed to just the four terms previously provided), and is therefore unobtainable. For the purposes of most statistical estimation techniques, only the first two terms are required. Note that later terms will be unimportant because of the rapidly growing factorial function in the denominator.

The assumption is that the problem aims at finding the point x_1 such that $f(x_1) = 0$. This is a root of function $f(\cdot)$. The Taylor series expansion in equation (B.1) then becomes:

$$\begin{aligned} 0 = f(x_0) + (x_1 - x_0)f'(x_0) \\ + \frac{1}{2!}(x_1 - x_0)^2 f''(x_0) + \frac{1}{3!}(x_1 - x_0)^3 f'''(x_0) + \cdots . \end{aligned} \tag{B.2}$$

Considering only the two first terms in the series expansion (B.2), the Gauss-Newton method yields:

$$0 \cong f(x_0) + (x_1 - x_0)f'(x_0). \tag{B.3}$$

The Newton-Raphson method rearranges (B.3) at the $(t+1)^{\text{st}}$ step to produce:

$$x^{(t+1)} = x^{(t)} - \frac{f\big(x^{(t)}\big)}{f'\big(x^{(t)}\big)}, \tag{B.4}$$

so that progressively improved estimates are produced until $f\big(x^{(t+1)}\big)$ is sufficiently close to zero. It can be shown that this method converges rapidly to a solution provided that the selected starting point is reasonably close to the solution and $f(\cdot)$ certifies some constraints. These are in particular satisfied if $f(\cdot)$ is sufficiently smooth and convex.

When applied to finding roots in statistical settings, the Newton-Raphson method adapts (B.3) to find the root of the first derivative of the log-likelihood function $l(\cdot)$, also called the *score function* (cf. Appendix A).

First, for a single-component parameter θ, following equation (B.4), iterative estimates are produced by

$$\theta^{(t+1)} = \theta^{(t)} - \frac{\partial/\partial\theta\, l\big(\theta^{(t)}\big)}{\partial^2/\partial\theta^2\, l\big(\theta^{(t)}\big)}.$$

Then, generalizing to a multiple components parameter, the update equation becomes

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \Big(\nabla^2_{\boldsymbol{\theta}} l\big(\boldsymbol{\theta}^{(t)}\big)\Big)^{-1} \nabla_{\boldsymbol{\theta}} l\big(\boldsymbol{\theta}^{(t)}\big), \tag{B.5}$$

where $\nabla^2_{\boldsymbol{\theta}} l(\cdot)$ and $\nabla_{\boldsymbol{\theta}} l(\cdot)$ are the Hessian and the gradient, respectively, of the log-likelihood function $l(\cdot)$ with respect to the parameter vector $\boldsymbol{\theta}$. The update equation (B.5) can also be modified as follows:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \alpha^{(t+1)} \Big(\nabla^2_{\boldsymbol{\theta}} l\big(\boldsymbol{\theta}^{(t)}\big) \Big)^{-1} \nabla_{\boldsymbol{\theta}} l\big(\boldsymbol{\theta}^{(t)}\big), \tag{B.6}$$

where $\alpha^{(t+1)}$ is the step size chosen for the $(t+1)^{\text{st}}$ iteration. In a machine learning environment, the step size is often referred to as the learning rate [49, 79]. It enables an adjustment in the relative size of the change in the parameter vector $\boldsymbol{\theta}$. The learning rate is usually taken to be a constant, and can also be optimized by a line search that maximizes the loss function at each update.

Sometimes the Hessian matrix is difficult to calculate and is replaced by its expectation with regard to $\boldsymbol{\theta}$, i.e., $E_{\boldsymbol{\theta}} \Big[\nabla^2_{\boldsymbol{\theta}} l\big(\boldsymbol{\theta}^{(t)}\big) \Big]$. This modification is referred to as *Fisher scoring* [6]. For exponential family distributions and canonical link functions, the observed and expected Hessian matrices are identical [5–7, 76, 110]. At each step of the Newton-Raphson algorithm, the following equations must be solved:

$$\big(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\big) E_{\boldsymbol{\theta}} \Big[\nabla^2_{\boldsymbol{\theta}} l\big(\boldsymbol{\theta}^{(t)}\big) \Big] = -\nabla_{\boldsymbol{\theta}} l\big(\boldsymbol{\theta}^{(t)}\big). \tag{B.7}$$

It can be shown that these equations correspond to normal equations in a least squares environment [81]. Therefore, the problem of root finding reduces to a repeated weighted least squares application in which the inverse of the diagonal values of $E_{\boldsymbol{\theta}} \Big[\nabla^2_{\boldsymbol{\theta}} l\big(\boldsymbol{\theta}^{(t)}\big) \Big]$ are the appropriate weights. The weights being constantly updated, the overall strategy is called the iterative weighted least squares [6, 82], also known as iterative reweighted least squares [7, 15], or iteratively weighted least squares [5].

C Non-parametric mixture models

Non-parametric mixture models have become popular over the last twenty years. One reason is that they provide a natural framework for practitioners to deal with *unobserved population heterogeneity* [34]. This situation arises when, under standard conditions, a certain model is valid. However, because of variation of the parameters describing the model in the population, these assumptions are no longer met, though they are still true in subpopulations described by variations of the parameters. Since one has not observed which subpopulation each observed data point belongs to, one can treat the variable describing subpopulation membership only as a *missing* variable. The corresponding marginal model is a specific form of the non-parametric mixture model. First, Appendix C.1 presents the theory around non-parametric mixture models within the Generalized Linear Statistics (GLS) framework, including the Non-Parametric Maximum Likelihood (NPML) estimation technique used in Section 3 and Section 5. Then, Appendix C.2 develops the Expectation-Maximization (EM) algorithm for the NPML estimation technique with a special focus on exponential family distributions.

C.1 Theory of non-parametric mixture models

Mixture models express the presence of extra-population heterogeneity in the following way: the (non-conditional) probability density function $p(\mathbf{x})$ of the

observation variable $\mathbf{x}$ takes the following form:

$$p(\mathbf{x}) = \int p(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}, \tag{C.1}$$

where $\pi(\boldsymbol{\theta})$ is the probability density function of the parameter vector $\boldsymbol{\theta}$. Because a different value of $\boldsymbol{\theta}$ means that the observation variable belongs to a different subpopulation, the parameter $\boldsymbol{\theta}$ expresses the population heterogeneity. Here it is represented as a continuous quantity with a continuous distribution for the sake of generality. Given the observation matrix $\mathbf{X} = \big[\mathbf{x}[1]^T, \ldots, \mathbf{x}[n]^T\big]^T \in \mathbb{R}^{n\times d}$, composed of n independent and identically distributed statistical data samples, each assumed to be stochastically equivalent to the random row vector $\mathbf{x}$, $\mathbf{x}[k] = \big[x_1[k], \ldots, x_d[k]\big] \sim \mathbf{x}$, the data likelihood function is given by:

$$p(\mathbf{X}) = \prod_{k=1}^{n} p\big(\mathbf{x}[k]\big) = \prod_{k=1}^{n} \int p\big(\mathbf{x}[k]|\boldsymbol{\theta}\big)\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}. \tag{C.2}$$

For a specified exponential family density $p(\cdot|\cdot)$, maximum likelihood identification of model (C.1) corresponds to identifying the vector $\boldsymbol{\theta}$ and its density function $\pi(\boldsymbol{\theta})$. This difficult problem is usually attacked using approximation methods which correspond to replacing the integrals in (C.1) and (C.2) by sums [17, 20, 22, 25, 26, 34, 66]:

$$p(\mathbf{x}) = \sum_{l=1}^{m} p\big(\mathbf{x}|\underline{\boldsymbol{\theta}}[l]\big)\pi_l = \sum_{l=1}^{m}\prod_{i=1}^{d} p(x_i|\underline{\theta}_i[l])\pi_l, \tag{C.3}$$

$$p(\mathbf{X}) = \prod_{k=1}^{n}\sum_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l \tag{C.4}$$

$$= \prod_{k=1}^{n}\sum_{l=1}^{m}\prod_{i=1}^{d} p\big(x_i[k]|\underline{\theta}_i[l]\big)\pi_l \tag{C.5}$$

over a finite number of discrete support points, or "atoms", $\underline{\boldsymbol{\theta}}[l]$ (equivalently, $\underline{\mathbf{a}}[l]$) for $l = 1, \ldots, m$, $1 \le m \le n$, with point-mass probabilities

$$\pi_l \triangleq \pi\big(\boldsymbol{\theta} = \underline{\boldsymbol{\theta}}[l]\big).$$

The data likelihood (C.4) thus approximates the likelihood of a finite mixture of exponential family densities with unknown mixture proportions or point-mass

probability estimates π_l and unknown point-mass support points $\underline{\boldsymbol{\theta}}[l]$. In the mixture models literature, the point-mass probabilities π_l are called mixing proportions or weights, the densities $p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)$ are called the component densities of the mixture and equation (3.11) is referred to as the m-component finite mixture density [53]. As clearly described in [29], the proposed approximation is justified either as a Gaussian quadrature approximation to the integral in (C.1), in the case of a Gaussian assumption for the probability density function $\pi(\boldsymbol{\theta})$ [4, 5, 38], or by appealing to the fact that the Non-Parametric Maximum Likelihood (NPML) estimate [17, 53, 66] of the mixture density $\pi(\boldsymbol{\theta})$ yields a solution which takes a finite number of points of support [17, 20, 22, 25, 26, 66].

The model in equation (C.3) enables the variation of the parameter over a diversity of subpopulations to be captured. In this case, the population consists of various subpopulations with parameter $\underline{\boldsymbol{\theta}}[1], \underline{\boldsymbol{\theta}}[2], \ldots, \underline{\boldsymbol{\theta}}[m]$ where m denotes the number (possibly unknown) of subpopulations. This situation is called a *heterogeneous* case. The same type of density in each subpopulation l is assumed, but with a potentially different parameter: $p\big(\mathbf{x}|\underline{\boldsymbol{\theta}}[l]\big)$ is the density of subpopulation l, $l = 1, \ldots, m$. In contrast, a *homogeneous* case assumes the parameter to be part of a unique subpopulation.

Following the notations in [34], the *mixing distribution* is denoted by $\mathcal{Q} = \big(\underline{\boldsymbol{\theta}}[l], \pi_l, l = 1, \ldots, m\big) = \big\{\underline{\boldsymbol{\theta}}[l], \pi_l\big\}_{l=1}^{m}$ and encompasses the parameters $\underline{\boldsymbol{\theta}}[l], l = 1, \ldots, m$ and their associated weights or point-mass probabilities $\pi_l, l = 1, \ldots, m$. Estimation is conventionally performed by maximum likelihood, i.e., the Non-Parametric Maximum Likelihood estimator $\widehat{\mathcal{Q}} = \big\{\widehat{\underline{\boldsymbol{\theta}}}[l], \widehat{\pi}_l\big\}_{l=1}^{m}$ maximizes the log-likelihood function $L(\mathcal{Q}) = \sum_{k=1}^{n} \log \sum_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l$. In this formulation of the NPML estimation problem, the number m of points of support is considered fixed. However, in many applications, the value of m is unknown and needs to be inferred from the available data; in this case $\widehat{m}$ denotes its estimator.

Corollary 1 ([34]). *Suppose that $p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}\big)$, as a function of $\underline{\boldsymbol{\theta}}$, has a* unique *mode for all $\mathbf{x}[k]$, which lies in the interval $\big[\mathbf{x}_{min}, \mathbf{x}_{max}\big]$ where $\mathbf{x}_{min}$ and $\mathbf{x}_{max}$ are the*

minimum and maximum of the observed data $\mathbf{x}[1], \ldots, \mathbf{x}[n]$*, respectively. Then,* $\widehat{\mathcal{Q}}$ *can only have support points, or atoms, in the interval* $\left[\mathbf{x}_{min}, \mathbf{x}_{max}\right]$.

For many densities, the assumption of Corollary 1 is fulfilled. For example, the one-dimensional Gaussian density function $\mathcal{N}(\mu, 1)$

$$p\big(x|\underline{\theta} = \mu\big) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2}\right\}$$

is maximized for $\underline{\theta} = x$, i.e., for $\mu = x$. However, the Poisson density function

$$p\big(x|\underline{\theta} = \log\lambda\big) = \frac{\lambda^x \mathrm{e}^{-\lambda}}{x!}$$

is maximized for $\lambda = x$, i.e., $\underline{\theta} = \log x$. Consequently, $\widehat{\mathcal{Q}}$ will have points of support only in the interval $\big[\min\{\log(x)\}, \max\{\log(x)\}\big] = \big[\log(x_{min}), \log(x_{max})\big]$. Similarly, consider the Binomial case

$$p\left(x\big|\underline{\theta} = \log\frac{p}{1-p}\right) = \frac{N!}{x!(N-x)!} p^x (1-p)^{N-x}.$$

The Binomial density function is maximized for $p = x/N$, i.e., $\underline{\theta} = \log\frac{x/N}{1-x/N}$. Consequently, $\big[\min\left\{\log\frac{x/N}{1-x/N}\right\}, \max\left\{\log\frac{x/N}{1-x/N}\right\}\big]$ will be the interval for the points of support.

Corollary 1 has implications in that there is no need to search for the NPML estimates outside the range of observed data. This reduces the computational burden enormously and facilitates the critical initialization step.

Now suppose that the training set incorporates replications, i.e., there are only $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K)}$ different values. Consider the K-dimensional set

$$\Gamma = \left\{\big(p(\mathbf{x}^{(1)}|\underline{\boldsymbol{\theta}}), \ldots, p(\mathbf{x}^{(K)}|\underline{\boldsymbol{\theta}})\big) \,\big\|\, \underline{\boldsymbol{\theta}} \in \Theta\right\}.$$

Corollary 2 (Existence and number of NPML atoms [34])**.** *(a) If* Γ *is closed and bounded, then an NPML estimate* $\widehat{\mathcal{Q}}$ *exists. (b)* $\widehat{\mathcal{Q}}$ *has at most* K *points of support.*

Corollary 2 is based on a well-known theorem of Carathéodory and gives a bound on the maximum number of necessary support points or atoms. This

bound is the size of the training sample, n, if all data points are different from each other. However, if there are many replications, this bound is largely reduced. In practice, the bound for the number of support points given in Corollary 2 is seldom sharp and there will often be fewer support points required than the bound indicates [34].

There exist several suitable algorithms for constructing an NPML estimate. First, the Vertex Direction Method (VDM) and the Vertex Exchange Method (VEM) both consider the convex set of all discrete distributions in which one-point mixing distributions $\mathcal{Q}_{\underline{\boldsymbol{\theta}}} = \left\{\underline{\boldsymbol{\theta}}, \pi\right\}$ are interpreted as vertices of the simplex, and they both find the mixing distribution that maximizes the log-likelihood function by a succession of appropriate moves inside the simplex [34]. The directions of movement for the VDM is toward a vertex whereas the VEM moves parallel to the edges of the simplex. Second, the Expectation-Maximization (EM) algorithm is the most commonly used technique and is presented below.

C.2 The EM algorithm for exponential family distributions

First, the atoms $\underline{\boldsymbol{\theta}}[1], \ldots, \underline{\boldsymbol{\theta}}[m]$ are considered fixed and known. Because $\pi_1 + \pi_2 + \cdots + \pi_m = 1$, π_m can be replaced by $1 - \sum_{l=1}^{m-1} \pi_l$. Estimation of the point-mass probabilities $\pi_1, \ldots, \pi_m$ is performed by maximizing the following log-likelihood function

$$L(\mathcal{Q}) = \sum_{k=1}^{n} \log \sum_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l \tag{C.6}$$

$$= \sum_{k=1}^{n} \log \left\{ \sum_{l=1}^{m-1} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l + p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[m]\big) \left(1 - \sum_{l=1}^{m-1} \pi_l\right) \right\}, \tag{C.7}$$

where $p(\cdot|\cdot)$ is an exponential family distribution. Taking the partial derivatives with respect to $\pi_l, l = 1, \ldots, m-1$, gives

$$\frac{\partial L(\mathcal{Q})}{\partial \pi_l} = \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big) - p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[m]\big)}{\sum_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l},$$

yielding the following likelihood equations for $l = 1, \ldots, m-1$

$$\begin{aligned}
&\frac{\partial L(\mathcal{Q})}{\partial \pi_l} = \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big) - p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[m]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} = 0 \\
\iff &\sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} = \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[m]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} \qquad \text{(C.8)} \\
\iff &\sum_{l=1}^{m-1} \pi_l \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} = \sum_{l=1}^{m-1} \pi_l \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[m]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} \\
\iff &\sum_{l=1}^{m-1} \pi_l \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[j]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} + \pi_m \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[m]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} \\
&\qquad = \sum_{l=1}^{m-1} \pi_l \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[m]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} + \pi_m \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[m]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} \\
\iff &\sum_{l=1}^{m} \pi_l \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} = \sum_{l=1}^{m} \pi_l \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[m]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} \\
\iff &\sum_{k=1}^{n} \frac{\sum_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} = \sum_{l=1}^{m} \pi_l \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[m]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} \\
\iff &\sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[m]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} = n. \qquad \text{(C.9)}
\end{aligned}$$

Using equation (C.9) in equation (C.8) results in

$$\sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} = n \qquad \text{(C.10)}$$

for $l = 1, \ldots, m$. Equation (C.10) can also take the form of a *fixed point* equation:

$$\sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l}{n\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r} = \pi_l \qquad \text{(C.11)}$$

for $l = 1, \ldots, m$.

As it turns out, equation (C.11) is a special case of a more general fixed point algorithm, the Expectation-Maximization (EM) algorithm [68]. The EM approach introduces a *missing* or unobserved variable $\mathbf{z}_k = [z_{k1}, \ldots, z_{km}]$ for $k = 1, \ldots, n$, where n is the number of data samples and m the number of mixture components. This variable is an m-dimensional binary vector whose lth component equals 1 if the observed variable $\mathbf{x}[k]$ was drawn from the lth mixture component

and 0 otherwise and hence is also referred to as the mixture component indicator. Then, the observed data matrix $\mathbf{X} = \big[\mathbf{x}[1]^T, \ldots, \mathbf{x}[n]^T\big]^T$ is viewed as being *incomplete* since the associated component indicators are not available and the log-likelihood function defined by equation (C.6) is also called *incomplete*. The *complete* log-likelihood function is therefore declared to be:

$$\begin{aligned} L^{(c)}\big(\mathcal{Q}, \{\mathbf{z}_k\}_{k=1}^n\big) &= \log \prod_{k=1}^{n} \prod_{l=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)^{z_{kl}} \pi_l^{z_{kl}} \\ &= \underline{\sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)} + \sum_{k=1}^{n} \sum_{l=1}^{m} z_{kl} \log \pi_l, \end{aligned} \tag{C.12}$$

where the underlined term is independent of $\pi_l, l = 1, \ldots, m$. The EM algorithm overcomes the fact that the component indicator vectors $\mathbf{z}_k$, $k = 1, \ldots, n$, are unknown by iteratively working with the conditional expectation of the complete log-likelihood function given the observed data, which is computed using the current fit for the unknown parameters [53].

First, the E-step, or Expectation-step, allows one to obtain an estimate of the missing variables $\mathbf{z}_k = [z_{k1}, \ldots, z_{km}]^T, k = 1, \ldots, n$, by replacing them with their expected values given the data set $\big\{\mathbf{x}[k]\big\}_{k=1}^n$:

$$\begin{aligned} \widehat{z}_{kl} = \mathrm{E}\left\{z_{kl}|\mathbf{x}[k], \pi_1, \ldots, \pi_m\right\} &= \Pr\big(z_{kl} = 1|\mathbf{x}[k]\big) \\ &= \frac{\Pr\big(\mathbf{x}[k]|z_{kl} = 1\big)\Pr(z_{kl} = 1)}{\sum_{r=1}^{m} \Pr\big(\mathbf{x}[k]|z_{kr} = 1\big)\Pr(z_{kr} = 1)} \\ &= \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l}{\sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r}, \end{aligned} \tag{C.13}$$

for $l = 1, \ldots, m$, where the notation $\Pr(\cdot)$ expresses the probability of an event.

Then, the complete log-likelihood function becomes:

$$L^{(c)}\big(\mathcal{Q}, \{\widehat{\mathbf{z}}_k\}_{k=1}^n\big) = \sum_{k=1}^{n} \sum_{l=1}^{m} \widehat{z}_{kl} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big) + \sum_{k=1}^{n} \sum_{l=1}^{m} \widehat{z}_{kl} \log \pi_l. \tag{C.14}$$

Finally, maximizing equation (C.14) leads to the M-step, or Maximization-step and yields the estimates for the point-mass probabilities:

$$\widehat{\pi}_l = \frac{\sum_{k=1}^{n} \widehat{z}_{kl}}{n} = \sum_{k=1}^{n} \frac{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l]\big)\pi_l}{n \sum_{r=1}^{m} p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)\pi_r}, \tag{C.15}$$

which is the *fixed point* equation noticed earlier in equation (C.11) and which is simply the proportion of the data from each mixture component.

Now the component parameters $\underline{\boldsymbol{\theta}}[l], l = 1, \ldots, m$, are assumed to be unknown and need to be estimated in the M-step in the following way for $l = 1, \ldots, m$:

$$\begin{aligned}
\widehat{\underline{\boldsymbol{\theta}}}[l] &= \arg\max_{\underline{\boldsymbol{\theta}}[l]} L^{(c)}\big(\underline{\boldsymbol{\theta}}[l], \widehat{\pi}_l, l = 1, \ldots, m, \{\widehat{\mathbf{z}}_k\}_{k=1}^n\big) \\
&= \arg\max_{\underline{\boldsymbol{\theta}}[l]} \left\{ \sum_{k=1}^{n} \sum_{r=1}^{m} \widehat{z}_{kr} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big) + \sum_{k=1}^{n} \sum_{r=1}^{m} \widehat{z}_{kr} \log \widehat{\pi}_r \right\} \\
&= \arg\max_{\underline{\boldsymbol{\theta}}[l]} \sum_{k=1}^{n} \sum_{r=1}^{m} \widehat{z}_{kr} \log p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r]\big)
\end{aligned}$$

and, since $p(\cdot|\cdot)$ is assumed to be an exponential family distribution,

$$= \arg\max_{\underline{\boldsymbol{\theta}}[l]} \sum_{k=1}^{n} \sum_{r=1}^{m} \widehat{z}_{kr} \big\{ \underline{\boldsymbol{\theta}}[r]\mathbf{x}[k]^T - G(\underline{\boldsymbol{\theta}}[r]) \big\}.$$

For all $l = 1, \ldots, m$, the function $l(\underline{\boldsymbol{\theta}}[l])$ is defined as collecting the elements of the loss function $-\sum_{k=1}^{n} \sum_{r=1}^{m} \widehat{z}_{kr} \big\{ \underline{\boldsymbol{\theta}}[r]\mathbf{x}[k]^T - G(\underline{\boldsymbol{\theta}}[r]) \big\}$ that only depends on the vector $\underline{\boldsymbol{\theta}}[l]$. The computation of the gradient vector $\nabla_{\underline{\boldsymbol{\theta}}} l\big(\underline{\boldsymbol{\theta}}[l]\big)$ goes as follows, for $l = 1, \ldots, m$:

$$\begin{aligned}
l\big(\underline{\boldsymbol{\theta}}[l]\big) &= -\sum_{k=1}^{n} \widehat{z}_{kl} \big\{ \underline{\boldsymbol{\theta}}[l]\mathbf{x}[k]^T - G(\underline{\boldsymbol{\theta}}[l]) \big\} = \sum_{k=1}^{n} \widehat{z}_{kl} \big\{ G(\underline{\boldsymbol{\theta}}[l]) - \underline{\boldsymbol{\theta}}[l]\mathbf{x}[k]^T \big\} \\
\nabla_{\underline{\boldsymbol{\theta}}} l\big(\underline{\boldsymbol{\theta}}[l]\big) &= \frac{\partial l\big(\underline{\boldsymbol{\theta}}[l]\big)}{\partial \underline{\boldsymbol{\theta}}[l]} = \frac{\partial}{\partial \underline{\boldsymbol{\theta}}[l]} \sum_{k=1}^{n} \widehat{z}_{kl} \big\{ G(\underline{\boldsymbol{\theta}}[l]) - \underline{\boldsymbol{\theta}}[l]\mathbf{x}[k]^T \big\} \\
&= \sum_{k=1}^{n} \widehat{z}_{kl} \big\{ G'(\underline{\boldsymbol{\theta}}[l]) - \mathbf{x}[k] \big\} = \sum_{k=1}^{n} \widehat{z}_{kl} G'(\underline{\boldsymbol{\theta}}[l]) - \sum_{k=1}^{n} \widehat{z}_{kj} \mathbf{x}[k] \\
&= G'(\underline{\boldsymbol{\theta}}[l]) \sum_{k=1}^{n} \widehat{z}_{kl} - \sum_{k=1}^{n} \big(\widehat{z}_{kl} \mathbf{x}[k] \big),
\end{aligned}$$

where

$$G'(\underline{\boldsymbol{\theta}}[l]) = \left[\frac{\partial G(\underline{\boldsymbol{\theta}}[l])}{\partial \underline{\theta}_1[l]}, \ldots, \frac{\partial G(\underline{\boldsymbol{\theta}}[l])}{\partial \underline{\theta}_d[l]} \right] = \big[g(\underline{\theta}_1[l]), \ldots, g(\underline{\theta}_d[l]) \big],$$

as in Section 4. Consequently,

$$\nabla_{\underline{\boldsymbol{\theta}}} l\big(\widehat{\underline{\boldsymbol{\theta}}}[l]\big) = 0 \Longleftrightarrow G'(\widehat{\underline{\boldsymbol{\theta}}}[l] \sum_{k=1}^{n} \widehat{z}_{kl} - \sum_{k=1}^{n} \big(\widehat{z}_{kl}\mathbf{x}[k]\big) = 0$$
$$\Longleftrightarrow G'(\widehat{\underline{\boldsymbol{\theta}}}[l] = \frac{\sum_{k=1}^{n} \widehat{z}_{kl}\mathbf{x}[k]}{\sum_{k=1}^{n} \widehat{z}_{kl}},$$

where $G'(\widehat{\underline{\boldsymbol{\theta}}}[l]$ takes on a different form depending on the exponential family distribution considered. Several one-dimensional examples are presented below:

(i) Gaussian with unit-variance $\mathcal{N}(\mu, 1)$: $G(\underline{\theta}) = \underline{\theta}^2/2$ and $G'(\underline{\theta}) = \underline{\theta}$. Hence, for $l = 1, \ldots, m$,

$$\widehat{\underline{\theta}}[l] = \frac{\sum_{k=1}^{n} \widehat{z}_{kl} x[k]}{\sum_{k=1}^{n} \widehat{z}_{kl}}.$$

(ii) Exponential(λ): $G(\underline{\theta}) = -\log(-\underline{\theta})$ and $G'(\underline{\theta}) = 1/\underline{\theta}$. Hence, for $l = 1, \ldots, m$,

$$\widehat{\underline{\theta}}[l] = \frac{\sum_{k=1}^{n} \widehat{z}_{kl}}{\sum_{k=1}^{n} \widehat{z}_{kl} x[k]}.$$

(iii) Bernoulli(p): $G(\underline{\theta}) = \log\big(1 + \exp\{\underline{\theta}\}\big)$ and $G'(\underline{\theta}) = \exp\{\underline{\theta}\}/\big(1 + \exp\{\underline{\theta}\}\big)$. Hence, for $l = 1, \ldots, m$,

$$\frac{\exp\big\{\widehat{\underline{\theta}}[l]\big\}}{1 + \exp\big\{\widehat{\underline{\theta}}[l]\big\}} = \frac{\sum_{k=1}^{n} \widehat{z}_{kl} x[k]}{\sum_{k=1}^{n} \widehat{z}_{kl}}$$
$$\Longleftrightarrow \frac{1}{1 + \exp\big\{-\widehat{\underline{\theta}}[l]\big\}} = \frac{\sum_{k=1}^{n} \widehat{z}_{kl} x[k]}{\sum_{k=1}^{n} \widehat{z}_{kl}}$$
$$\Longleftrightarrow \sum_{k=1}^{n} \widehat{z}_{kl} = \Big(1 + \exp\big\{-\widehat{\underline{\theta}}[l]\big\}\Big) \sum_{k=1}^{n} \widehat{z}_{kl} x[k]$$
$$\Longleftrightarrow \exp\big\{-\widehat{\underline{\theta}}[l]\big\} = \frac{\sum_{k=1}^{n} \widehat{z}_{kl} - \sum_{k=1}^{n} \widehat{z}_{kl} x[k]}{\sum_{k=1}^{n} \widehat{z}_{kl} x[k]}$$
$$\Longleftrightarrow \widehat{\underline{\theta}}[l] = \log \frac{\sum_{k=1}^{n} \widehat{z}_{kl} x[k]}{\sum_{k=1}^{n} \widehat{z}_{kl} - \sum_{k=1}^{n} \widehat{z}_{kl} x[k]}.$$

(iv) Binomial(p, N): $G(\underline{\theta}) = N \log\big(1 + \exp\{\underline{\theta}\}\big)$ and $G'(\underline{\theta}) = N \frac{\exp\{\underline{\theta}\}}{1+\exp\{\underline{\theta}\}}$. Hence, for $l = 1, \ldots, m$,

$$\widehat{\underline{\theta}}[l] = \log \frac{\sum_{k=1}^{n} \widehat{z}_{kl} x[k]}{N \sum_{k=1}^{n} \widehat{z}_{kl} - \sum_{k=1}^{n} \widehat{z}_{kl} x[k]}.$$

(v) Poisson(λ): $G(\underline{\theta}) = \exp\{\underline{\theta}\}$ and $G'(\underline{\theta}) = \exp\{\underline{\theta}\}$. Hence, for $l = 1, \ldots, m$,

$$\exp\left\{\widehat{\underline{\theta}}[l]\right\} = \frac{\sum_{k=1}^{n} \widehat{z}_{kl} x[k]}{\sum_{k=1}^{n} \widehat{z}_{kl}} \Longleftrightarrow \underline{\theta}[l] = \log \frac{\sum_{k=1}^{n} \widehat{z}_{kl} x[k]}{\sum_{k=1}^{n} \widehat{z}_{kl}}.$$

The EM algorithm for the Non-Parametric Maximum Likelihood estimation for exponential family distribution is summarized below in Table C.1.

Table C.1 EM algorithm for Non-Parametric Maximum Likelihood estimation in exponential family distributions mixture models.

Algorithm: EM algorithm for exponential family distributions mixture models

Input: a set of observations $\left\{\mathbf{x}[k]\right\}_{k=1}^{n} \subseteq \mathbb{R}^d$, an exponential family distribution $p(\cdot)$ defined by its cumulant generating function $G(\cdot)$, a number of atoms m.

Output: the NPML estimator that maximizes the complete log-likelihood function $L^{(c)}\big(\mathcal{Q}, \{\mathbf{z}_k\}_{k=1}^{n}\big)$: $\widehat{\mathcal{Q}} = \left\{\widehat{\underline{\boldsymbol{\theta}}}[l], \widehat{\pi}_l\right\}_{l=1}^{m}$.

Method:

Initialize $\left\{\underline{\boldsymbol{\theta}}[l], \pi_l\right\}_{l=1}^{m}$ with $\pi_l \geq 0$ for all l and $\sum_{l=1}^{m} \pi_l = 1$; $\underline{\boldsymbol{\theta}}[l] \in \Theta$ for all l;
repeat
{The Expectation Step}
for $k = 1$ to n **do**
for $l = 1$ to m **do**
$\widehat{z}_{kl} \longleftarrow \frac{p(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[l])\pi_l}{\sum_{r=1}^{m} p(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[r])\pi_r}$
end for
end for
{The Maximization Step}
for $l = 1$ to m **do**
$\widehat{\pi}_l \longleftarrow \frac{1}{n}\sum_{k=1}^{n} \widehat{z}_{kl}$
$\underline{\boldsymbol{\theta}}[l] \longleftarrow G'(\underline{\boldsymbol{\theta}}[l]) = \frac{\sum_{k=1}^{n} \widehat{z}_{kl}\mathbf{x}[k]}{\sum_{k=1}^{n} \widehat{z}_{kl}}$
end for
until *convergence*;
return $\widehat{\mathcal{Q}} = \left\{\widehat{\underline{\boldsymbol{\theta}}}[l], \widehat{\pi}_l\right\}_{l=1}^{m}$.

Often, the choice of initial values for the EM algorithm is critical. Three methods for choosing initial values in the case $m = 2$ are proposed in [34]. In the first two strategies, starting values for $\underline{\boldsymbol{\theta}}[1]$, $\underline{\boldsymbol{\theta}}[2]$ and π are found by classifying the n data samples into two disjoint sets of size s and $n-s$ respectively ($s < n$). Then,

$\underline{\boldsymbol{\theta}}[1]$ is estimated by the arithmetic mean of the first set and $\underline{\boldsymbol{\theta}}[2]$ by the arithmetic mean of the second set. The prior π is estimated as s/n. In particular, ordered values $\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(n)}$ are considered, and the sets are updated by including one observation at each step. Thus, at first, set 1 consists of $\mathbf{x}_{(1)}$, set 2 of $\mathbf{x}_{(2)}, \ldots, \mathbf{x}_{(n)}$. In the second step, set 1 consists of $\mathbf{x}_{(1)}$,$\mathbf{x}_{(2)}$ whereas set 2 contains $\mathbf{x}_{(3)}, \ldots, \mathbf{x}_{(n)}$, and so forth. The procedure considers $n-1$ partitions. Strategies differ in the way they select the optimal partition. Strategy I maximizes

$$L_I\big(\underline{\boldsymbol{\theta}}[1], \underline{\boldsymbol{\theta}}[2], \pi\big) = \sum_{k=1}^{n} \log\left\{p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[1]\big)\pi + p\big(\mathbf{x}[k]|\underline{\boldsymbol{\theta}}[2]\big)(1-\pi)\right\}$$

in $\underline{\boldsymbol{\theta}}[1] = \bar{\mathbf{x}}_1$ the arithmetic mean of set 1 and $\underline{\boldsymbol{\theta}}[2] = \bar{\mathbf{x}}_1$ the arithmetic mean of set 2, and $\pi = s/n$, meaning that the log-likelihood function has to be evaluated $n-1$ times. Strategy II is minimizing the total sum of squares in s

$$L_{II}\big(\underline{\boldsymbol{\theta}}[1], \underline{\boldsymbol{\theta}}[2], \pi\big) = \sum_{k=1}^{s} \big(\mathbf{x}[k] - \bar{\mathbf{x}}_1\big)^2 + \sum_{k=s+1}^{n} \big(\mathbf{x}[k] - \bar{\mathbf{x}}_2\big)^2,$$

where $\bar{\mathbf{x}}_1$ and $\bar{\mathbf{x}}_2$ are the means of the first s and the remaining $n-s$ ordered values, respectively. In Strategy III the means are chosen as values of certain order statistics. The following values are chosen: $\underline{\boldsymbol{\theta}}[1] = \mathbf{x}_{(1)}$, $\underline{\boldsymbol{\theta}}[1] = \mathbf{x}_{(n)}$ (Strategy III.1), $\underline{\boldsymbol{\theta}}[1] = \mathbf{x}_{(5)}$, $\underline{\boldsymbol{\theta}}[1] = \mathbf{x}_{(n-5)}$ (Strategy III.5), $\underline{\boldsymbol{\theta}}[1] = \mathbf{x}_{(30)}$, $\underline{\boldsymbol{\theta}}[1] = \mathbf{x}_{(n-30)}$ (Strategy III.30). The prior π is chosen to be $1/2$ in all three cases.

D Work on UC Irvine data sets

This chapter demonstrates the utility of the Generalized Linear Statistics (GLS) approach with experiments on real data sets, for which classification in parameter space often outperforms classification in data space. The data sets used here are from the University of California, Irvine machine learning repository [41]. Table D.1 presents characteristics of the data sets used in this work, i.e., their name, the number of classes to identify as well as the number of instances in both training and test sets.

The Twenty Newsgroups and the Reuters-21578 data sets account for most of the experimental work in text categorization. Text categorization is the activity of labeling natural language texts with thematic categories from a predefined set [111] and is one example of information retrieval tasks. The Abalone data set task is to predict the age of an abalone based on physical measurements and can be seen as either a regression or a classification problem.

Table D.1 Characteristics of the University of California, Irvine machine learning repository data sets used in this work.

data set	# classes	training set	test set
Twenty Newsgroups	3	1764	1236
Reuters-21578	10	6490	2545
Abalone	3	2506	1671

For each data set, a low-dimensional parameter subspace is identified using GLS while classical Principal Component Analysis (PCA) selects a low-dimensional data subspace. Classification is performed on both subspaces and performances are compared. The benefits of decision making in parameter space rather than in data space as with more classical approaches are illustrated with examples of categorical data supervised and unsupervised text categorization and mixed-type data classification. As a text document preprocessing tool, an extension from binary to categorical data of the conditional mutual information maximization based feature selection algorithm is presented.

D.1 Twenty Newsgroups data set

The Twenty Newsgroups data set consists of Usenet articles collected from twenty different newsgroups. Each newsgroup contains 1000 articles.

This work considers the three following newsgroups: sci.med comp.sys.mac.hardware and comp.sys.ibm.pc.hardware. A classification problem with two distinct classes is studied, the first class consisting of the newsgroup sci.med, the second class consisting of the newsgroups comp.sys.ibm.pc.hardware and comp.sys.mac.hardware. The first class articles are assigned a target value equal to 1 and the second class a target value equal to 0.

D.1.1 Preprocessing and document representation for text categorization

It has been acknowledged by the text categorization community that words seem to work well as features of a document for many classification tasks. In addition, it is usually assumed that the ordering of the words in a document does not matter. Hence, a document can simply be represented as a bag of words, i.e., as a vector for which each distinct word is a feature [112]. There are two ways to characterize the value of each feature that are commonly used in the literature:

Boolean and *tf*×*idf* weighting schemes. In Boolean weighting, the weight of a word is considered to be 1 if the word appears in the document and 0 otherwise. We choose to characterize the value of each feature by using the *tf*×*idf* (term frequency × inverse document frequency) document representation scheme proposed in [113]. This scheme argues that terms (or words) appearing in documents should be weighted proportional to the term frequency and inversely proportional to the document frequency. The weight is a statistical measure used to evaluate how important a word is to a corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. This weighting scheme is commonly used for document representation and the combination of tf×idf weights and document length normalization have been shown to perform generally better retrieval results [111, 113–115]; interestingly, in practice, the Boolean approach does not always perform worse than the tf×idf approach [116]. The *term frequency tf* is the number of times a specific word occurs in a specific document. The *document frequency df* is the number of documents in which the specific word occurs at least once. The *inverse document frequency idf* can be calculated from the document frequency as follows:

$$idf = \log\left(\frac{\text{total \# of documents}}{df}\right).$$

Therefore, tf×idf weighting goes as follows:

$$w_i = tf_i \cdot \log\left(\frac{\text{total \# of documents}}{df_i}\right)$$

for each feature, i.e., for all i, and tf×idf weighting with length normalization is:

$$w_i = \frac{tf_i \cdot \log\left(\frac{\text{total \# of documents}}{df_i}\right)}{\sqrt{\sum_{j=1}^{|T|}\left[tf_j \cdot \log\left(\frac{\text{total \# of documents}}{df_j}\right)\right]^2}},$$

where $|T|$ is the length of the document, i.e., the number of distinct words in the document (after stopword removal and stemming is performed as explained below). Length normalization ensures that each document vector is of unit length,

removing the advantage that long documents have over short documents with respect to information retrieval [111]. However, if a document is long, but has quite often a term that represents key information for a specific text categorization task, normalization would reduce the importance of the term as compared to a short document, where the term appears equally often in absolute term. Hence we decide to discard the length normalization step.

We choose to bin the weights and to work with integer valued weights (five bins are selected), i.e., categorical features.

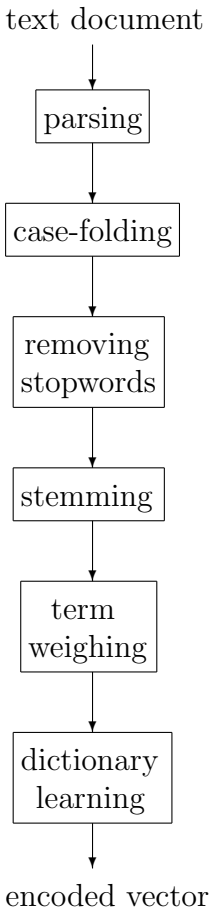

Figure D.1 Preprocessing and document representation for text categorization.

In regard to the newsgroup articles, following the steps described in Figure D.1, we first chose to discard all header fields such as Cc, Bcc, Message-ID, as well

as the Subject field (this step is called parsing). Case-folding, which stands for converting all the characters in a document into the same case, is performed by converting all the characters into lower-case. We use a stop list, i.e., a list of words that will not be taken into account. Indeed, there are words such as pronouns, prepositions and conjunctions which are encountered very frequently but carry no useful information about the content of the document. We used the following stop list: ftp://ftp.cs.cornell.edu/pub/smart/english.stop. It consists of 571 stopwords and is commonly used in the literature. It yields a drastic reduction in the number of features. Then, some simple stemming is performed, such as removing the third person and plural "s". In addition to removing very frequent words with the stop list, we remove rare words, i.e., words appearing less than 10 times in the corpus. At this point, each document is a vector in a 4383-dimensional space, i.e., 4383 distinct words were identified to represent the newsgroups documents.

Last, we construct a dictionary, and hence reduce the dimensionality of the feature space. There are various methods commonly applied for dimensionality reduction in document categorization [112]. We chose a conditional mutual information based approach to select a dictionary of 150 words. We modify the binary feature selection with conditional mutual information algorithm proposed in [117] to fit a categorical feature. The feature selection algorithm proposed in [117] is based on the Conditional Mutual Information Maximization (CMIM) criterion and selects features that maximize both the information about the class and the independence between features. The modification from binary to categorical is simple; following the definition of entropy and mutual information shown in [117], the summations are changed from summing over two values to summing over five values, i.e., the number of bins selected.

We use this data set leaving out a randomly selected 40% of the instances of each class to use as a test set. The training set then consists of 1764 instances and the test set consists of 1236 instances. The dictionary is learned using the training set only. Table D.2 presents the first twenty words of the dictionary

Table D.2 Twenty Newsgroups data set: first twenty words of the dictionary learned to differentiate the newsgroup sci.med from the newsgroups comp.sys.mac.hardware and comp.sys.ibm.pc.hardware.

doctor
card
mac
drive
disease
medical
treatment
food
patient
effect
medicine
drug
skepticism
pc
body
health
blood
study
hardware
infection

learned with the purpose of differentiating the newsgroup sci.med from the two other newsgroups.

D.1.2 Classification and clustering results

Figure D.2 and Figure D.3 represent the training set documents in the low-dimensional subspace of the parameter space learned with classical PCA for a dimension q of the subspace equal to 2 and 3. Similarly, Figure D.4 and Figure D.5 represent the training set documents in the low-dimensional subspace of the parameter space learned with the GLS approach using a Binomial distribution ($N = 5$) for a dimension q of the subspace equal to 2 and 3.

Supervised approach: Table D.3 presents the classification performances

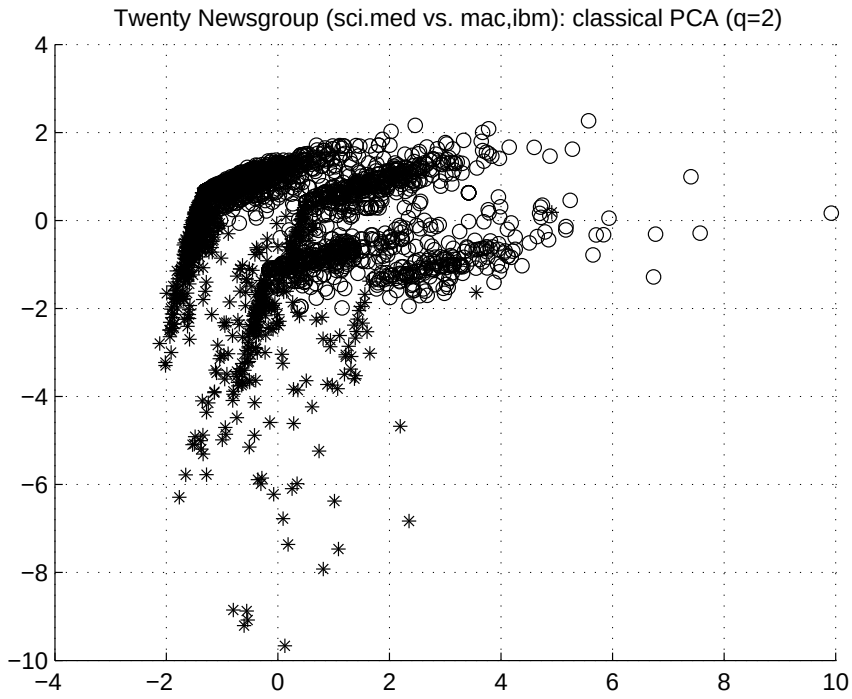

Figure D.2 Twenty Newsgroups data set: training documents in the lower dimensional subspace of the parameter space learned with classical PCA, $q = 2$ (sci.med: *, comp.sys.ibm.pc.hardware and comp.sys.mac.hardware: ∘).

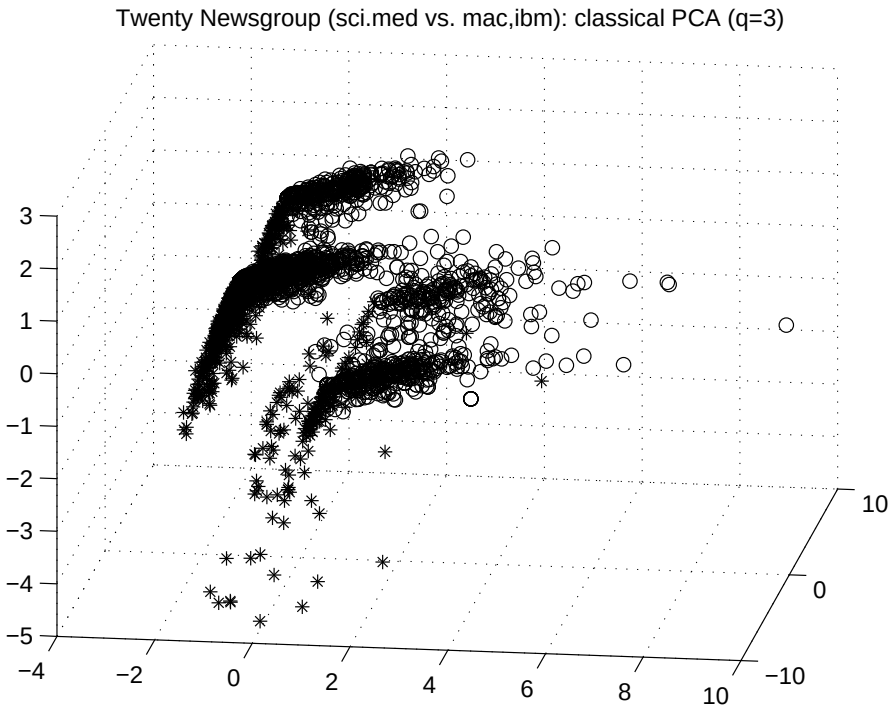

Figure D.3 Twenty Newsgroups data set: training documents in the lower dimensional subspace of the parameter space learned with classical PCA, $q = 3$ (sci.med: *, comp.sys.ibm.pc.hardware and comp.sys.mac.hardware: ∘).

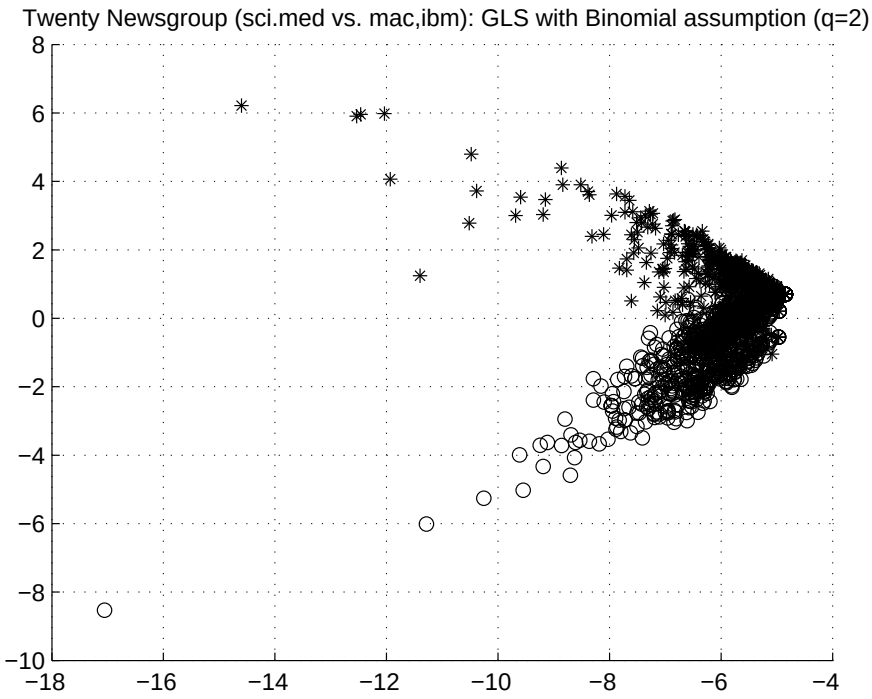

Figure D.4 Twenty Newsgroups data set: training documents in the low-dimensional parameter subspace learned with the GLS approach (Binomial, $N = 5$), $q = 2$ (sci.med: *, comp.sys.ibm.pc.hardware and comp.sys.mac.hardware: $\circ$).

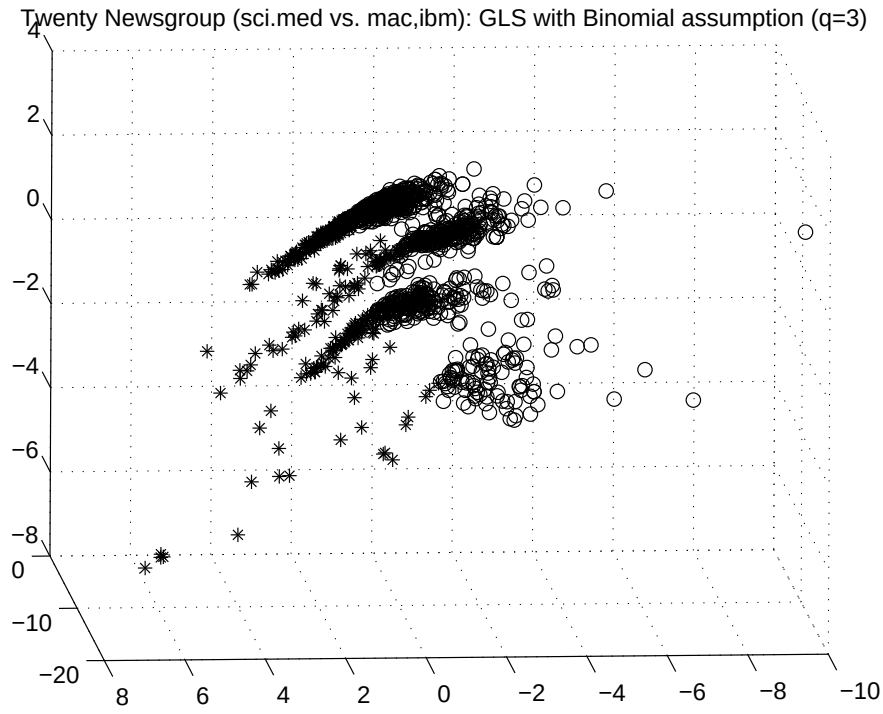

Figure D.5 Twenty Newsgroups data set: training documents in the low-dimensional parameter subspace learned with the GLS approach (Binomial, $N = 5$), $q = 3$ (sci.med: *, comp.sys.ibm.pc.hardware and comp.sys.mac.hardware: $\circ$).

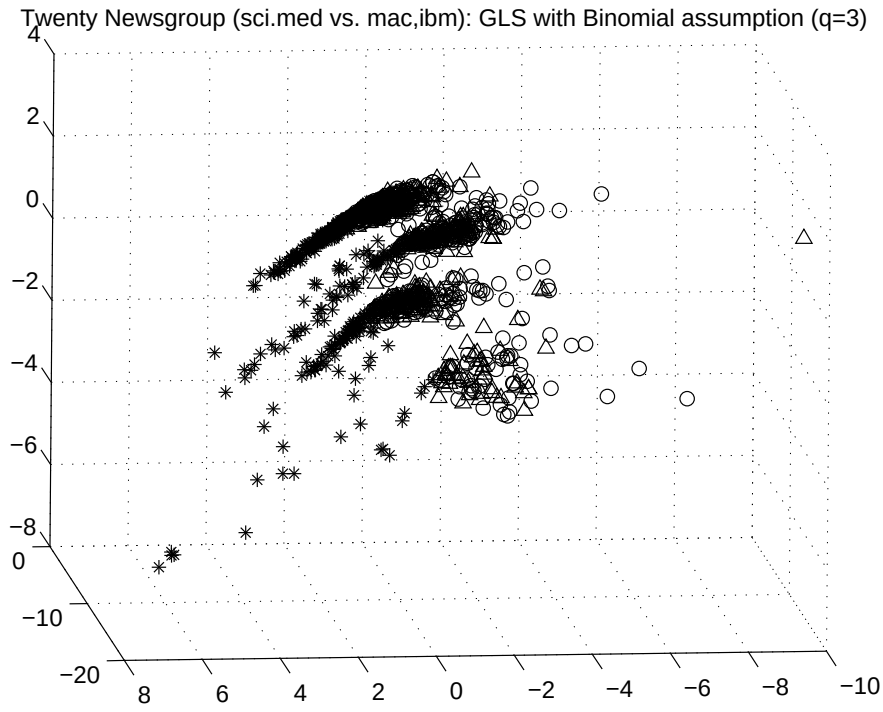

Figure D.6 Twenty Newsgroups data set: training documents in the low-dimensional parameter subspace learned with the GLS approach (Binomial, $N = 5$), $q = 3$ (sci.med: *, comp.sys.ibm.pc.hardware: $\circ$, comp.sys.mac.hardware: $\triangle$).

of Support Vector Machines (SVMs) on the q-dimensional latent variable space learned with classical PCA and the GLS framework with a Binomial distribution ($N = 5$). The SVMs results were obtained with the SVMs and Kernel Methods Matlab Toolbox developed at INSA, France [118]. The SVMs regularization parameter C value was chosen to give the best classification performance on a subset of the training set called validation set. We subdivided the training set (1764 instances) into 1071 training instances and 693 validation instances for this purpose. We then learned the SVMs with the previously chosen regularization parameter value on the full 1764 training instances and evaluated the performance on the test set (1236 instances). We did not further optimize performance by tuning parameters to achieve optimal performance on the test set. The number in parentheses is the number of support vectors obtained during the training phase. The classification performance is expressed in terms of the percentage of correctly classified documents.

Classification effectiveness is often measured in terms of *precision* and *recall* in the text categorization community [111]. Precision with respect to a class $\mathcal{C}_i$ (π_i) is defined as the probability that, if a random document is classified under $\mathcal{C}_i$, this decision is correct. Recall with respect to a class $\mathcal{C}_i$ (ρ_i) is defined as the probability that, if a random document ought to be classified under $\mathcal{C}_i$, this decision is taken. These probabilities are estimated in terms of the contingency table for $\mathcal{C}_i$ on a given test set as follows:

$$\widehat{\pi}_i = \frac{TP_i}{TP_i + FP_i} \quad \text{and} \quad \widehat{\rho}_i = \frac{TP_i}{TP_i + FN_i},$$

where TP_i, FP_i and FN_i refer to the sets of *true positives with respect to* $\mathcal{C}_i$ (documents correctly deemed to belong to class $\mathcal{C}_i$), *false positives with respect to* $\mathcal{C}_i$ (documents incorrectly deemed to belong to class $\mathcal{C}_i$), and *false negatives with respect to* $\mathcal{C}_i$ (documents incorrectly deemed not to belong to class $\mathcal{C}_i$). The notion of *breakeven point* is used to describe the value at which π equals ρ. Additionally, the F_1 measure combines precision and recall, attributing equal importance to π

Table D.3 Twenty Newsgroups data set: SVMs classification performances on the q-dimensional latent variable space learned with classical PCA and GLS with a Binomial distribution (1764 training instances and 1236 test instances).

	classical PCA training set	classical PCA test set	GLS Binomial training set	GLS Binomial test set
$q = 1$	85.32% (14)	**84.95**%	85.09% (17)	84.87%
$q = 2$	95.75% (42)	**94.17**%	93.54% (119)	91.50%
$q = 3$	96.03% (77)	94.98%	96.54% (129)	**95.57**%
$q = 4$	96.15% (111)	94.74%	96.37% (174)	**94.82**%
$q = 5$	96.20% (167)	94.98%	96.71% (228)	**95.06**%
$q = 6$	96.26% (237)	**94.90**%	96.77% (311)	**94.90**%
$q = 8$	96.43% (382)	**94.58**%	96.54% (469)	94.09%
$q = 10$	96.09% (524)	**93.69**%	97.00% (627)	93.53%

Table D.4 Averaging precision, recall and F_1 measure across different classes.

	microaveraging (μ)	macroaveraging (M)
precision (π)	$\widehat{\pi}^{\mu} = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i}{\sum_{i=1}^{\lvert\mathcal{C}\rvert}(TP_i+FP_i)}$	$\widehat{\pi}^{M} = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \pi_i}{\lvert\mathcal{C}\rvert} = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \frac{TP_i}{TP_i+FP_i}}{\lvert\mathcal{C}\rvert}$
recall (ρ)	$\widehat{\rho}^{\mu} = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i}{\sum_{i=1}^{\lvert\mathcal{C}\rvert}(TP_i+FN_i)}$	$\widehat{\rho}^{M} = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \pi_i}{\lvert\mathcal{C}\rvert} = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \frac{TP_i}{TP_i+FN_i}}{\lvert\mathcal{C}\rvert}$
F_1	$F_1^{\mu} = \frac{2\cdot\sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i}{2\cdot\sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i+\sum_{i=1}^{\lvert\mathcal{C}\rvert} FP_i+\sum_{i=1}^{\lvert\mathcal{C}\rvert} FN_i}$	$F_1^{M} = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} F_{1,i}}{\lvert\mathcal{C}\rvert} = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \frac{2\cdot TP_i}{2\cdot TP_i+FP_i+FN_i}}{\lvert\mathcal{C}\rvert}$

and ρ:

$$F_1 = \frac{2 \cdot \pi\rho}{\pi + \rho}.$$

When effectiveness is computed for several classes, the results for individual classes can be averaged in two ways: *microaveraging*, where π and ρ are obtained by summing over all individual classes (the subscript "μ" indicates microaveraging), and *macroaveraging*, where π and ρ are first evaluated "locally" for each class and then "globally" by averaging over the results of the different classes (the subscript "M" indicates macroaveraging) [111], cf. Table D.4.

Tables D.5, D.6, D.7 and D.8 compare classification performances on the q-dimensional latent variable space learned with classical PCA and GLS with a Binomial distribution in terms of precision, recall and F_1 measure. Table D.5 compares logistic regression classification performances, Table D.6 linear discriminant classification performances, Table D.7 Naive Bayes classification performances and Table D.8 k-NN ($k = 5$) classification performances. These results were obtained by using the MatlabArsenal toolbox, a package for classification algorithms [119]. The classification performances are often very similar, at times at the advantage of GLS (linear discriminant classifier for $q = 4$ and $q = 10$, Naive Bayes classifier for $q = 2$ to 4, k-NN classifier for $q = 4$, 6 and 8).

Unsupervised approach: A simple K-means algorithm is used to cluster the training documents into two distinct classes, cf. Figure D.7. Based on this clus-

Table D.5 Twenty Newsgroups data set: logistic regression classification performances on the q-dimensional latent variable space learned with classical PCA and GLS with a Binomial distribution (1236 test instances).

	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.6906	0.6923	**0.6915**	0	0	0
$q = 2$	0.8872	0.8317	**0.8586**	0.9370	0.7861	0.8549
$q = 3$	0.9391	0.8894	**0.9136**	0.9136	0.8389	0.8747
$q = 4$	0.9569	0.9063	**0.9309**	0.9463	0.8894	0.9170
$q = 5$	0.9641	0.9038	**0.9330**	0.9636	0.8918	0.9263
$q = 6$	0.9666	0.9038	**0.9342**	0.9636	0.8918	0.9263
$q = 8$	0.9716	0.9063	**0.9378**	0.9636	0.8918	0.9263
$q = 10$	0.9692	0.9063	**0.9366**	0.9691	0.9038	0.9353

Table D.6 Twenty Newsgroups data set: linear discriminant classification performances on the q-dimensional latent variable space learned with classical PCA and GLS with a Binomial distribution (1236 test instances).

	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.5045	0.8149	**0.6232**	0.3677	0.6603	0.4744
$q = 2$	0.7843	0.9351	**0.8531**	0.7844	0.8918	0.8346
$q = 3$	0.9388	0.8846	**0.9109**	0.8641	0.8558	0.8599
$q = 4$	0.9389	0.8870	0.9122	0.8830	0.9615	**0.9206**
$q = 5$	0.9038	0.9712	**0.9363**	0.8931	0.9639	0.9272
$q = 6$	0.9038	0.9712	**0.9363**	0.8914	0.9663	0.9273
$q = 8$	0.9040	0.9736	**0.9375**	0.8813	0.9639	0.9208
$q = 10$	0.8904	0.9760	0.9312	0.9691	0.9038	**0.9353**

Table D.7 Twenty Newsgroups data set: Naive Bayes classification performances on the q-dimensional latent variable space learned with classical PCA and GLS with a Binomial distribution (1236 test instances).

	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.5366	0.7404	**0.6222**	0.4314	0.0529	0.0942
$q = 2$	0.9077	0.5913	0.7162	0.8950	0.8197	**0.8557**
$q = 3$	0.9817	0.6442	0.7779	0.9829	0.6923	**0.8124**
$q = 4$	0.9693	0.6827	0.8011	0.8329	0.7909	**0.8113**
$q = 5$	0.7364	0.9135	**0.8155**	0.7045	0.8365	0.7648
$q = 6$	0.7303	0.9375	**0.8211**	0.6786	0.8678	0.7616
$q = 8$	0.7046	0.9231	**0.7992**	0.6551	0.8630	0.7448
$q = 10$	0.7183	0.9255	**0.8088**	0.6679	0.9036	0.7681

Table D.8 Twenty Newsgroups data set: k-NN ($k = 5$) classification performances on the q-dimensional latent variable space learned with classical PCA and GLS with a Binomial distribution (1236 test instances).

	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.7724	0.6851	**0.7261**	0.4398	0.3774	0.4062
$q = 2$	0.8862	0.9543	**0.9190**	0.8744	0.8870	0.8807
$q = 3$	0.8860	0.9712	**0.9266**	0.8911	0.9639	0.9261
$q = 4$	0.8855	0.9663	0.9241	0.8877	0.9688	**0.9264**
$q = 5$	0.8936	0.9688	**0.9296**	0.8874	0.9663	0.9252
$q = 6$	0.8879	0.9712	0.9277	0.8936	0.9688	**0.9296**
$q = 8$	0.8862	0.9543	0.9190	0.8867	0.9591	**0.9215**
$q = 10$	0.8599	0.9591	**0.9068**	0.8447	0.9543	0.8962

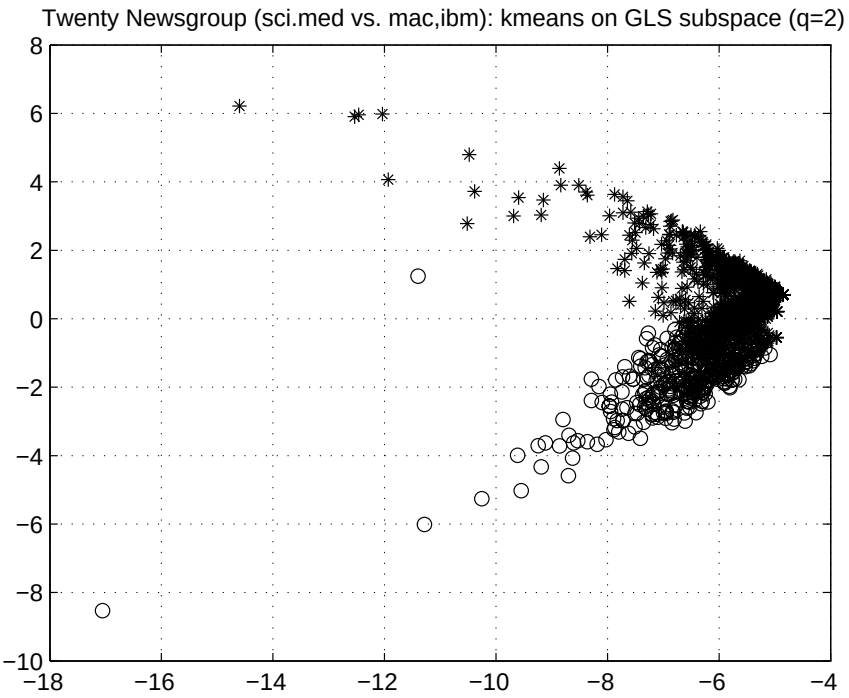

Figure D.7 Twenty Newsgroups data set: k-means results for a two-class classification of the training documents in GLS subspace ($q = 2$).

tering information, a linear discriminant is learned on the training documents and used to classify the test documents. Figure D.8 presents the corresponding ROC curve for this unsupervised approach performed on both the GLS subspace and the classical PCA subspace ($q = 2$). The performance is best when the unsupervised approach is used on the GLS subspace rather than on the classical PCA subspace. In this example, even though it is of interest, we did not further investigate the impact of the value for q on the performance.

D.2 Reuters-21578, Distribution 1.0 data set

The Reuters-21578 text categorization test collection Distribution 1.0 is considered as the standard benchmark for automatic document organization systems and consists of documents that appeared on the Reuters newswire in 1987 [120]. This corpus contains 21578 documents assigned to 135 different economic

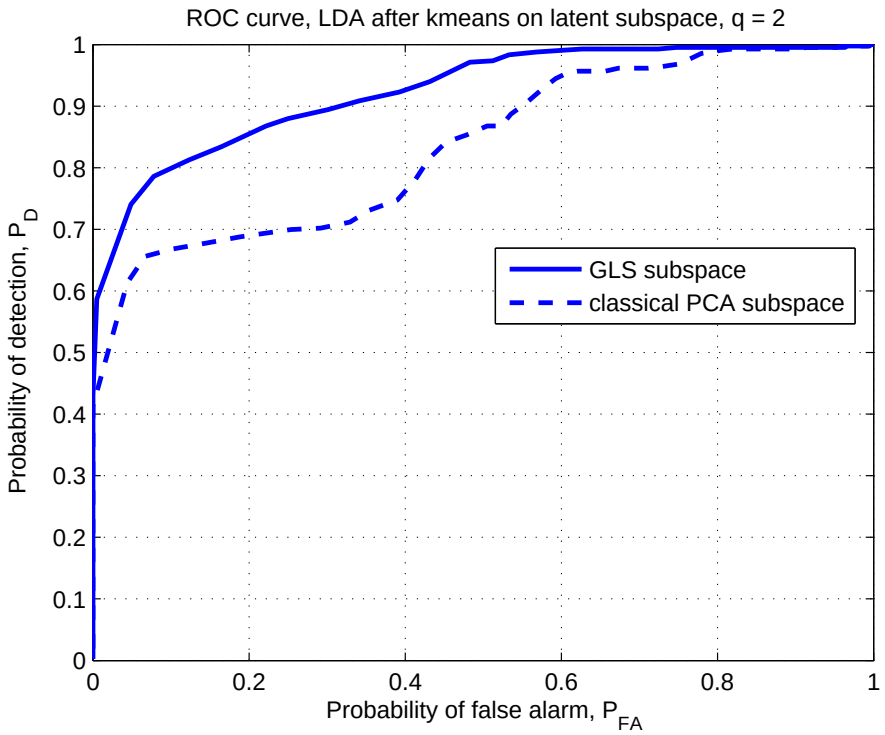

Figure D.8 Twenty Newsgroups data set: ROC curve for the unsupervised approach learned on the GLS subspace (solid line) and the classical PCA subspace (dashed line) ($q = 2$).

Table D.9 The ten topics with the highest number of training documents in the Reuters-21578 data set with the number of documents belonging to each topic in the training and test sets.

topics	training set	test set
earn	2877	1087
acq	1650	719
money-fx	538	179
grain	433	149
crude	389	189
trade	369	118
interest	347	131
wheat	212	71
ship	197	89
corn	181	56

subject categories called *topics*. The topics are not disjoint. For the training test division of the data, the "Modified Apte" (ModApte) split is used as suggested on the README file describing the data set to divide the corpus into a training set of 9603 documents and a test set of 3299 documents. We reduce the size of the training set and test set by only considering the ten topics that have the highest number of training documents as suggested in [121]. These topics are given in Table D.9 and yield a training set of 6490 documents and a test set of 2545 documents. These topics cover almost all of the data, hence, researchers are able to restrict their work to them and still capture the essence of the data set.

The data is preprocessed as explained in Section D.1.1: parsing, case-folding, elimination of words from a stop list, stemming by using Porter's stem-

ming algorithm commonly used for word stemming in English [122], elimination of words that appear less than 20 times in the corpus, tf×idf weighting (no length normalization). At this point, each document is represented as a vector in a 3613-dimensional space, i.e., 3613 distinct words were identified.

Then, we choose to bin the weights and to work with integer valued weights (5 bins are selected), i.e., categorical features. A dictionary of 50 words is learned using the following approach. The dictionary is learned on the training set only. The dictionary is build independently for each of the 10 classes. Feature selection was incremental. First we do a backward selection to 300 features with linear regression. From these 300 features, we use a logistic regression with number of iterations reduced down to just 5 for convergence, and do a backward selection down to 100 features. Finally we do a standard full-convergence logistic regression from those 100 down to 50 features (they differ by topic, of course).

Tables D.10 to D.19 compare *supervised* classification performances on the q-dimensional latent variable space learned with classical PCA and GLS with a Binomial distribution (1087 positive test instances) for each of the top ten categories of the Reuters-21578 data set, using logistic regression, linear discriminant and Naive Bayes classifiers. Tables D.20 compare classification performances micro- and macroaveraged over the same top ten categories for the linear discriminant classifier. The averaging is performed as explained in Table D.4. Microaveraging and macroaveraging methods give quite different results: the linear discriminant classifier performs better based on GLS information than based on classical PCA information when the macroaveraging method is used, whereas microaveraging emphasizes how similar the two results are. It is known that the ability of a classifier to behave well on categories with few positive training instances will be highlighted by macroaveraging compared to microaveraging [111]. As presented in Tables D.15 to D.19, the linear discriminant classifier based on GLS information performs very well for the categories with fewer positive training (and test) instances, yielding a better macroaveraged performance than the microaveraged one.

Table D.10 Compared supervised classifiers performances on the Reuters-21578 data set - earn category (1087 positive test instances).

(a) Logistic regression performances

logistic regression	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.8743	0.8896	**0.8819**	0.4563	0.1104	0.1778
$q = 2$	0.9197	0.9374	**0.9285**	0.9120	0.9154	0.9137
$q = 3$	0.9189	0.9384	0.9285	0.9307	0.9384	**0.9345**
$q = 4$	0.9273	0.9393	0.9333	0.9341	0.9393	**0.9367**
$q = 5$	0.9349	0.9374	0.9362	0.9401	0.9393	**0.9397**
$q = 6$	0.9436	0.9384	0.9410	0.9453	0.9374	**0.9413**
$q = 10$	0.9542	0.9577	0.9559	0.9560	0.9604	**0.9582**

(b) Linear discriminant performances

linear discriminant	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.8893	0.8868	**0.8881**	0.4483	0.3707	0.4058
$q = 2$	0.9682	0.8951	**0.9302**	0.9834	0.8712	0.9239
$q = 3$	0.9644	0.8960	0.9289	0.9907	0.8804	**0.9323**
$q = 4$	0.9709	0.8905	0.9290	0.9866	0.8822	**0.9315**
$q = 5$	0.9630	0.9108	**0.9362**	0.9768	0.8914	0.9322
$q = 6$	0.9631	0.9117	**0.9367**	0.9634	0.8970	0.9290
$q = 10$	0.9667	0.9338	**0.9499**	0.9592	0.9301	0.9444

(c) Naive Bayes performances

Naive Bayes	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.8942	0.8859	**0.8900**	0.4646	0.1086	0.1760
$q = 2$	0.9185	0.9227	**0.9206**	0.9341	0.8868	0.9099
$q = 3$	0.9085	0.9043	**0.9064**	0.8984	0.9034	0.9009
$q = 4$	0.8892	0.9006	**0.8949**	0.8875	0.8997	0.8936
$q = 5$	0.9004	0.9062	**0.9033**	0.9012	0.8979	0.8995
$q = 6$	0.9631	0.9117	**0.9367**	0.8621	0.9144	0.8875
$q = 10$	0.8930	0.9218	**0.9072**	0.8606	0.9144	0.8867

Table D.11 Compared supervised classifiers performances on the Reuters-21578 data set - acq category (719 positive test instances).

(a) Logistic regression performances

logistic regression	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.	0.	0.	0.7897	0.2559	0.3866
$q = 2$	0.8700	0.6606	**0.7510**	0.7703	0.2378	0.3634
$q = 3$	0.8756	0.7051	**0.7812**	0.8777	0.6690	0.7593
$q = 4$	0.8871	0.6996	**0.7823**	0.8839	0.6885	0.7740
$q = 5$	0.8858	0.7010	**0.7826**	0.8897	0.6843	0.7736
$q = 6$	0.8842	0.7010	**0.7820**	0.8907	0.6801	0.7713
$q = 10$	0.9242	0.7803	0.8462	0.9136	0.7942	**0.8497**

(b) Linear discriminant performances

linear discriminant	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.3547	0.6231	0.4521	0.6317	0.7538	**0.6874**
$q = 2$	0.7547	0.8387	**0.7945**	0.6189	0.7204	0.6658
$q = 3$	0.7892	0.8540	**0.8203**	0.8000	0.8401	0.8195
$q = 4$	0.7948	0.8456	**0.8194**	0.8048	0.8317	0.8181
$q = 5$	0.7945	0.8442	0.8186	0.8084	0.8331	**0.8205**
$q = 6$	0.7924	0.8442	**0.8175**	0.7995	0.8317	0.8153
$q = 10$	0.8695	0.8526	**0.8610**	0.8806	0.8414	0.8606

(c) Naive Bayes performances

Naive Bayes	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.	0.	0.	0.7778	0.2726	0.4037
$q = 2$	0.8453	0.7218	**0.7787**	0.7768	0.3825	0.5126
$q = 3$	0.8741	0.7149	0.7865	0.8587	0.7524	**0.8021**
$q = 4$	0.8740	0.7330	**0.7973**	0.8700	0.7260	0.7915
$q = 5$	0.8503	0.7427	**0.7929**	0.8552	0.6982	0.7688
$q = 6$	0.8048	0.7510	**0.7770**	0.8070	0.7093	0.7550
$q = 10$	0.8536	0.7705	**0.8099**	0.8154	0.7803	0.7974

Table D.12 Compared supervised classifiers performances on the Reuters-21578 data set - money category (179 positive test instances).

(a) Logistic regression performances

logistic regression	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.5652	0.1453	0.2311	0.6316	0.2011	**0.3051**
$q = 2$	0.5618	0.2793	**0.3731**	0.6000	0.2346	0.3373
$q = 3$	0.6875	0.4302	**0.5292**	0.5743	0.3240	0.4143
$q = 4$	0.7344	0.5251	**0.6124**	0.6957	0.3575	0.4723
$q = 5$	0.7939	0.5810	**0.6710**	0.7231	0.5251	0.6084
$q = 6$	0.7867	0.6592	**0.7173**	0.7481	0.5642	0.6433
$q = 10$	0.7826	0.7039	**0.7412**	0.7867	0.6592	0.7173

(b) Linear discriminant performances

linear discriminant	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.2009	1	0.3346	0.2791	0.5084	**0.3604**
$q = 2$	0.2027	1	0.3371	0.3324	0.6648	**0.4432**
$q = 3$	0.3296	0.6648	0.4407	0.3458	0.6704	**0.4563**
$q = 4$	0.3732	0.7151	**0.4904**	0.3443	0.5866	0.4339
$q = 5$	0.5325	0.6872	**0.6000**	0.4943	0.7263	0.5882
$q = 6$	0.6438	0.8380	**0.7282**	0.5230	0.6983	0.5981
$q = 10$	0.6827	0.7933	**0.7339**	0.6825	0.7207	0.7011

(c) Naive Bayes performances

Naive Bayes	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.2174	0.0279	0.0495	0.4854	0.2793	**0.3546**
$q = 2$	0.3654	0.4246	**0.3928**	0.3585	0.4246	0.3887
$q = 3$	0.6087	0.4693	**0.5300**	0.3750	0.7207	0.4933
$q = 4$	0.7077	0.5140	**0.5955**	0.4561	0.4358	0.4457
$q = 5$	0.6536	0.5587	**0.6024**	0.4758	0.6034	0.5320
$q = 6$	0.6564	0.5978	**0.6257**	0.4709	0.4525	0.4615
$q = 10$	0.4571	0.7151	**0.5577**	0.3333	0.5028	0.4009

Table D.13 Compared supervised classifiers performances on the Reuters-21578 data set - crude category (189 positive test instances).

(a) Logistic regression performances

logistic regression	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.	0.	0.	0.7701	0.3545	0.4855
$q = 2$	0.7692	0.4762	**0.5882**	0.7526	0.3862	0.5105
$q = 3$	0.8750	0.7037	**0.7801**	0.7928	0.4656	0.5867
$q = 4$	0.8742	0.6984	0.7765	0.8790	0.7302	**0.7977**
$q = 5$	0.8675	0.6931	0.7706	0.8790	0.7302	**0.7977**
$q = 6$	0.8675	0.6931	0.7706	0.8910	0.7354	**0.8058**
$q = 10$	0.8897	0.6825	0.7725	0.8621	0.7937	**0.8264**

(b) Linear discriminant performances

linear discriminant	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.0892	0.3280	0.1403	0.5946	0.6984	**0.6423**
$q = 2$	0.6712	0.7884	**0.7251**	0.5891	0.6296	0.6087
$q = 3$	0.6794	0.9418	**0.7894**	0.7682	0.6138	0.6824
$q = 4$	0.6794	0.9418	0.7894	0.8247	0.8466	**0.8355**
$q = 5$	0.6794	0.9418	0.7894	0.8247	0.8466	**0.8355**
$q = 6$	0.6794	0.9418	0.7894	0.8226	0.8095	**0.8160**
$q = 10$	0.8116	0.8889	**0.8485**	0.6795	0.9312	0.7857

(c) Naive Bayes performances

Naive Bayes	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.	0.	0.	0.6753	0.5503	0.6064
$q = 2$	0.7097	0.6984	**0.7040**	0.7133	0.5661	0.6313
$q = 3$	0.6910	0.8519	**0.7630**	0.7459	0.7302	0.7380
$q = 4$	0.6569	0.8307	**0.7336**	0.5982	0.7090	0.6489
$q = 5$	0.5639	0.7937	**0.6593**	0.5422	0.7143	0.6164
$q = 6$	0.5106	0.7619	**0.6115**	0.5116	0.6984	0.5906
$q = 10$	0.4828	0.9630	**0.6431**	0.4916	0.9259	0.6422

Table D.14 Compared supervised classifiers performances on the Reuters-21578 data set - grain category (149 positive test instances).

(a) Logistic regression performances

logistic regression	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.	0.	0.	0.	0.	0.
$q = 2$	0.	0.	0.	0.	0.	0.
$q = 3$	0.	0.	0.	0.	0.	0.
$q = 4$	0.	0.	0.	0.	0.	0.
$q = 5$	0.	0.	0.	0.	0.	0.
$q = 6$	0.	0.	0.	0.	0.	0.
$q = 10$	0.	0.	0.	0.	0.	0.

(b) Linear discriminant performances

linear discriminant	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.0615	0.5906	**0.1114**	0.0587	0.6779	0.1080
$q = 2$	0.0697	0.3020	0.1132	0.0654	0.7047	**0.1197**
$q = 3$	0.0717	0.3221	**0.1174**	0.0583	0.7517	0.1083
$q = 4$	0.1089	0.3289	**0.1636**	0.0595	0.6644	0.1093
$q = 5$	0.1079	0.3289	**0.1625**	0.1059	0.3154	0.1585
$q = 6$	0.1082	0.3289	**0.1628**	0.1047	0.3154	0.1572
$q = 10$	0.1185	0.3691	**0.1794**	0.1126	0.3423	0.1694

(c) Naive Bayes performances

Naive Bayes	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.	0.	0.	0.	0.	0.
$q = 2$	0.	0.	0.	0.	0.	0.
$q = 3$	0.	0.	0.	0.	0.	0.
$q = 4$	0.1818	0.0134	0.0250	0.	0.	0.
$q = 5$	0.2143	0.0201	0.0368	0.1852	0.0336	**0.0568**
$q = 6$	0.5000	0.0134	0.0261	0.1739	0.0268	**0.0465**
$q = 10$	0.2000	0.0134	0.0252	0.1379	0.0268	**0.0449**

Table D.15 Compared supervised classifiers performances on the Reuters-21578 data set - trade category (118 positive test instances).

(a) Logistic regression performances

logistic regression	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.4667	0.0593	**0.1053**	0.1667	0.0085	0.0161
$q = 2$	0.5714	0.1017	**0.1727**	0.5238	0.0932	0.1583
$q = 3$	0.5278	0.1610	**0.2468**	0.4762	0.0847	0.1439
$q = 4$	0.4848	0.1356	0.2119	0.4872	0.1610	**0.2420**
$q = 5$	0.5429	0.1610	**0.2484**	0.4872	0.1610	0.2420
$q = 6$	0.5429	0.1610	**0.2484**	0.5128	0.1695	**0.2548**
$q = 10$	0.6250	0.1695	0.2667	0.5789	0.1864	**0.2821**

(b) Linear discriminant performances

linear discriminant	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.1712	0.6356	**0.2698**	0.1587	0.6186	0.2526
$q = 2$	0.1980	0.6695	0.3056	0.2051	0.6186	**0.3080**
$q = 3$	0.2710	0.6017	**0.3737**	0.2065	0.5932	0.3063
$q = 4$	0.2703	0.5932	0.3714	0.3014	0.5339	**0.3853**
$q = 5$	0.2789	0.5932	0.3794	0.2920	0.5593	**0.3837**
$q = 6$	0.2789	0.5932	**0.3794**	0.2920	0.5593	**0.3837**
$q = 10$	0.2954	0.5932	**0.3944**	0.2974	0.5847	0.3943

(c) Naive Bayes performances

Naive Bayes	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.3158	0.2542	**0.2817**	0.2468	0.1610	0.1949
$q = 2$	0.2782	0.3136	0.2948	0.2797	0.3390	**0.3065**
$q = 3$	0.2217	0.4153	**0.2891**	0.2416	0.3051	0.2697
$q = 4$	0.2060	0.4661	**0.2857**	0.1789	0.4322	0.2531
$q = 5$	0.2030	0.4661	**0.2828**	0.1834	0.4492	0.2604
$q = 6$	0.2229	0.6271	**0.3289**	0.1693	0.4576	0.2471
$q = 10$	0.2071	0.6441	**0.3134**	0.2005	0.6441	0.3058

Table D.16 Compared supervised classifiers performances on the Reuters-21578 data set - interest category (131 positive test instances).

(a) Logistic regression performances

logistic regression	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.6250	0.0763	0.1361	0.5000	0.1527	**0.2339**
$q = 2$	0.6364	0.1069	0.1830	0.5870	0.2061	**0.3051**
$q = 3$	0.6316	0.2748	0.3830	0.6981	0.2824	**0.4022**
$q = 4$	0.6712	0.3740	**0.4804**	0.5902	0.2748	0.3750
$q = 5$	0.6829	0.4275	**0.5258**	0.5882	0.3053	0.4020
$q = 6$	0.7234	0.5191	**0.6044**	0.6034	0.2672	0.3704
$q = 10$	0.7188	0.5267	**0.6079**	0.7416	0.5038	0.6000

(b) Linear discriminant performances

linear discriminant	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.2077	0.7863	0.3285	0.3172	0.7023	**0.4371**
$q = 2$	0.1683	0.7939	0.2777	0.3100	0.7099	**0.4316**
$q = 3$	0.3448	0.8397	0.4889	0.3557	0.8092	**0.4942**
$q = 4$	0.3852	0.7939	**0.5187**	0.3732	0.7863	0.5061
$q = 5$	0.3810	0.7939	0.5149	0.4091	0.7557	**0.5308**
$q = 6$	0.2698	0.7557	0.3976	0.4091	0.7557	**0.5308**
$q = 10$	0.4737	0.8244	0.6017	0.4933	0.8473	**0.6236**

(c) Naive Bayes performances

Naive Bayes	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.7143	0.2290	0.3468	0.4674	0.3282	**0.3857**
$q = 2$	0.6731	0.2672	0.3825	0.4245	0.4504	**0.4370**
$q = 3$	0.5263	0.3817	0.4425	0.5079	0.4885	**0.4981**
$q = 4$	0.5357	0.5725	**0.5535**	0.4706	0.4275	0.4480
$q = 5$	0.5269	0.6718	**0.5906**	0.3757	0.4962	0.4276
$q = 6$	0.4947	0.7176	**0.5857**	0.3410	0.5649	0.4253
$q = 10$	0.4066	0.8473	**0.5495**	0.3676	0.7099	0.4844

Table D.17 Compared supervised classifiers performances on the Reuters-21578 data set - ship category (89 positive test instances).

(a) Logistic regression performances

logistic regression	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.	0.	0.	0.6471	0.1236	0.2075
$q = 2$	0.6000	0.0337	0.0638	0.6087	0.1573	**0.2500**
$q = 3$	0.8696	0.2247	0.3571	0.6842	0.2921	**0.4094**
$q = 4$	0.9000	0.5056	**0.6475**	0.7317	0.3371	0.4615
$q = 5$	0.8548	0.5955	**0.7020**	0.7368	0.4719	0.5753
$q = 6$	0.8649	0.7191	**0.7853**	0.7778	0.5506	0.6447
$q = 10$	0.8767	0.7191	**0.7901**	0.8553	0.7303	0.7879

(b) Linear discriminant performances

linear discriminant	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.0466	0.6404	0.0868	0.2031	0.7416	**0.3188**
$q = 2$	0.0711	1	0.1328	0.2510	0.7416	**0.3750**
$q = 3$	0.0789	0.9888	0.1461	0.2915	0.8090	**0.4286**
$q = 4$	0.0665	0.9775	0.1246	0.3480	0.8876	**0.5000**
$q = 5$	0.3304	0.8539	0.4765	0.3850	0.9213	**0.5430**
$q = 6$	0.6500	0.8764	**0.7464**	0.4686	0.9213	0.6212
$q = 10$	0.7419	0.7753	0.7582	0.7927	0.7303	**0.7602**

(c) Naive Bayes performances

Naive Bayes	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.	0.	0.	0.5690	0.3708	0.4490
$q = 2$	0.	0.	0.	0.5455	0.3371	0.4167
$q = 3$	0.	0.	0.	0.5172	0.1685	0.2542
$q = 4$	0.	0.	0.	0.6087	0.1573	0.2500
$q = 5$	0.3043	0.3933	**0.3431**	0.7143	0.1124	0.1942
$q = 6$	0.4153	0.8539	**0.5588**	0.5455	0.1348	0.2162
$q = 10$	0.2878	0.8989	0.4360	0.3284	0.7528	**0.4573**

Table D.18 Compared supervised classifiers performances on the Reuters-21578 data set - wheat category (71 positive test instances).

(a) Logistic regression performances

logistic regression	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.2000	0.0423	0.0698	0.2963	0.1127	**0.1633**
$q = 2$	0.6567	0.6197	**0.6377**	0.6596	0.4366	0.5254
$q = 3$	0.6471	0.6197	**0.6331**	0.6508	0.5775	0.6119
$q = 4$	0.6471	0.6197	**0.6331**	0.6615	0.6056	0.6324
$q = 5$	0.6615	0.6056	**0.6324**	0.6452	0.5634	0.6015
$q = 6$	0.6818	0.6338	**0.6569**	0.6500	0.5493	0.5954
$q = 10$	0.7397	0.7606	0.7500	0.7571	0.7465	**0.7518**

(b) Linear discriminant performances

linear discriminant	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.1567	0.6620	0.2534	0.1943	0.5775	**0.2908**
$q = 2$	0.5690	0.9296	**0.7059**	0.4754	0.8169	0.6010
$q = 3$	0.5739	0.9296	0.7097	0.6444	0.8169	**0.7205**
$q = 4$	0.5789	0.9296	0.7135	0.6744	0.8169	**0.7389**
$q = 5$	0.5603	0.9155	0.6952	0.6744	0.8169	**0.7389**
$q = 6$	0.5603	0.9155	0.6952	0.6444	0.8169	**0.7205**
$q = 10$	0.5752	0.9155	0.7065	0.6374	0.8169	**0.7160**

(c) Naive Bayes performances

Naive Bayes	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.1848	0.2394	0.2086	0.2330	0.3380	**0.2759**
$q = 2$	0.5664	0.9014	**0.6957**	0.3118	0.4085	0.3537
$q = 3$	0.5000	0.8451	**0.6283**	0.4403	0.8310	0.5756
$q = 4$	0.4014	0.8310	**0.5413**	0.3020	0.8592	0.4469
$q = 5$	0.3554	0.8310	**0.4979**	0.2300	0.6901	0.3451
$q = 6$	0.3245	0.8592	**0.4710**	0.2414	0.6901	0.3577
$q = 10$	0.2383	0.8592	**0.3731**	0.2259	0.8592	0.3578

Table D.19 Compared supervised classifiers performances on the Reuters-21578 data set - corn category (56 positive test instances).

(a) Logistic regression performances

logistic regression	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.	0.	0.	0.4286	0.2143	0.2857
$q = 2$	0.	0.	0.	0.4483	0.2321	0.3059
$q = 3$	0.5714	0.2857	**0.3810**	0.4231	0.1964	0.2683
$q = 4$	0.5208	0.4464	0.4808	0.5778	0.4643	**0.5149**
$q = 5$	0.5319	0.4464	0.4854	0.5652	0.4643	**0.5098**
$q = 6$	0.5306	0.4643	0.4952	0.5556	0.4464	**0.4950**
$q = 10$	0.5476	0.4107	0.4694	0.5854	0.4286	**0.4948**

(b) Linear discriminant performances

linear discriminant	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.0220	0.2500	0.0404	0.1542	0.6250	**0.2473**
$q = 2$	0.0900	0.5000	0.1526	0.1757	0.6964	**0.2806**
$q = 3$	0.2394	0.8036	**0.3689**	0.2009	0.7679	0.3185
$q = 4$	0.3659	0.8036	0.5028	0.4272	0.7857	**0.5535**
$q = 5$	0.3600	0.8036	0.4972	0.4175	0.7679	**0.5409**
$q = 6$	0.3600	0.8036	0.4972	0.4433	0.7679	**0.5621**
$q = 10$	0.4362	0.7321	0.5467	0.4490	0.7857	**0.5714**

(c) Naive Bayes performances

Naive Bayes	PCA Precision	PCA Recall	PCA F_1	GLS Precision	GLS Recall	GLS F_1
$q = 1$	0.	0.	0.	0.3143	0.3929	0.3492
$q = 2$	0.2581	0.2857	0.2712	0.3036	0.3036	**0.3036**
$q = 3$	0.3431	0.6250	**0.4430**	0.1732	0.3929	0.2404
$q = 4$	0.3388	0.7321	**0.4633**	0.2727	0.6429	0.3830
$q = 5$	0.3037	0.7321	**0.4293**	0.2000	0.6250	0.3030
$q = 6$	0.2971	0.7321	**0.4227**	0.1805	0.6607	0.2835
$q = 10$	0.2070	0.8393	**0.3322**	0.1273	0.6250	0.1026

Table D.20 Reuters-21578 data set: linear discriminant classification performances (micro- and macroaveraged) on the q-dimensional latent variable space learned with classical PCA and GLS with a Binomial distribution.

(a) Microaveraged performances

microav. linear discriminant	PCA Precision$^{\mu}$	PCA Recall$^{\mu}$	PCA F_1^{μ}	GLS Precision$^{\mu}$	GLS Recall$^{\mu}$	GLS F_1^{μ}
$q = 1$	0.2408	0.7306	0.3622	0.2845	0.5653	**0.3785**
$q = 2$	0.3704	0.8303	0.5123	0.4087	0.7665	**0.5331**
$q = 3$	0.4553	0.8296	**0.5880**	0.4239	0.8099	0.5565
$q = 4$	0.4709	0.8260	**0.5998**	0.4743	0.8128	0.5990
$q = 5$	0.6178	0.8275	**0.7075**	0.6233	0.7895	0.6966
$q = 6$	0.6265	0.8364	0.7164	0.6484	0.8056	**0.7185**
$q = 10$	0.6902	0.8415	**0.7584**	0.6881	0.8318	0.7532

(b) Macroaveraged performances

macroav. linear discriminant	PCA Precision^{M}	PCA Recall^{M}	PCA F_1^{M}	GLS Precision^{M}	GLS Recall^{M}	GLS F_1^{M}
$q = 1$	0.2200	0.6403	0.2905	0.3040	0.6274	**0.3751**
$q = 2$	0.3763	0.7717	0.4475	0.4006	0.7174	**0.4757**
$q = 3$	0.4342	0.7842	0.5184	0.4662	0.7552	**0.5267**
$q = 4$	0.4594	0.7820	0.5423	0.5138	0.7611	**0.5804**
$q = 5$	0.4988	0.7673	0.5870	0.5307	0.7386	**0.6007**
$q = 6$	0.5306	0.7809	**0.6150**	0.5471	0.7373	0.6134
$q = 10$	0.5971	0.7678	**0.6580**	0.5984	0.7531	0.6527

D.3 Abalone data set

The task is to predict the age of an abalone based on physical measurements. The data set information given by the UC Irvine machine learning repository goes as follows: "The age of [an] abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope – a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age. Further information, such as weather patterns and location (hence food availability) may be required to solve the prob-

Table D.21 Abalone data set: attributes description.

name	data type	measurement unit	description
sex	nominal		M, F and I (infant)
length	continuous	mm	longest shell measurement
diameter	continuous	mm	perpendicular to length
height	continuous	mm	with meat in shell
whole weight	continuous	grams	whole abalone
shucked weight	continuous	grams	weight of meat
viscera weight	continuous	grams	gut weight (after bleeding)
shell weight	continuous	grams	after being dried
rings (target)	integer		+1.5 gives the age in years

lem." The Abalone data set consists of 4177 instances with 8 attributes. Table D.21 presents the attributes name, data type, measurement unit and description. Table D.22 presents the number of instances for each number of rings.

The problem can be seen as either a continuous-value regression modeling problem [123, 124] or as a classification problem [125, 126]. The classification problem can aim to distinguish three classes (number of rings $= 1 - 8$, number of

Table D.22 Abalone data set: number of instances per number of rings.

rings	1	2	3	4	5	6	7	8	9	10	11	12	13	14
inst.	1	1	15	57	115	259	391	568	689	634	487	267	203	126
rings	15	16	17	18	19	20	21	22	23	24	25	26	27	29
inst.	103	67	58	42	32	26	14	6	9	2	1	1	2	1

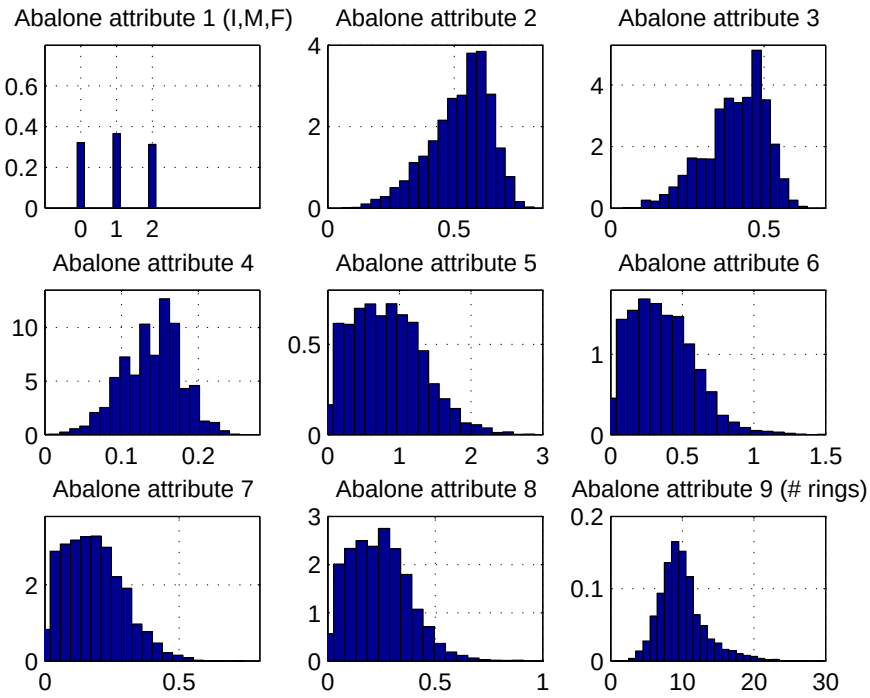

Figure D.9 Histograms performed on each attribute of the Abalone data set.

rings $= 9 - 10$, number of rings $= 11$ and higher) as in [125], or only two classes (number of rings $= 9$, number of rings $= 18$) as in [126]. The latest approach is interesting because one class has about sixteen times more instances than the second class.

We use this data set leaving out a randomly selected 40% of the instances to use as a test set (2506 training points and 1671 test points). Figure D.9 represents the histograms of the complete data set for each attribute. Note that there is one instance for which attribute 4 takes the value 1.13 (this abalone has 8 rings) and one instance for which attribute 4 takes the value 0.515 (this abalone has 10 rings); however, these instances were not considered for the three following histograms as they do not represent significative values for attribute 4.

Considering a three-class classification problem, Figure D.10 represents the histograms of each attribute for each class (class A: number of rings $= 1$ to 8, class B: number of rings $= 9$ and 10, and class C: number of rings $= 11$ and

higher).

Attribute 1 (sex) is the only noncontinuous attribute. We choose to model this attribute with a Binomial distribution (parameter $N = 2$). Figure D.11 presents a distribution fitting option for attribute 1.

We first choose a Binomial-Gaussian mixed-data assumption for the data set, with attribute 1 modeled as a Binomial variable and the other 7 attributes as Gaussian variables. Table D.23 presents the classification performances per class using a linear discriminant classifier on the classical PCA q-dimensional subspace learned in data space. Table D.24 presents the classification performances per class using a linear discriminant classifier on the latent q-dimensional variable subspace learned with GLS using a mixed Binomial-Gaussian distribution assumption. Table D.25 compares the micro- and macroaveraged classification performances corresponding to the results presented in Table D.23 and Table D.24. Performances are best when classification is performed on the GLS parameter subspace.

Then, we try to fit a distribution to the attributes 5, 6, 7 and 8. Possible distributions are the Weibull distribution, the Gamma distribution, the Beta distribution, the Chi-square distribution and the Non-central Chi-square distribution. The Beta distribution has a special constraint that the data should be greater than 0 and smaller than 1; only attributes 7 and 8 verify this constraint. Figure D.12, Figure D.13, Figure D.14 and Figure D.15 present distribution fitting options for attributes 5, 6, 7 and 8. The Gamma distribution is chosen as a good candidate to fit attributes 5, 6, 7 and 8.

We then choose a Binomial-Gaussian-Gamma mixed-data assumption for the data set, with attribute 1 modeled as a Binomial variable, attributes 2, 3 and 4 modeled as Gaussian variables and attributes 5, 6, 7 and 8 modeled as Gamma variables. Table D.26 presents the classification performances per class using a linear discriminant classifier on the latent q-dimensional variable subspace learned with GLS using a mixed Binomial-Gaussian-Gamma distribution assumption. Table D.27 compares the micro- and macroaveraged classification performances cor-

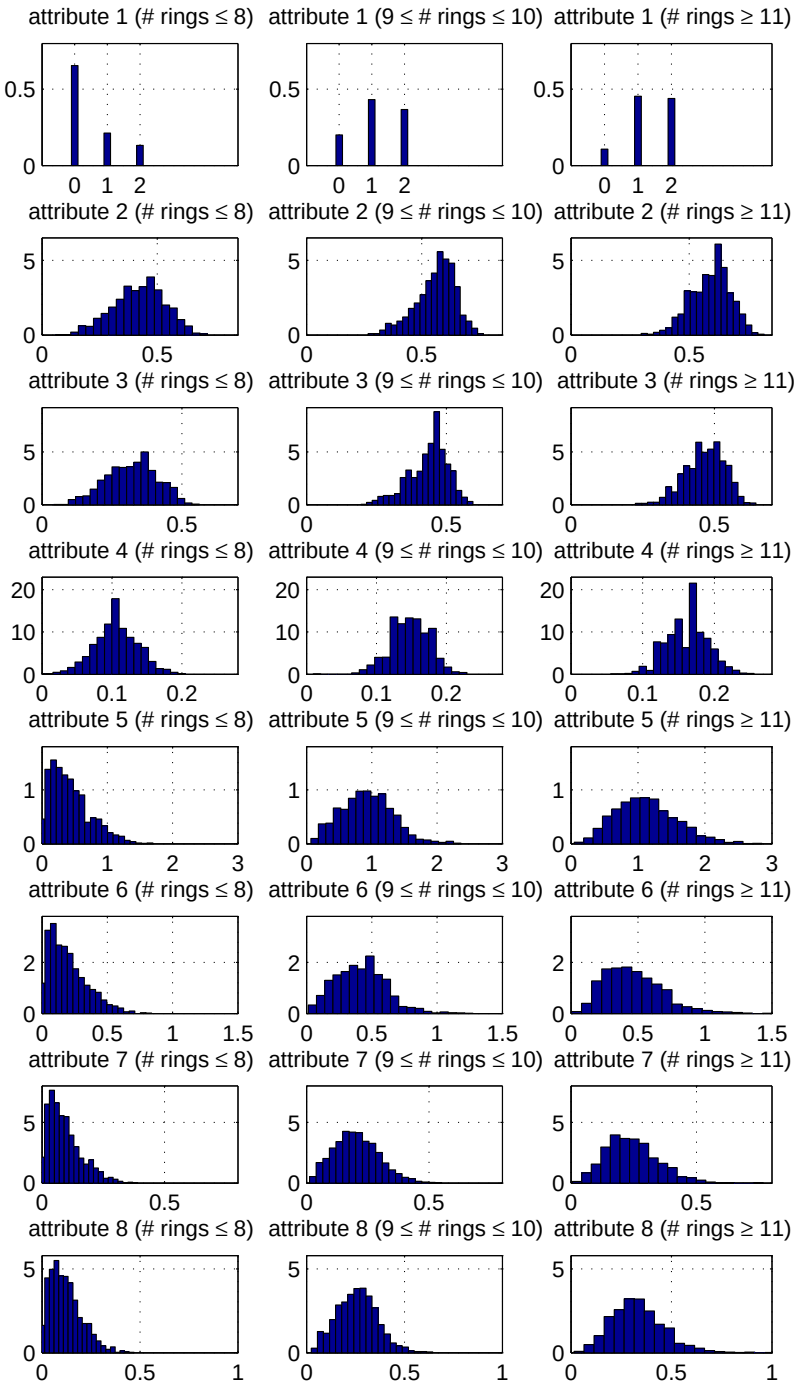

Figure D.10 Histograms performed separately on all Abalone data set attributes for a three-class classification problem (# rings ≤ 8, $9 \leq$ # rings ≤ 10 and # rings ≥ 11).

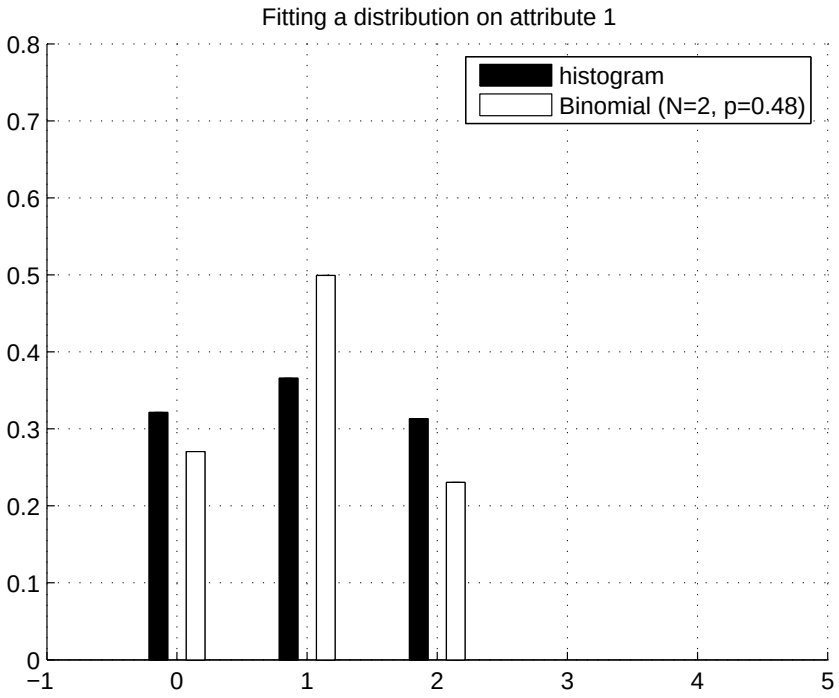

Figure D.11 Abalone data set: distribution fitting on attribute 1.

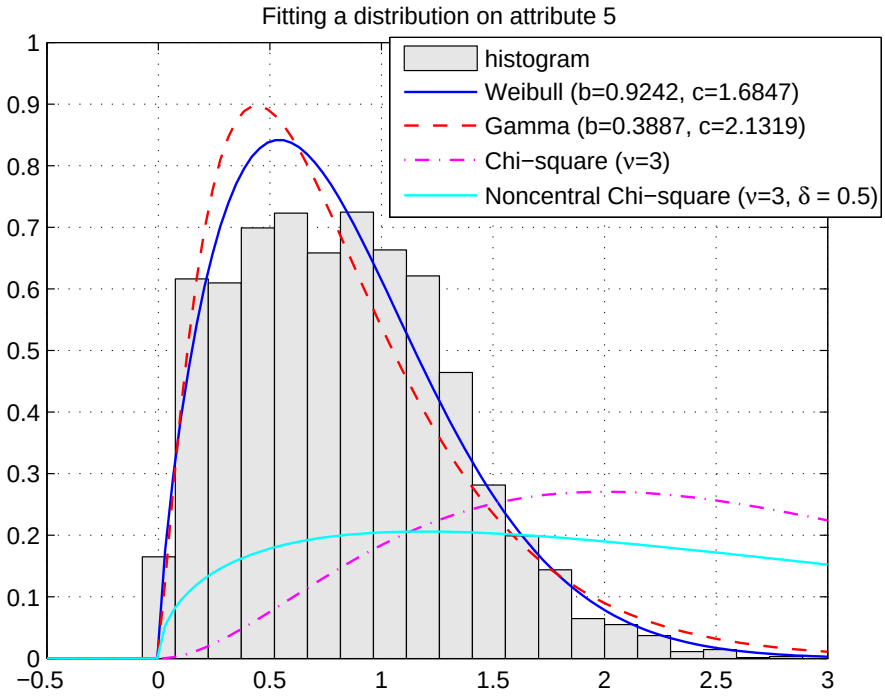

Figure D.12 Abalone data set: distribution fitting on attribute 5.

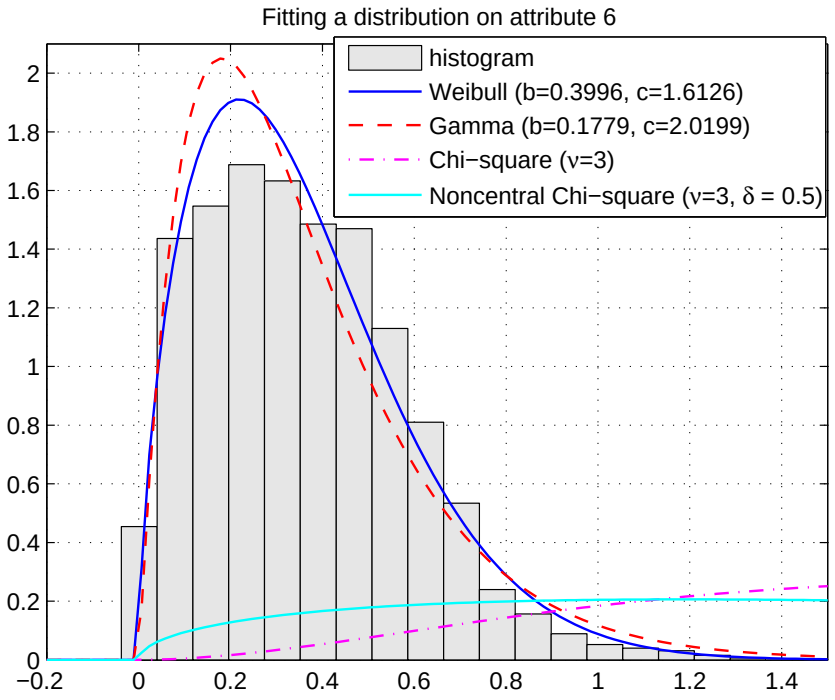

Figure D.13 Abalone data set: distribution fitting on attribute 6.

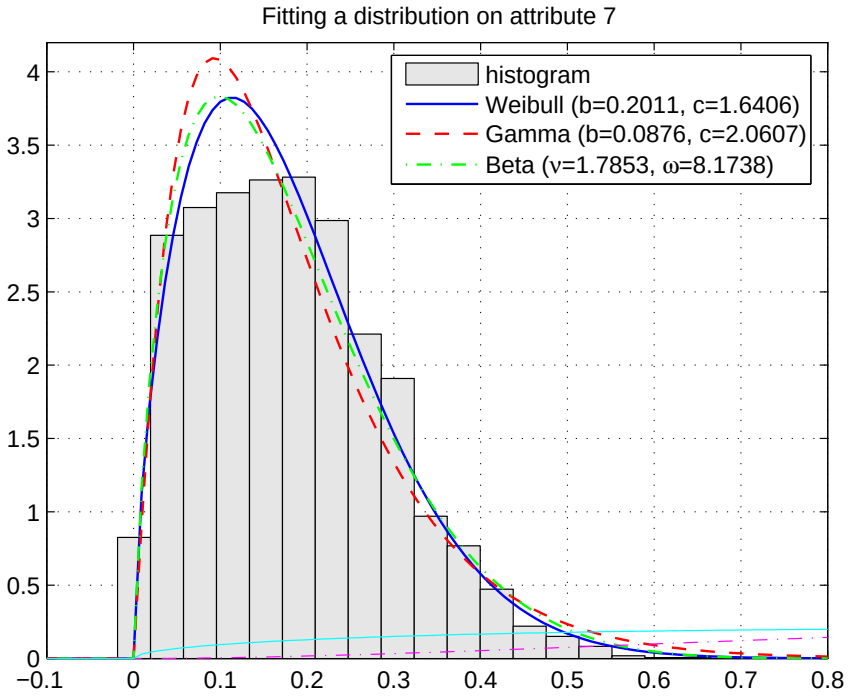

Figure D.14 Abalone data set: distribution fitting on attribute 7.

Table D.23 Abalone data set: linear discriminant classification performances on the q-dimensional latent variable space learned with classical PCA.

(a) Class A (number of rings = 1 to 8) performances

	Precision	Recall	F_1
$q = 1$	0.6815	0.7254	0.7028
$q = 2$	0.6876	0.7610	0.7224
$q = 3$	0.7293	0.7763	0.7521

(b) Class B (number of rings = 9 and 10) performances

	Precision	Recall	F_1
$q = 1$	0.4262	0.6779	0.5234
$q = 2$	0.4292	0.6957	0.5309
$q = 3$	0.4315	0.6441	0.5168

(c) Class C (number of rings = 11 and higher) performances

	Precision	Recall	F_1
$q = 1$	0.4535	0.7418	0.5629
$q = 2$	0.4557	0.7437	0.5652
$q = 3$	0.5393	0.7803	0.6378

responding to the results presented in Table D.23 and Table D.26. There are no statistically significant differences between the performances obtained with a mixed Binomial-Gaussian GLS assumption and performances obtained with a mixed Binomial-Gaussian-Gamma GLS assumption. As a conclusion, using a Gamma modeling assumption for the last four attributes was not useful to the linear discriminant classifier.

Table D.24 Abalone data set: linear discriminant classification performances on the q-dimensional latent variable space learned with GLS (Binomial-Gaussian distribution assumption).

(a) Class A (number of rings = 1 to 8) performances

	Precision	Recall	F_1
$q = 1$	0.6805	0.7508	0.7139
$q = 2$	0.7032	0.7712	0.7357
$q = 3$	0.6806	0.7441	0.7109

(b) Class B (number of rings = 9 and 10) performances

	Precision	Recall	F_1
$q = 1$	0.4283	0.7011	0.5317
$q = 2$	0.4433	0.7028	0.5437
$q = 3$	0.4192	0.5996	0.4934

(c) Class C (number of rings = 11 and higher) performances

	Precision	Recall	F_1
$q = 1$	0.4536	0.7726	0.5716
$q = 2$	0.4544	0.7399	0.5630
$q = 3$	0.4662	0.7437	0.5731

Acknowledgement

Appendix D, in part, is a reprint of the material as it will appear in "Classifying non-Gaussian and mixed data sets in their natural parameter space," C. Levasseur, U. Mayer and K. Kreutz-Delgado, in *Proceedings of the Nineteenth IEEE International Workshop on Machine Learning for Signal Processing*, Sept. 2009 as well as a reprint of the material as it appears in "Generalized statistical methods for mixed exponential families, part II: applications," C. Levasseur, U. F. Mayer and K. Kreutz-Delgado, submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Sept. 2009. The dissertation author was the primary author of these papers.

Table D.25 Abalone data set: linear discriminant classification performances (micro- and macroaveraged) on the q-dimensional latent variable space learned with classical PCA and GLS with a Binomial-Gaussian distribution.

(a) Microaveraged performances

	PCA Precision$^{\mu}$	PCA Recall$^{\mu}$	PCA F_1^{μ}	GLS Precision$^{\mu}$	GLS Recall$^{\mu}$	GLS F_1^{μ}
$q = 1$	0.5036	0.7120	0.5899	0.5043	0.7409	**0.6001**
$q = 2$	0.5085	0.7337	0.6007	0.5178	0.7385	**0.6088**
$q = 3$	0.5523	0.7331	**0.6300**	0.5103	0.6954	0.5887

(b) Macroaveraged performances

	PCA Precision^{M}	PCA Recall^{M}	PCA F_1^{M}	GLS Precision^{M}	GLS Recall^{M}	GLS F_1^{M}
$q = 1$	0.5204	0.7126	0.5952	0.5208	0.7415	**0.6058**
$q = 2$	0.5242	0.7335	0.6062	0.5337	0.7380	**0.6141**
$q = 3$	0.5667	0.7336	**0.6355**	0.5220	0.6958	0.5925

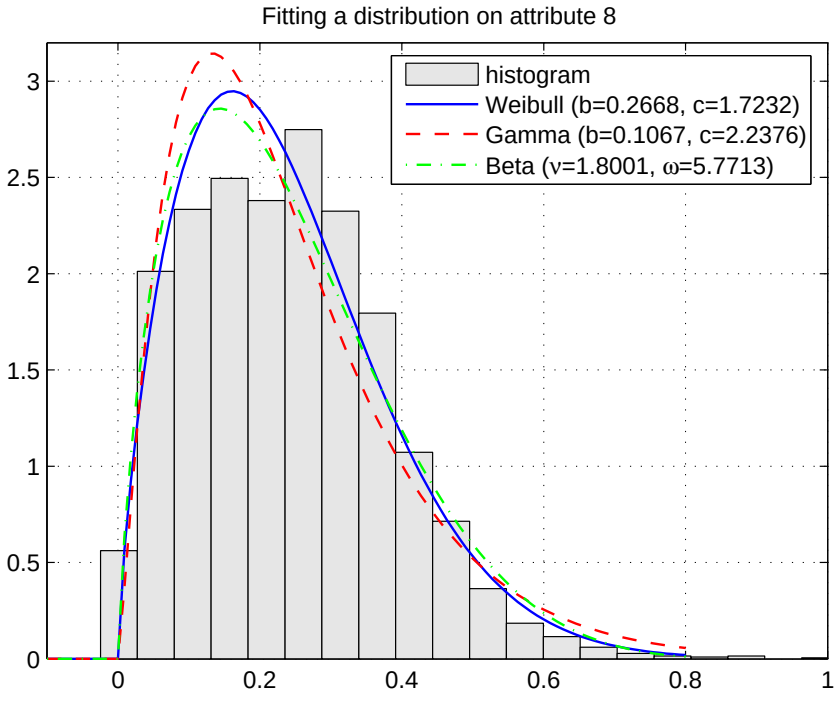

Figure D.15 Abalone data set: distribution fitting on attribute 8.

Table D.26 Abalone data set: linear discriminant classification performances on the q-dimensional latent variable space learned with GLS (Binomial-Gaussian-Gamma distribution assumption).

(a) Class A (number of rings = 1 to 8) performances

	Precision	Recall	F_1
$q = 1$	0.6799	0.7500	0.7133
$q = 2$	0.7077	0.7731	0.7390
$q = 3$	0.6875	0.7432	0.7143

(b) Class B (number of rings = 9 and 10) performances

	Precision	Recall	F_1
$q = 1$	0.4262	0.6964	0.5288
$q = 2$	0.4428	0.6982	0.5419
$q = 3$	0.4209	0.5993	0.4945

(c) Class C (number of rings = 11 and higher) performances

	Precision	Recall	F_1
$q = 1$	0.4535	0.7737	0.5718
$q = 2$	0.4539	0.7413	0.5630
$q = 3$	0.4644	0.7476	0.5729

Table D.27 Abalone data set: linear discriminant classification performances (micro- and macroaveraged) on the q-dimensional latent variable space learned with classical PCA and GLS with a Binomial-Gaussian-Gamma distribution.

(a) Microaveraged performances

	PCA Precision$^{\mu}$	PCA Recall$^{\mu}$	PCA F_1^{μ}	GLS Precision$^{\mu}$	GLS Recall$^{\mu}$	GLS F_1^{μ}
$q = 1$	0.5036	0.7120	0.5899	0.5037	0.7394	**0.5992**
$q = 2$	0.5085	0.7337	0.6007	0.5191	0.7382	**0.6096**
$q = 3$	0.5523	0.7331	**0.6300**	0.5119	0.6960	0.5899

(b) Macroaveraged performances

	PCA Precision^{M}	PCA Recall^{M}	PCA F_1^{M}	GLS Precision^{M}	GLS Recall^{M}	GLS F_1^{M}
$q = 1$	0.5204	0.7126	0.5952	0.5199	0.7400	**0.6046**
$q = 2$	0.5242	0.7335	0.6062	0.5348	0.7375	**0.6146**
$q = 3$	0.5667	0.7336	**0.6355**	0.5243	0.6967	0.5939

Bibliography

[1] D. J. Bartholomew and M. Knott, *Latent Variable Models and Factor Analysis*, vol. 7 of *Kendall's Library of Statistics*. Oxford University Press, New York, 2nd edition, 1999.

[2] A. Skrondal and S. Rabe-Hesketh, *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Interdisciplinary Statistics, Chapman and Hall/CRC, 2004.

[3] P. McCullagh and J. A. Nelder, *Generalized Linear Models*. Monographs on Statistics and Applied Probability 37, Chapman and Hall/CRC, London, 2nd edition, 1989.

[4] C. E. McCulloch and S. R. Searle, *Generalized, Linear and Mixed Models*. Wiley Series in Probability and Statistics, Wiley-Interscience, New York, 2001.

[5] L. Fahrmeir and G. Tutz, *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer Series in Statistics, Springer-Verlag, New York, 2nd edition, 2001.

[6] J. Gill, *Generalized Linear Models: A Unified Approach*. Quantitative Applications in the Social Sciences, Sage Publications, Thousand Oaks, California, 2001.

[7] C. R. Rao and H. Toutenburg, *Linear Models: Least Squares and Alternatives*. Springer Series in Statistics, Springer-Verlag, New York, 2nd edition, 1999.

[8] E. W. Frees, *Longitudinal and Panel Data: Analysis and Applications in the Social Sciences*. Cambridge University Press, New York, 2nd edition, 2004.

[9] H. S. Lynn and C. E. McCulloch, "Using principal component analysis and correspondence analysis for estimation in latent variable models," *Journal of the American Statistical Society*, vol. 95, no. 450, pp. 561–572, 2000.

[10] M. Collins, S. Dasgupta, and R. E. Schapire, "A generalization of principal components analysis to the exponential family," in *Advances in Neural Information Processing Systems*, vol. 14, 2001.

[11] D. B. Dunson and S. D. Perreault, "Factor analytic models of clustered multivariate data with informative censoring," *Biometrics*, vol. 57, pp. 302–308, 2001.

[12] D. B. Dunson, Z. Chen, and J. Harry, "A Bayesian approach for joint modeling of cluster size and subunit-specific outcomes," *Biometrics*, vol. 59, pp. 521–530, 2003.

[13] A. Skrondal and S. Rabe-Hesketh, "Some applications of generalized linear latent and mixed models in epidemiology," *Norsk Epidenmiologi*, vol. 13, no. 2, pp. 265–278, 2003.

[14] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *SIAM International Conference on Data Mining (SDM)*, 2004.

[15] Sajama and A. Orlitsky, "Semi-parametric exponential family PCA," *Advances in Neural Information Processing Systems*, vol. 17, 2004.

[16] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, 2005.

[17] N. Laird, "Nonparametric maximum likelihood estimation of a mixing distribution," *Journal of the American Statistical Society*, vol. 73, 1978.

[18] L. Simar, "Maximum likelihood estimation of a compound Poisson process," *The Annals of Statistics*, vol. 4, no. 6, pp. 1200–1209, 1976.

[19] N. P. Jewell, "Mixtures of exponential distributions," *The Annals of Statistics*, vol. 10, no. 2.

[20] B. G. Lindsay, "The geometry of mixture likelihoods: a general theory," *Annals of Statistics*, vol. 11, no. 1, pp. 86–94, 1983.

[21] B. G. Lindsay, "The geometry of mixture likelihoods, part ii: the exponential family," *Annals of Statistics*, vol. 11, no. 3, pp. 783–792, 1983.

[22] A. Mallet, "A maximum likelihood estimation method for random coefficient regression models," *Biometrika*, vol. 73, no. 3, pp. 645–656, 1986.

[23] A. Wald, "Estimation of a parameter when the number of unknown parameters incresases indefinitely with the number of observations," *Annals of Mathematical Statistics*, vol. 19, 1948.

[24] A. Wald, "Note on the consistency of the maximum likelihood estimate," *Annals of Mathematical Statistics*, vol. 20, 1949.

[25] J. Kiefer and J. Wolfowitz, "Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters," *The Annals of Mathematical Statistics*, vol. 27, pp. 887–906, 1956.

[26] A. Schumitzky, "Nonparametric EM algorithms for estimating prior distributions," *Applied Mathematics and Computation*, vol. 45, no. 2, pp. 143–157, 1991.

[27] M. Aitkin, "A general maximum likelihood analysis of overdispersion in generalized linear models," *Statistics and Computing*, vol. 6, pp. 251–262, 1996.

[28] M. Aitkin and I. Aitkin, "A hybrid EM/Gauss-Newton algorithm for maximum likelihood in mixture distributions," *Statistics and Computing*, vol. 6, pp. 127–130, 1996.

[29] M. Aitkin, "A general maximum likelihood analysis of variance components in generalized linear models," *Biometrics*, vol. 55, pp. 117–128, 1999.

[30] M. Aitkin, "Meta-analysis by random effect modelling in generalized linear models," *Statistics in Medicine*, vol. 18, pp. 2343–2351, 1999.

[31] D. J. Bartholomew, "The foundations of factor analysis," *Biometrika*, vol. 71, no. 2, pp. 221–232, 1984.

[32] I. Moustaki and M. Knott, "Generalised latent trait models," *Psychometrika*, vol. 65, no. 3, 2000.

[33] J. K. Vermunt, "Multilevel latent class models," *Sociological Methodology*, vol. 33, pp. 213–239, 2003.

[34] D. Boehning, *Computer-Assited Analysis of Mixtures and Applications: Meta-Analysis, Disease Mapping, and Others*. Monographs on Statistics and Applied Probability 81, Chapman and Hall/CRC, New York, 2000.

[35] G. F. Cooper, "Probabilistic inference using belief networks is NP-hard," *Technical report KSL 87-27, Stanford Knowledge Systems Laboratory*, 1987.

[36] D. M. Chickering, D. Heckerman, and C. Meek, "Large-sample learning of Bayesian networks is NP-hard," *Journal of Machine Learning Research*, vol. 5, pp. 1287–1330, 2004.

[37] C. M. Bishop, *Pattern Recognition and Machine Learning*. Information Science and Statistics, Springer, 2006.

[38] C. E. McCulloch, *Generalized Linear Mixed Models*. Institute of Mathematical Statistics, Hayward, California, 2003.

[39] N. Roy, G. Gordon, and S. Thrun, "Finding approximate POMDP solutions through belief compression," *Journal of Artificial Intelligence Research*, vol. 23, pp. 1–40, 2005.

[40] L. Cayton, "Fast nearest neighbor retrieval for Bregman divergences," in *Twenty-Fifth International Conference on Machine Learning (ICML)*, 2008.

[41] A. Asuncion and D. Newman, "UCI machine learning repository, http://www.ics.uci.edu/∼mlearn/MLRepository.html." University of California, Irvine, School of Information and Computer Sciences, 2007.

[42] A. I. Khuri, B. Mukherjee, B. K. Sinha, and M. Ghosh, "Design issues for generalized linear models: a review," *Journal of Statistical Science*, vol. 21, no. 3, pp. 376–399, 2006.

[43] J. A. Hagenaars, *Loglinear Models with Latent Variables*. Quantitative Applications in the Social Sciences, Sage Publications, Thousand Oaks, California, 1993.

[44] L. G. Grimm and P. R. Yarnold, *Reading and Understanding Multivariate Statistics*. American Psychological Association, 1st edition, 1995.

[45] D. N. Lawley and A. E. Maxwell, *Factor Analysis as a Statistical Method*. New York, American Elsevier Pub. Co., 1971.

[46] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, pp. 417–441, 1933.

[47] I. T. Jolliffe, *Principal Component Analysis*. Springer Series in Statistics, Springer-Verlag, New York, 2nd edition, 2002.

[48] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, Sixth Series*, no. 2, pp. 559–572, 1901.

[49] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience, 2nd edition, 2001.

[50] T. W. Anderson and H. Rubin, "Statistical inference in factor analysis," *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, pp. 111–150, 1956.

[51] S. Roweis, "EM algorithms for PCA and SPCA," *Advances in Neural Information Processing Systems*, vol. 10, 1998.

[52] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society, Series B*, vol. 61, no. 3, pp. 611–622, 1999.

[53] G. McLachlan and D. Peel, *Finite Mixture Models*. Wiley Series in Probability and Statistics, Wiley-Interscience, New York, 2000.

[54] O. E. Barndorff-Nielsen, *Information and Exponential Families in Statistical Theory*. Wiley Series in Probability and Mathematical Statistics, John Wiley and Sons, 1978.

[55] L. D. Brown, *Fundamentals of Statistical Exponential Families: with Applications in Statistical Decision Theory*. Institute of Mathematical Statistics, 1986.

[56] S. Amari and H. Nagaoka, *Methods of Information Geometry*, vol. 191 of *Translations of Mathematical Monographs*. American Mathematical Society and Oxford University Press, 2001.

[57] K. S. Azoury and M. K. Warmuth, "Relative loss bounds for on-line density estimation with the exponential family of distributions," *Machine learning*, vol. 43, pp. 211–246, 2001.

[58] Y. Lee and J. A. Nelder, "Hierarchical generalized linear models," *Journal of the Royal Statistical Society*, vol. 58, no. 4, pp. 619–678, 1996.

[59] P. J. Green, "Hierarchical generalized linear models," *Journal of the Royal Statistical Society, B*, vol. 58, no. 4, pp. 619–678, 1996.

[60] M. Aitkin and R. Rocci, "A general maximum likelihood analysis of measurement error in generalized linear models," *Statistics and Computing*, vol. 12, pp. 163–174, 2002.

[61] M. Aitkin, "Longitudinal analysis of repeated binary data using autoregressive and random effect modelling," *Statistical Modelling*, vol. 3, pp. 291–303, 2003.

[62] M. I. Jordan and T. J. Sejnowski, *Graphical Models: Foundations of Neural Computation*. Computational Neuroscience, The MIT Press, 1st edition, 2001.

[63] K. P. Murphy, "An introduction to graphical models," *Technical reports/informal notes, http://www.cs.ubc.ca/ murphyk/mypapers.html*, 2001.

[64] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *UC Berkeley, Dept. of Statistics, Technical Report*, 2003.

[65] M. J. Wainwright and M. I. Jordan, "A variational principle for graphical models," *Chapter 11 in New Directions in Statistical Signal Processing: From Systems to Brains (Neural Information Processing)*, pp. 21–93, 2005.

[66] B. G. Lindsay, *Mixture Models: Theory, Geometry, and Applications*. Institute of Mathematical Statistics, New York, 1995.

[67] B. G. Lindsay and M. L. Lesperance, "A review of semiparametric mixture models," *Journal of Statistical Planning and Inference*, vol. 47, pp. 29–99, 1995.

[68] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood estimation from incomplete data via the EM algorithm," *Journal of Royal Statistical Society, B*, vol. 39, pp. 1–38, 1977.

[69] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, 1984.

[70] D. Boehning, P. Schlattmann, and B. Lindsay, "Computer-assisted analysis of mixtures (C.A.MAN): statistical algorithms," *Biometrics*, vol. 48, pp. 283–303, 1992.

[71] D. Boehning, E. Dietz, and P. Schlattmann, "Recent developments in computer-assisted analysis of mixtures," *Biometrics*, vol. 54, pp. 525–536, 1998.

[72] R. S. Pilla and B. Lindsay, "Alternative EM methods for nonparametric finite mixture models," *Biometrika*, vol. 88, no. 2, pp. 535–550, 2001.

[73] D. Boehning and W. Seidel, "Editorial: recent developments in mixture models," *Computational Statistics and Data Analysis*, vol. 41, pp. 349–357, 2003.

[74] D. Boehning, "The EM algorithm with gradient function update for discrete mixtures with known (fixed) number of components," *Statistics and Computing*, vol. 13, pp. 257–265, 2003.

[75] S. Rabe-Hesketh, A. Pickles, and C. Taylor, "GLLAMM: A general class of multilevel models and a Stata program," *Multilevel Modelling Newsletter*, vol. 13, pp. 17–23, 2001.

[76] E. L. Lehmann and G. Castella, *Theory of Point Estimation*. Springer Texts in Statistics, Springer, 2nd edition, 1998.

[77] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Computer Science and Scientific Computing Series, Academic Press, Boston, 2nd edition, 1990.

[78] A. J. Miller, *Subset Selection in Regression*. Monographs on Statistics and Applied Probability 95, Chapman and Hall/CRC, New York, 2nd edition, 1990.

[79] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics, Springer-Verlag, New York, 2001.

[80] M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*. Wiley-Interscience, 2nd edition, 1993.

[81] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Prentice Hall Signal Processing Series, Prentice Hall, 1993.

[82] Sajama, *Nonparametric Methods for Learning from Data*. Ph.D. thesis, University of California, San Diego, 2006.

[83] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra Book and Solutions Manual*. Society for Industrial and Applied Mathematics (SIAM); Package edition, 2001.

[84] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, Belmont, Massachusetts, 2003.

[85] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambrigde University Press, 2004.

[86] H. Hindi, "A tutorial on convex optimization II: duality and interior point methods," *Proceedings of the 2006 American Control Conference*, pp. 686–696, 2006.

[87] J. Forster and M. K. Warmuth, "Relative expected instantaneous loss bounds," *Journal of Computer and System Sciences*, vol. 64, pp. 76–102, 2002.

[88] J. Lafferty, S. Della Pietra, and V. Della Pietra, "Statistical learning algorithm based on Bregman distances," *Proceedings of 1997 Canadian Workshop on Information Theory*, pp. 77–80, 1997.

[89] S. Della Pietra, V. Della Pietra, and J. Lafferty, "Duality and auxiliary functions for Bregman distances," *Technical Report CMU-CS-01-109, School of Computer Science, CMU*, 2001.

[90] H. H. Bauschke and A. S. Lewis, "Dykstra's algorithm with Bregman projections: a convergence proof," *Optimization*, vol. 48, pp. 409–427, 2000.

[91] H. H. Bauschke, "Duality for Bregman projections onto translated cones and affine subspaces," *Journal of Approximation Theory*, vol. 121, pp. 1–12, 2003.

[92] H. H. Bauschke, J. M. Borwein, and P. L. Combettes, "Bregman monotone optimization algorithms," *SIAM Journal on Control and Optimization*, vol. 42, no. 2, pp. 596–636, 2003.

[93] D. R. Cox, "Some remarks on overdispersion," *Biometrika*, vol. 70, pp. 269–274, 1983.

[94] D. D. Dey, A. E. Gelfand, and F. Peng, "Overdispersed generalized linear models," *Journal of Statistical Planning and Inference*, vol. 64, pp. 93–107, 1997.

[95] K. Kreutz-Delgado and B. D. Rao, "FOCUSS-based dictionary learning algorithms," *Proceedings of the SPIE Volume 4119: Wavelet Applications in Signal and Image Processing VIII*, vol. 4119–53, 2000.

[96] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Computation*, vol. 15, no. 2, pp. 349–396, 2003.

[97] J. F. Murray and K. Kreutz-Delgado, "Visual recognition and inference using dynamic overcomplete sparse learning," *Neural Computation*, vol. 19, no. 9, pp. 2301–2352, 2007.

[98] A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet, "A direct formulation for sparse PCA using semidefinite programming," *Advances in Neural Information Processing Systems*, vol. 17, 2004.

[99] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 262–286, 2006.

[100] C. Levasseur, "Topics in unsupervised learning," *private lecture notes for the class CSE 291 taught by S. Dasgupta at UCSD*, 2004.

[101] H. Cramér, *Mathematical Methods of Statistics*. Princeton Landmarks in Mathematics, Princeton University Press, Nineteenth Edition, 1999.

[102] L. M. Bregman, "The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, pp. 200–217, 1967.

[103] F. Nielsen, J.-D. Boissonnat, and R. Nock, "On Bregman Voronoi diagrams," *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.

[104] F. Nielsen, J.-D. Boissonnat, and R. Nock, "On visualizing Bregman Voronoi diagrams," *Proceedings of the 23rd ACM Conference on Computational Geometry*, 2007.

[105] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley Series in Telecommunications, Wiley-Interscience, 1991.

[106] F. Pereira, N. Tishby, and L. Lee, "Distributional clustering of English words," *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 183–190, 1993.

[107] L. Lee, "Measures of distributional similarity," *Proceedings of the 37st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 25–32, 1999.

[108] M.-J. Nederhof and G. Satta, "Kullback-Leibler distance between probabilistic context-free grammars and probabilistic finite automata," *Proceedings of 20th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 71–77, 2004.

[109] M.-J. Nederhof and G. Satta, "Estimation of consistent probabilistic context-free grammars," *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 343–350, 2006.

[110] A. Agresti, *Categorical Data Analysis*. Wiley Series in Probability and Mathematical Statistics, Wiley-Interscience, 1990.

[111] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[112] A. Özgür, *Supervised and Unsupervised Machine Learning Techniques for Text Document Categorization*. M.S. thesis, Computer Engineering, Bogazici University, Istanbul, Turkey, 2004.

[113] G. Salton, *Introduction to Modern Information Retrieval*. Computer Science Series, McGraw-Hill, 1983.

[114] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," *Information Processing and Nanagement*, vol. 24, no. 5, pp. 513–523, 1988.

[115] D. Hiemstra, "A probabilistic justification for using the tf×idf term weighting in information retrieval," *International Journal on Digital Libraries*, 2000.

[116] A. Özgür and T. Güngör, "Classification of skewed and homogenous document corpora with class-based and corpus-based keywords," *Lecture Notes in Artificial Intelligence*, vol. 4314, pp. 91–101, 2006.

[117] F. Fleuret, "Fast binary feature selection with conditional mutual information," *Journal of Machine Learning Research*, vol. 5, pp. 1531–1555, 2004.

[118] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy, "SVM and Kernel methods Matlab toolbox." Perception Systems and Information, INSA de Rouen, Rouen, France, 2005.

[119] R. Yan, "MATLABArsenal, a MATLAB package for classification algorithms." Informedia, School of Computer Science, Carnegie Mellon University, 2006.

[120] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: a new benchmark collection for text categorization research," *Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[121] F. Debole and F. Sebastiani, "An analysis of the relative hardness of Reuters-21578 subsets," *Journal of the American Society for Information Science and Technology*, vol. 56, pp. 584–596, 2005.

[122] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[123] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[124] C. C. Aggarwal and P. S. Yu, "A condensation approach to privacy preserving data mining," *Lecture Notes in Computer Science*, vol. 2992, pp. 183–199, 2004.

[125] D. Clark, "Using feature trimming to improve the performance of Dystal," *Proceedings of the Third International Conference on Knowledge-Based Intelligent Information Engineering Systems*, 1999.

[126] H. Guo and H. L. Viktor, "Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach," *ACM SIGKDD Explorations Newsletter*, pp. 30–39, 2004.