Carl von Ossietzky Universität Oldenburg

DISSERTATION

Non-Linear Latent Variable Models for Inference and Learning from Non-Gaussian Data

by

Hamid Mousavi

born on February 3, 1988 in Mashhad, Iran

This thesis has been accepted in fulfillment of the requirements
for the degree of Doctor of Natural Sciences in
Faculty V - Mathematics and Science

Supervisor:
Prof. Dr. Jörg Lücke

Other assessors:
Prof. Dr. Asja Fischer
Prof. Dr. Alexander Hartmann

October 12, 2021

Declaration of Authorship

I, Hamid MOUSAVI, declare that this thesis titled, "Non-Linear Latent Variable Models for Inference and Learning from Non-Gaussian Data" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: *S. Hamid Mousavi*

Date: 12.10.2021

"*The secrets eternal neither you know nor I*
And answers to the riddle neither you know nor I
Behind the veil there is much talk about us, why
When the veil falls, neither you remain nor I."

From **Rubaiyat of Omar Khayyam** translated by Edward FitzGerald (1809–1883).

Omar Khayyam (1048–1131) was a Persian polymath, philosopher, mathematician, astronomer and poet. As a mathematician, he is most notable for his work on the classification and solution of *cubic equations*, where he provided geometric solutions by the intersection of *conics*. Khayyam also contributed to the understanding of the parallel axiom. As an astronomer, he designed the *Jalali calendar*, a solar calendar with a very precise 33-year intercalation cycle which provided the basis for the Persian calendar that is still in use after nearly a millennium ("Omar Khayyam," from Wikipedia, n.d.).

CARL VON OSSIETZKY UNIVERSITÄT OLDENBURG

Abstract

Faculty V – Mathematics and Science

Doctor of Natural Sciences

Non-Linear Latent Variable Models for Inference and Learning from Non-Gaussian Data

by Hamid MOUSAVI

Latent Variable Models (LVMs) are well established tools to accomplish a range of different data processing tasks. Applications exploit the ability of LVMs to identify latent data structure in order to improve data (e.g., through denoising) or to estimate the relation between latent causes and measurements in medical or other complex datasets. Examples of the frequently used LVMs are Factor Analysis (FA) and probabilistic Sparse Coding (SC) which assume weighted linear summation of the latents to determine the mean of a Gaussian distribution for the observables. In many cases, however, observables do not follow a Gaussian distribution, and a linear superposition model may (for many types of data) not be closely aligned with the true data generating process (even for Gaussian observables). Therefore, we here ask how these two assumptions can be modified in the core aspect of our purposes which is analyzing medical and other challenging datasets. In this direction, we propose a family of probabilistic generative models that encompasses a variety of probability distributions (including Gaussian, Gamma, Beta, Poisson and many more) from the exponential family. In addition, we investigate a point-wise maximum function and introduce a novel non-linear superposition for coupling the latents and observables using two matrices (if the considered noise distribution has two parameters): One to model the component means and another for component variances. We further exploit the Expectation Maximization (EM) algorithm and show that the presented link function allows for the derivation of a very general and concise set of parameter update equations. Concretely, we derive a set of updates that have the same functional form for all regular distributions of the exponential family. Our results then provide directly applicable learning equations for commonly as well as for unusually distributed data.

Moreover, to assess the reliability of our theoretical findings, we consider different applications of the proposed generative models and investigate a variety of complex datasets including both synthetic and real data. As real datasets, we study natural images, acoustic and image data and importantly medical data of hearing impairments. A substantial application of LVMs is their usage in Medicine for the inference of causes (latents) from disease symptoms (observables). Current LVMs for disease estimation are mainly based on noisy-OR Bayes nets which assume binary observables for symptoms. Instead, we here show that the usage of a Beta distribution (as one example of the distributions investigated in this thesis) can generalize the standard noisy-OR Bayes nets by encoding continuous observables, i.e., variables that model symptom severity in an interval from healthy to pathological. Our experimental results reveal that such extra information enables the model to produce reliable results in estimating the responsible causes for unseen data. We further leverage variational approximations to provide large-scale applicability of the proposed optimization algorithms. Consequently, the proposed models are efficient and scalable that can, in some cases, improve on previously used approaches. For instance, with the use of a Poisson distribution, we illustrate high-quality denoising results that can compete with other state-of-the-art approaches.

Zusammenfassung

Latent Variable Models (LVMs) sind etablierte Werkzeuge, um eine Reihe verschiedener Datenverarbeitungsaufgaben zu erfüllen. Anwendungen nutzen die Fähigkeit von LVMs, latente Datenstrukturen zu identifizieren, um Daten zu verbessern (z. B. durch Rauschunterdrückung) oder um den Zusammenhang zwischen latenten Ursachen und Messwerten in medizinischen oder anderen komplexen Datensätzen abzuschätzen. Beispiele häufig verwendeter LVMs sind Factor Analysis (FA) und probabilisches Sparse Coding (SC), die eine gewichtete lineare Summation der Latenten annehmen, um den Mittelwert einer Gauß-Verteilung für die Observablen zu bestimmen. In vielen Fällen folgen Observablen jedoch keiner Gaußschen Verteilung und ein lineares Superpositionsmodell kann zur Modellierung des wahren Datenerzeugungsprozess für viele Datentypen (selbst für Gaußsche Observablen) ungeeignet sein. Daher fragen wir hier im Kernaspekt dieser Arbeit, der Analyse medizinischer und anderer anspruchsvoller Datensätze, wie diese beiden Annahmen modifiziert werden können. Vor dem Hintergrund dieser Fragestellung schlagen wir eine Familie von probabilistischen generativen Modellen vor, die eine Vielzahl von Wahrscheinlichkeitsverteilungen (einschließlich u.a. Gauß, Gamma, Beta, Poisson) aus der exponentiellen Familie umfassen. Darüber hinaus untersuchen wir eine punktweise Maximumfunktion und führen, für 2-parametrige Rauschverteilungen, eine neuartige nichtlineare Superposition zur Kopplung der Latenten und Observablen mit zwei Matrizen ein: Eine zur Modellierung der Komponentenmittelwerte und eine andere für Komponentenvarianzen . Wir nutzen den Expectation Maximization (EM) Algorithmus und zeigen, dass die vorgestellte Linkfunktion die Herleitung eines sehr allgemeinen und präzisen Satzes von Parameteraktualisierungsgleichungen ermöglicht. Konkret leiten wir eine Reihe von Gleichungen her, die für alle regulären Verteilungen der Exponentialfamilie dieselbe funktionale Form haben. Unsere Ergebnisse liefern direkt anwendbare Lerngleichungen sowohl für gewöhnlich als auch für ungewöhnlich verteilte Daten.

Um die Verlässlichkeit unserer theoretischen Ergebnisse zu beurteilen, betrachten wir außerdem verschiedene Anwendungen der vorgeschlagenen generativen Modelle und untersuchen eine Vielzahl komplexer Datensätze, die sowohl synthetische als auch reale Daten umfassen. Als reale Datensätze untersuchen wir visuelle, akustische und insbesondere medizinische Daten zu Gehörerkrankungen. Eine wesentliche Anwendung von LVMs ist ihre Verwendung in der Medizin zur Modellierung von Ursachen (latente Variablen) aus Krankheitssymptomen (beobachtete Variablen). Aktuelle LVMs zur Krankheitsabschätzung basieren hauptsächlich auf Noisy-OR-Bayes-Netzen, die binäre Observablen für Symptome annehmen. Stattdessen zeigen wir hier, dass die Verwendung einer Beta-Verteilung (als ein Beispiel für die in dieser Arbeit untersuchten Verteilungen) die standardmäßigen Noisy-OR-Bayes-Netze verallgemeinern kann, indem Observablen als kontinuierliche Variablen zur Beschreibung der Symptomschwere in einem Intervall von gesund zu pathologischen, modelliert werden. Unsere experimentellen Ergebnisse zeigen, dass diese zusätzlichen Informationen es dem Modell ermöglichen, zuverlässige Ergebnisse bei der Schätzung verantwortlicher Ursachen für beobachtete Symptome zu erzielen. Wir nutzen außerdem Variationsapproximationen, um eine großräumige Anwendbarkeit der vorgeschlagenen Optimierungsalgorithmen zu ermöglichen. Die vorgeschlagenen Modelle sind effizient und skalierbar und können, in einigen Fällen, frühere Ansätze verbessern. Beispielsweise veranschaulichen wir mit der Poisson-Verteilung qualitativ hochwertige Entrauschungsergebnisse, die mit anderen modernen Ansätzen konkurrieren können.

Acknowledgements

I would like to thank all who have been supporting me towards my studies. Importantly, I would like to sincerely express my appreciation towards my professors, family and friends who have encouraged me to continue my studies and have favored me during my educational life. I specifically thank my supervisor, Prof. Dr. Jörg Lücke, for his constant support and fruitful comments during my career that intuitively helped me to leverage my experiences and master new skills in order to find the most appropriate path for my investigations. His encouraging ideas have also motivated me to find my interests in this field and step forward to find my personal goals which I appreciate the most. Moreover, I would like to thank all my colleagues who have always been available for interesting discussions and helped me to better integrate into the new community. Further, special thanks goes to my lovely wife, Ati, who has been always supportive and encouraging, and accompanied me during this exciting journey; and also my family and my family-in-law who kindly supported me throughout my entire life. Finally, I would like to thank the University of Oldenburg and its kind staff for the great opportunity and for the excellent and outstanding level of academic education.
In addition, I gratefully acknowledge funding by the German Research Foundation (DFG) in projects 390895286 (cluster of excellence H4a 2.0, EXC 2177/1) and also by the German Ministry of Research and Education (BMBF) in project 05M2020 (SPAplus, TP3). Further, I acknowledge support by the HPC Cluster CARL of Oldenburg University and by the HLRN network of HPC clusters (project nim00006).

Contents

Chapter 1

Introduction

The content of this chapter is mainly based on our papers (Mousavi et al., 2020) (currently under review) and (Mousavi et al., 2021). The idea of using medical data as a motivation example has been suggested by Jörg Lücke which I further developed and prepared the details. Later, the detailed discussion and interpretation of this example (which also appears in these two papers) have been jointly developed by Jörg Lücke and me. Also, the literature review section has been written jointly by Jörg Lücke and me.

1.1 Motivation

The rapid development of technology has drastically affected all areas of science including machine learning and data science. These new technologies, in terms of both software and hardware developments, have specially resulted in storage and analysis of big data which are nowadays available to researchers. Subsequently, we have observed a sustainable growth of machine learning approaches capable of training such big data (mainly in an unsupervised manner) in order to extract rich information and recognize patterns. References in this direction can be made to the works of authors who, amongst others, established or exploited a machine learning algorithm for a certain dataset such as (Olshausen and Field, 1996a; Olshausen and Field, 1997; Puertas, Bornschein, and Lücke, 2010; Bornschein, Henniges, and Lücke, 2013) for natural images, (Rotmensch et al., 2017; Shen et al., 2018; Shickel et al., 2017; Miotto et al., 2016; Rajkomar et al., 2018) for electronic medical records, (Dai and Lücke, 2014; Lee et al., 2009) for text documents, (Salmon et al., 2014; Giryes and Elad, 2014; Rond, Giryes, and Elad, 2016; Sheikh, Shelton, and Lücke, 2014; Haft, Hofman, and Tresp, 2004; Goodfellow, Courville, and Bengio, 2012b) for images, (Roweis, 2003; Lücke and Sahani, 2008; Sheikh et al., 2019; Bornschein and Lücke, 2009) for acoustic data and many more. In fact, we take the assumption that all these complex datasets contain useful information (in terms of patterns and specific statistical properties) that can be extracted using appropriate tools. Accordingly, development, enhancement and more importantly the usage of such tools (in this study we are concerned with probabilistic generative models) is an undisputed need which has inspired many investigations.

To further illustrate, consider the example of analyzing medical and healthcare datasets. For such cases, an important task of medical doctors is to infer a possible set of causes from an array of patient symptoms. So far the traditional computer-aided medical diagnostic systems have been used in order to assist the experts in making decisions (e.g., for diagnosing different diseases and treatment suggestions). However, such traditional systems are mostly established on rule-based reasoning and cannot gain knowledge from a set of data (see, e.g., Cai et al., 2019). As a consequence, they lose valuable information which may be vital in medical diagnostic reasoning. In addition, these traditional systems are observed to yield poor results even in performing simple tasks as they can be easily affected by the decision rules that are, in particular, subjectively determined by the experts. Besides, during the last few years, it has been argued that such traditional methods for diagnosis and treatment (which are only

based on a single in-hospital data source) are not capable of addressing all new challenges of our life. For instance, we have observed a global spread of chronic diseases in the past decades which together with the rapidly aging populations led to many new difficulties that traditional systems cannot cope with. Hence, sophisticated automatic systems able to infer hidden information from data have been of considerable interest. Such a request that we discussed here together with many other instances have necessitated new demands for data management and decision-making models which eventually have opened a window for the development of novel machine learning algorithms in a wide range of applications.

Today, with the large amount of datasets at perusal of researchers, many sophisticated methods have been established that can be used for automatic medical reasoning (see, e.g., Gulshan et al., 2016; Lally et al., 2017; Ramaswami, 2015; Finlayson, LePendu, and Shah, 2014 and references therein). Such a huge success is mainly because of the availability of new technologies and resources and of course the researchers who have availed of this window of opportunity. Note that, however, medical data is only one example of complex data that we are concerned with and, in general, the same argument can be made for any specific dataset. Nonetheless, we hereafter exploit medical data analysis as a primary example in order to motivate our upcoming investigations.

In general, an automatic medical diagnostic system can be used either by regular patients, who search around to find out more information about their symptoms and their corresponding possible causes, and/or by experts and doctors, who seek extra assistance for their decisions. The importance of (and also the demand for) such automatic systems become especially evident in a pandemic such as the outbreak of novel coronavirus (SARS-CoV-2). During such time, specialists around the world attempt to diagnose patients contracted with the COVID-19 disease amongst a large group of patients with potential symptoms. In other words, given a set of symptoms such as "fever", "cough", "sneezing", "fatigue" etc, doctors infer causes such as "COVID-19", "common cold" or "flu" and take further actions based on their understandings. This is a formidable, time-consuming and importantly complicated procedure, however.

Amongst machine learning algorithms, probabilistic Latent Variable Models (LVMs), are well established tools that are widely exploited to automate such an inference procedure and further assist both patients and doctors. They are statistical models that can relate a set of random variables known as *latents* (whose realized values are hidden and should be inferred) to a set of observed variables known as *observables* (see, e.g., Skrondal and Rabe-Hesketh, 2007 for more information). These latents (or in another terminology *causes*) could, for instance, denote a set of diseases which we want to infer, and the observables could, for instance, denote a set of observed symptoms. In recent years, LVMs have shown promising results in constructing a causal graph between the latents and observables and have been consequently used to estimate a diagnostic model from a set of patient data (Wang et al., 2014; Rotmensch et al., 2017; Collins and Huynh, 2014; Enøe, Georgiadis, and Johnson, 2000). Once trained, they can assign probabilities to hidden causes (e.g., diseases) given a set of observables (e.g., symptoms).

Current LVMs for medical data analysis, e.g. in the form of noisy-OR networks (Šingliar and Hauskrecht, 2006; Jernite, Halpern, and Sontag, 2013; Xie et al., 2016; Arora et al., 2017; Rotmensch et al., 2017), assume medical symptoms to be encoded by binary observables: A symptom is considered to be present or absent. One example of studies in this direction is the work by Rotmensch et al. (Rotmensch et al., 2017) in which authors have used three different probabilistic LVMs (namely noisy-OR, naive Bayes net and logistic regression) to infer a knowledge graph from a set of electronic health records. Amongst these three models, they have shown that a probabilistic noisy-OR model can outperform the other two and produce state-of-the-art results that are further used for diagnosis. Nevertheless, it can be argued that assuming a binary representation is simplistic. Given a dataset with records of symptoms' measurements, the distribution of symptoms can range from not present, over intermediate,

to severe. In fact, many schemes for medical records establish a more continuous recording of symptoms (Buhl et al., 2020; van Esch et al., 2013; Lehnhardt, 2009). If symptoms are binarized, as assumed by noisy-OR networks or as provided by datasets such as QMR-DT (Shwe et al., 1991), then not all the relevant statistical information can be leveraged. Therefore, new LVMs capable of describing the statistical representations of such complex data and making more accurate predictions for a set of previously unseen data points have been proposed accordingly. For instance, a model based on the Beta distribution can generalize the assumption of binary observables in noisy-OR-based networks which exploit the Bernoulli distribution. Such a generalization can be also beneficial in the aspect that parameters of the variance or of the second moment will be learned alongside the parameters of the mean. This is specially important for precise disease characterization as for any given cause (any disease) the distribution of the symptoms are of interest in different areas of data analysis.

In this study, we seek sophisticated machine learning algorithms that can be used for training different complex datasets such as medical data and images. The medical data will be provided to us in a collaborative work from the Medical Physics group at University of Oldenburg (Buhl et al., 2019; Buhl et al., 2020) and for other datasets (including images, natural image patches and acoustic data), we use benchmarks that are commonly used in similar studies. The aim is to then develop algorithms that can extract rich information and high-level statistical structures from these challenging datasets. We build upon previously established LVMs and present a family of novel probabilistic generative models that can be trained using unlabelled data. The proposed models will be then applied for training the aforementioned datasets in order to recognize patterns associated with the latents (e.g. diseases) and observables (e.g. symptoms). The ultimate goal is, nonetheless, to use this information to automate the inference procedure and further assist the experts. Specifically, for our investigations here, we consider some essential ingredients to build our generative models where each of them will be elaborately discussed in the upcoming chapters. Before that, let us first develop some intuition using the example of diseases and symptoms modelling that mentioned above in order to motivate the necessity of such ingredients.

Consider, for instance, three causes of medical diseases: "COVID-19", "common cold" and "flu"; and further assume four medical symptoms: "fever", "cough", "sneezing" and "fatigue". For the given set of diseases and symptoms, a binary encoding of the diseases/symptoms relations is to assign 0 to the cases where the symptom is not pathological (is normal or healthy) and 1 otherwise. That is the array of inputs embedded to the model consists of 0 and 1 values representing the absence and presence of symptoms, respectively. The canonical choice for modelling such a binary encoding is the Bernoulli distribution (as it is used, for e.g., by the noisy-OR model). An alternative for binary representation is the usage of ordinal data that is to consider a set of finite and ordered categories such as {"normal", "mild", "intermediate" and "severe"}, and then relate each of the symptoms to the diseases using these categories (see, e.g., Vergari et al., 2019; Valera and Ghahramani, 2017). However, one can argue that neither of the two encodings may leverage all statistical information of the data, and thus suggest encoding the symptoms as being continuous from not to barely present (close to zero) over intermediate (around 0.5) to very severe (close to one). This graded modelling is different from conventional binary or ordinal encodings and can be considered as a generalization of both cases. Consequently, the usage of a Beta distribution noise model (instead of, e.g., a Bernoulli distribution) can be suggested as it accepts symptom values in the interval $[0, 1]$. Figure 1.1 then depicts an example of binary versus interval representation of the diseases/symptoms relations which we refer to it as the *disease profile*. Note that, however, the presented disease profile is only an example of how such diseases and symptoms could be related to each other and the sketch may not be accurate (we refer the readers to see, e.g., Jiang et al., 2020; Rothan and Byrareddy, 2020 for more details of the given diseases/symptoms relations). Nevertheless, the disease profile illustrates how a continuous interval encoding, in contrast to a binary

encoding, enables us to capture higher-order statistics of the observables and therefore, obtain more information about the corresponding symptoms.

Similar arguments can be further made for other types of data and encourage development of new models that can leverage various complex probability distributions. For instance, the usage of Poisson distribution has been motivated for photon-limited images, medical imaging and microscopy (Salmon et al., 2014; Giryes and Elad, 2014), and the usage of Gamma distribution for acoustic data (Schauerte and Stiefelhagen, 2013; Bui et al., 2015). Therefore, in this thesis, our goal is to investigate a family of generative models which encompass a variety of probability distributions (including Beta, Gamma, Poisson, Gaussian and many more) for modelling the observables. Importantly, such plausible models should produce interpretable results as it is a vital feature for medical data analysis (and also for other types of datasets). We will later elaborate on this concept when we discuss the differences between LVMs and other alternative approaches.

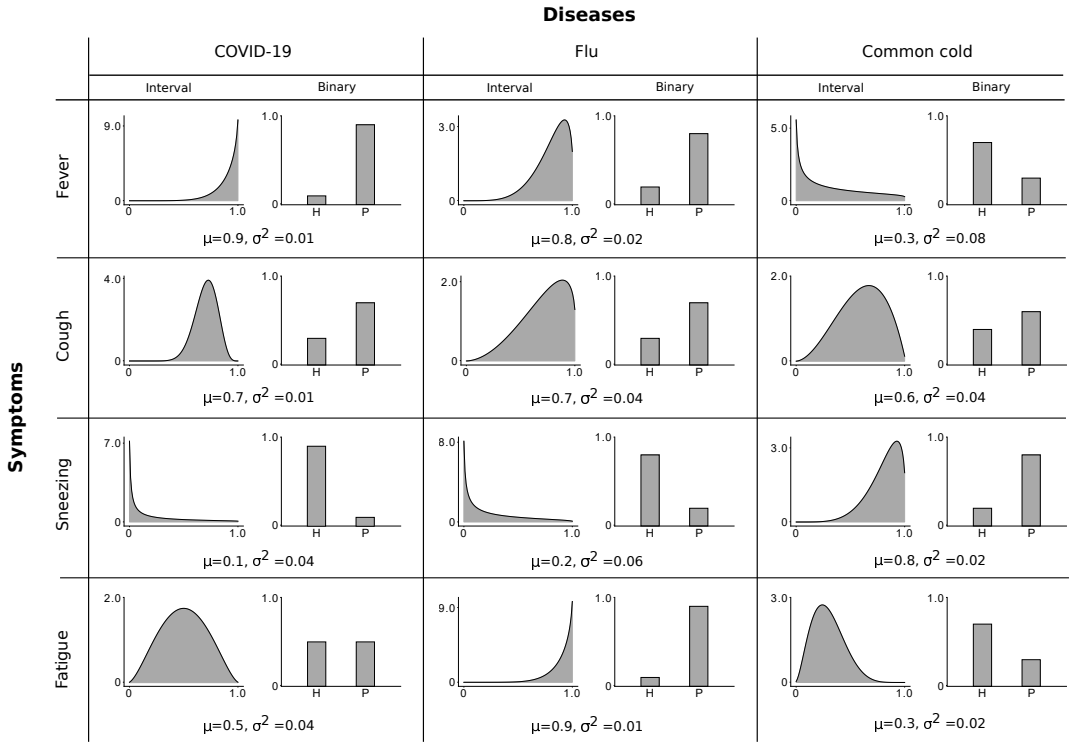


FIGURE 1.1: A disease profile illustrating the differences between a binary and a continuous interval encoding of the observables. Here, the interval $[0, 1]$ is used to model the severity of the symptoms, where 0 is the state of being healthy (normal) and 1 is pathological. Also, 'H' and 'P' in the binary encodings refer to the healthy and pathological cases, respectively. As it can be seen, a continuous interval encoding models higher-order statistics of the data and provides more information compared to a binary encoding. For instance, a continuous encoding considers two different statistics for the symptom "cough" and the two causes of "COVID-19" and "flu" while a binary encoding would assume a similar effect for both causes.

Considering the disease profile presented in Figure 1.1, it can be easily seen that each disease causes different symptoms; e.g., "COVID-19" reliably causes "cough" and a strong "fever". For our generative models, we thus demand a model to relate diseases to the symptoms using a weight matrix M with $D \times H$ entries (corresponding to the number of D symptoms and H diseases; for our running example $D = 4$ and $H = 3$). Subsequently, each element M_{dh} for $d = 1, \ldots, D$ and $h = 1, \ldots, H$ distinguishes the strength connection between

cause (disease) h and observation (symptom) d. For instance, the weight M_{dh} modelling the connection from "COVID-19" to "fever" is strong (a value close to one in this case) while the weight M_{dh} modelling the connection from "COVID-19" to "fatigue" is rather weak (a value close to zero), and so forth. Now, suppose that only the "COVID-19" disease is active, then a feasible model should set the corresponding symptom for "fever" to a value close to one.

Moreover, causes (diseases) have different probabilities to occur which should be modelled by a prior parameter such as $\vec{\pi}$, a vector with H elements. That is, in some regions for instance, contracting to "COVID-19" may be very likely which demands for a high value of prior component π_h for the "COVID-19" while the components for other causes ("common cold" and "flu" in this case) could be low. Importantly, causes (diseases) can co-occur, i.e. a patient can be contracted to different diseases at the same time. Thus, the model should be able to account for the combination of different causes. These types of models are referred to as *multiple-causes* which contrast the so-called mixture models such as Gaussian Mixture Models (GMMs) (see, e.g., Bishop, 2006 for more information). Furthermore, for any symptom, a plausible model for symptom combination should assume the symptom *mean* (i.e. the mean of observables) to be set by the disease with the strongest influence on the symptom. Such a demand is because of the fact that when two diseases (causes) are active at the same time, it is more likely that the one which produces stronger symptoms will be the dominant disease (cause). In other words, a non-linear superposition of the latents and observables is desirable. This is in contrast to a linear superposition which has been frequently used in the literature as a conventional choice for linking the latents to the observables (see, e.g., Olshausen and Field, 1996a).

Now consider another case where, for instance, the strongest disease (like "COVID-19") always causes a strong "fever", i.e. we frequently see a severe fever for almost all the patients. This suggests a mean value close to 1 for the symptom "fever" and a very small value of variance (see the corresponding plot in Figure 1.1). Then even though other diseases may be active, we should model the "fever" with a mean value close to 1 and with a small variance. Subsequently, a disease with an intermediate probability for "fever" (a mean value close to 0.5), and a high variance (like "common cold") should not make the variance for "fever" large if "COVID-19", for e.g., is active. For this specific example, however, most of the previously established LVMs generate the symptoms for "fever" using a global variance σ^2 that results in a mild "fever". Such models will fail to concretely distinguish between different variances corresponding to different causes. This limitation propounds another ingredient of our family of generative models where a global variance encoding of the observables should be replaced by a more generalized approach for encoding the variance parameters. Hence, the desired model should consider (a) a variance component per cause and per symptom, and (b) a non-linear superposition such that the symptom *variance* (i.e. the variance of observables) is also set by the disease with the strongest influence on the symptom.

The discussion above ascertained some of the important properties that an appropriate model should consider, e.g., for medical diagnostic reasoning. In this study, we will investigate each of these properties carefully and seek a family of generative models capable of addressing all issues raised here. However, note that medical and healthcare datasets are only one example of the data that we can consider and also the medical diagnostic reasoning is only one application of such generally applicable generative models. Broadly speaking, the models that we propose here can be applied to a wide range of applications including but not limited to feature extraction, denoising of images and/or acoustic datasets, inpainting, source separation and many more. Regardless of the specific application that we may aim for, the illustrative example above provides the right direction for our research here. Let us now briefly review all these generalizations that we will study in this thesis.

1.1.1 Salient features of the considered generative models

In the following, we summarize the essential properties that we seek here for the investigated generative models:

- **Interpretability** of the obtained results. This is importantly vital for analysis of medical data where the task is to gain an intuition of the causes/symptoms relations.
- **Binary latents** which correspond to assuming a disease (or an object in an image) to be absent or present with a certain probability. The usage of binary latents have been motivated for different datasets and actively researched in previous studies in conjunction with Gaussian or Poisson observables (Haft, Hofman, and Tresp, 2004; Lücke and Sahani, 2008; Henniges et al., 2010; Lücke and Eggert, 2010). We will use Bernoulli distributions here to model binary latents, but we emphasize that all the upcoming theoretical results and findings are valid for other types of binary latents.
- Exploiting **a broad range of noise distributions**. We mentioned the usage of a Beta distribution above, but we will aim at a more general and somehow popular family of distributions here. Namely we investigate the exponential family of distributions.
- **Non-linear superposition** of the latents and observables. We consider a common framework of Maximal Causes Analysis (MCA) models where a maximum function has been used to superimpose the causes (see, e.g., Lücke and Sahani, 2008). Importantly, for our specific purposes here, we investigate a novel generalization of the MCA models where component means alongside component variances are learned.
- **Scalability** of the proposed approaches. With the advancement in information technology, huge amounts of data (mostly unlabelled) are available nowadays. Consequently, the number of causes (diseases) and observations (symptoms) are potentially large which renders exact inference and learning intractable. Hence, the proposed models should be able to consume large-scale datasets. For this purpose, we leverage the theoretical results of, e.g., (Lücke, 2019; Guiraud, Drefs, and Lücke, 2018) to obtain a feasible learning method.

1.2 Literature Review – Latent Variable Models

As mentioned above, LVMs are statistical models which exploit a specific connection between the so-called latents and observables in order to extract hidden information from data. In detail, such widely used models assume a probabilistic graphical model (Koller and Friedman, 2009) that connects a set of latents (parameters that we want to infer as the hidden variables) to observables (parameters that we study as the features or measurements). The considered graph further demonstrates the statistical dependencies between different variables of the model and enables us to successfully encode probability distributions over complex domains. Moreover, the specific connection between latents and observables in LVMs is parameterized using a so-called *link function* defined over the model parameters. Concretely, the assumption of a linear or a non-linear superposition can be embedded to the model through this link function which further sets the foundation for different types of LVMs. In fact, LVMs differ from each other based on their link functions, the general assumptions that they consider regarding the distributions of the variables (either latents or observables) and the specific structure that they assume to connect the variables. All these differences and more importantly the algorithm that each LVM uses for parameter optimization yield a variety of statistical models which can be used for many different tasks. In the previous section, we pointed out one important application of such statistical models for medical data analysis using the

example of COVID-19 (note that we specifically considered the outbreak of COVID-19 as an introductory motivation but will use medical data of hearing impairments (Buhl et al., 2020) for the numerical evaluations in Chapter 5 as such data is directly at our perusal). Nonetheless, LVMs subsume a broad range of statistical models and analysis of medical data is an important but also just one example of their practical applications.

Although LVMs have been first established and developed in the fields of psychology, educational measurement and economics (Izenman, 2008; Bollen, 2002; Farouni, 2017), the current developments have mainly been made by researchers in the fields of machine learning, deep learning, and probabilistic graphical modelling. All these new and ongoing researches have further given the models an emerging context for success. Subsequently, the LVMs, as powerful statistical models, have shown a huge capability in a variety of tasks such as dimensionality reduction, statistical clustering and medical data reasoning. Besides their proficiency in data analysis, LVMs have also shown great potential because of their explanatory and interpretative powers (Farouni, 2017). This is especially efficacious when LVMs are cast in a generative probabilistic framework. The aforementioned feature is very important for our studies here as we seek such an explanatory power for analysis of medical and other complex datasets. Therefore, LVMs are the most appropriate tools (in our opinion) that we can employ for our studies in finding the hidden structures of complex datasets and further capturing the corresponding data generating mechanism. We will review some seminal works that have been done in this direction further below.

Probabilistic Principal Component Analysis (p-PCA) (Tipping and Bishop, 1999; Roweis, 1998), Factor Analysis (FA) (Cattell, 2012) and Sparse Coding (SC) (Olshausen and Field, 1996a) are well-known LVMs which seek to model the observables' distributions. The statistical data models of these approaches consist of latent variables whose weighted linear sum determines the mean of a Gaussian distribution. Specifically, SC is a standard and active field of research with a high relevance for computational neuroscience and tasks such as feature learning, denoising, inpainting, compression and compressive sensing (Donoho, 2006; Baraniuk, 2007). In particular, the approach enables us to learn distinct interpretable and discriminative features from a set of unlabelled data. For many types of data, the generative fields inferred by standard SC models have been interpreted as structural primitives of the corresponding data (Młynarski and McDermott, 2018). Presumably most prominently, the generative fields inferred from whitened image patches have been linked to edges (Olshausen and Field, 1996a). Nevertheless, conventional SCs together with some other well-known LVMs such as PCA and FA have been developed based on two primary assumptions: Gaussian distributed data and linear superposition of the causes.

Not all data is Gaussian distributed, however, and a linear superposition of the latents does not often reflect the true data generating process. For instance, models based on the assumption of Gaussian observation noise have been seen to poorly perform for denoising of photon-limited images (Salmon et al., 2014; Rond, Giryes, and Elad, 2016; Giryes and Elad, 2014). In addition, it has been argued, e.g., for images (Bornschein, Henniges, and Lücke, 2013) or for cochlear representations of sounds (Roweis, 2003; Lücke and Sahani, 2008; Sheikh et al., 2019) that a linear superposition model is difficult to motivate. Consequently non-linear as well as non-Gaussian generalizations have previously been of interest.

Generalizations to non-linear superposition models have, for instance, been investigated in the form of non-linear Independent Component Analysis (ICA) (Hyvärinen and Pajunen, 1999; Hyvärinen, Sasaki, and Turner, 2019; Hyvärinen and Morioka, 2017). We can regard standard ICA as a noiseless limit of standard SC (Dayan and Abbott, 2001). The practical realizations of non-linear ICA make use of a post-linear non-linearity (i.e., a linear superposition followed by a sigmoidal non-linearity). Another relevant study that can be mentioned in this direction is a recent paper (Hyvärinen, Sasaki, and Turner, 2019) where authors introduce a general framework for the non-linear ICA. This new framework then unifies the previously used

models both in theory and in practice (also see Khemakhem et al., 2020; Hyvärinen and Morioka, 2016 for other related studies regarding the new generalizations and applications of the non-linear ICA model). Other approaches, for instance, investigate non-linearities in the form of a maximum in place of the sum (Roweis, 2003; Lücke and Sahani, 2008; Puertas, Bornschein, and Lücke, 2010; Sheikh et al., 2019). The usage of a maximum function is particularly interesting for our study here as it introduces certain occlusion properties to the model which can be beneficial for analysis of medical data (see, e.g., Lücke et al., 2009; Bornschein, Henniges, and Lücke, 2013; Henniges et al., 2014). More importantly, such a maximum function enables us to efficiently derive a general parameter optimization approach for training the proposed generative models (we will further elaborate in Chapter 4).

Likewise, LVMs for non-Gaussian observation noises have been of considerable interest. Related work includes FA with Poisson noise (Zhou et al., 2012), PCA with Poisson noise (Salmon et al., 2014), exponential family PCA (EF-PCA) (Collins, Dasgupta, and Schapire, 2002; Mohamed, Ghahramani, and Heller, 2008), and non-negative matrix factorization, where non-Euclidean distances are frequently used (see, e.g., Hoffman, 2012 and references therein for more information). Furthermore, SC with exponential family distributions has been previously investigated (Lee et al., 2009). Each of these contributions attempt to generalize the assumption of Gaussian distributed data to a broader case of exponential family or to some specific distributions such as Poisson. We will elaborate on these approaches in Chapter 3.

Notably, most approaches considered so far either focused on changing the linear superposition assumption: Like non-linear ICA (Hyvärinen and Pajunen, 1999; Hyvärinen, Sasaki, and Turner, 2019; Hyvärinen and Morioka, 2017) or Maximal Causes Analysis (MCA) (Lücke and Eggert, 2010; Puertas, Bornschein, and Lücke, 2010); or the noise model: Like EF-PCA (Collins, Dasgupta, and Schapire, 2002) or exponential family SC (EF-SC) (Lee et al., 2009). One exception is an early combination of the maximum non-linearity together with a Poisson noise model (Lücke and Sahani, 2008); although further developments for efficient training (Lücke and Eggert, 2010) dropped back to Gaussian noise. Latent variable approaches such as noisy-OR Bayes nets with sparse binary activations (Šingliar and Hauskrecht, 2006; Jernite, Halpern, and Sontag, 2013), Boolean Factor Analysis (BFA) (Frolov, Husek, and Polyakov, 2014) or shallow Sigmoid Belief Networks (SBN) (Saul, Jaakkola, and Jordan, 1996; Gan et al., 2015) could, furthermore, be considered as SC models with Bernoulli noise and a specific non-linear superposition (in the form of noisy-OR non-linearity or in the form of a post-linear superposition for SBNs; also compare approaches such as (van der Linden and Hambleton, 2013)).

Other lines of research, e.g. (Valera and Ghahramani, 2017; Vergari et al., 2019), suggest an automatic procedure using a Bayesian method to estimate the statistical dependencies from a set of heterogeneous data. In particular, the method introduced by Valera and Ghahramani in (Valera and Ghahramani, 2017) can successfully model the true statistical types of data as being real-valued, positive real-valued, interval, categorical or ordinal, and defines a mixture of likelihood functions that factorizes for each of the considered data types. Nevertheless, the approach uses the Markov chain Monte Carlo (MCMC) algorithm for training and therefore, application of the model at large-scales may be computationally challenging. The work by Vergari et al. (Vergari et al., 2019) further generalizes this approach so that the new model can also robustly estimate missing, corruption and anomalies in the data. The new approach, known as Automatic Bayesian Density Analysis (ABDA), then shows a consistent performance for automatic exploratory analysis of complex data (see also Valera et al., 2020 for a related study).

Alternatives to LVMs and Bayes nets are posed by neural network approaches where, for e.g., a variety of deep networks such as Variational Autoencoders (VAEs), Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been used for learning diagnostic models (see, e.g., Rajkomar et al., 2018; Shickel et al., 2017; Miotto et al., 2016;

Lipton et al., 2015; Suresh et al., 2017; Ling et al., 2017 and references therein). Particularly, Deep Neural Networks (DNNs) have shown a strong capability in analyzing medical datasets and have recently gained considerable attention as relatively large datasets, e.g. in the form of electronic health records, are available to researchers. To some extent, they can outperform probabilistic models in terms of both accuracy and run-time as evaluation measures. However, it can be argued that DNNs are not the most appropriate tools for analysis of medical data due to the following reasons: First of all, the results obtained from DNNs are less interpretable as the taken assumptions and also the procedure of producing the results are sometimes unclear (Pearl, 2014; Fei and Li, 2017). This is specifically very important for medical data reasoning. Secondly, DNNs mostly provide the final results, i.e., not the second-most likely solution, nor the statistical properties of the causes/symptoms can be optimally inferred (Shen et al., 2018). Thirdly, they generally require a large amount of (clean) data for training and have seen to poorly perform in the absence of certain important information (see, e.g., Ravuri et al., 2018 for a concrete comparison between different machine learning tools for medical data analysis including DNNs and probabilistic models).

The current study can thus be seen as one of the first approaches that assume a wide range of noise distributions along with a non-linear superposition model. On one hand, we generalize the previous MCA works by incorporating exponential family distributions (together with other generalizations that concerns, e.g., the variance modelling); and on the other hand, we exploit variational Expectation Maximization (EM) that allows for an efficient learning algorithm capable of training the models at large-scales. In addition, note that we will, in a broader sense, use the term LVM for the generative models that we investigate here. But this is only for the sake of simplicity since, in general, such generative models can be specified as *Bayesian networks* (a.k.a. *Bayes nets*), or generalized MCA models or even as non-linear SC models. Throughout the thesis, therefore, we may use any of these terms in order to refer to the proposed models given the specific context of each section.

1.3 The Proposed Approach

In this thesis, we study a family of probabilistic generative models that can generalize previously established LVMs in three different aspects: (a) a maximum non-linear superposition of the latents and observables is considered, (b) a well-behaved distribution of the so-called exponential family distributions including Gaussian, Gamma, Beta, Bernoulli, Exponential, Poisson and many more is considered as the observation noise, and finally (c) a novel combination of observable means alongside observable variances is investigated (for the cases where the considered noise distribution has two parameters to learn). Regarding the latter feature, we present a novel way of modelling the mean and variance parameters using two matrices: One to model the component means and another for component variances. This novel approach enables us to learn a variance component for each cause that can be importantly used for medical data analysis. In addition, besides the aforementioned contributions, the proposed approach here maintains the standard coupling of the latents to the observables in a sense that the non-linear superposition model always sets the *mean* of observables' distribution. That is, unlike previous approaches (such as EF-SC (Lee et al., 2009), EF-PCA (Collins, Dasgupta, and Schapire, 2002) and Bayesian EF-PCA (Mohamed, Ghahramani, and Heller, 2008)), the superposition model does not change with the used observable distribution. Such an important feature together with other generalizations that we consider here instantly result in significant challenges that have to be addressed.

A central challenge that we have to address for such a family of generative models is the derivation of parameter update equations (note that this is in general a formidable task for many of the LVMs with non-Gaussian observables and/or a non-linear superposition).

Here, by replacing the weighted summation over latents with a maximization function, we show that exceptionally general and concise update equations, in the context of Expectation Maximization (EM) algorithm, can be obtained that are directly applicable to any well-behaved member of the exponential family distributions. In this direction, we seek maximum likelihood of the data and exploit the EM algorithm to fit the proposed models to the data. This algorithm, which has been frequently used in the literature, enables us to learn feature representations of the data in an unsupervised manner. This is specifically important as the process of labelling data can be exceedingly time-consuming and, in some cases (e.g. medical data), very expensive. Furthermore, we show that the proposed approach can be scaled using variational EM. Concretely, we use variational EM based on truncated posteriors as variational distributions. Such a variational EM approach has, in our context, the advantage that it merely requires the joint probability, and no further derivations are required (i.e., the variational approach can be used as a *black-box*).

In addition, as mentioned above, the family of non-linear generative models that we study here subsume a variety of distributions as the noise model. This also includes the usage of Beta distribution that generalizes the binary encoding of noisy-OR-like networks to continuous interval encoding. Therefore, we can directly apply our Beta-distributed model for medical data analysis (along with other practical applications that can be considered). We will discuss each of the aforementioned features in the upcoming chapters and present the details of the proposed algorithms. We will further assess the performance of the proposed models using different practical experiments (namely feature extraction, noise type estimation and denoising tasks will be investigated). In short, taking all the aforementioned generalizations into account, we believe that this work can positively affect the current state of LVMs and their applications in achieving a better prediction performance.

1.3.1 The importance of current work

As mentioned above, in this dissertation we aim at developing a family of LVMs to be used for training complex datasets. We now discuss the reasons for and also the importance of the proposed approach and explain why we think that the current study is of interest in the realm of machine learning algorithms. In fact, one could argue against the chosen probabilistic framework and seek other alternatives. Intuitively, we could exploit any other deterministic or generative model which could, in principle, lead to a different learning approach and ultimately different inferences. Currently, a variety of machine learning algorithms have been introduced that range from fully Bayesian approaches such as (Valera and Ghahramani, 2017; Vergari et al., 2019), to deterministic approaches such as (Olshausen and Field, 1996a; Tibshirani, 1996), and Deep Neural Networks (DNNs) such as (Rajkomar et al., 2018; Shen et al., 2018), and to recently established generative models such as Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and Variational Autoencoders (VAEs) (Kingma and Welling, 2013; Rezende, Mohamed, and Wierstra, 2014). Each of these models have been developed and exploited during the last years and different variations/applications of them have been introduced in the community. Generally speaking, these approaches differ from each other in the way they tackle a specific task and the method they use for their optimization procedure. Based on different features of the dataset at hand (number of data points, dimensions, homogeneity or heterogeneity of the data points etc) and also the conditions of the desired task, one can employ each of these machine learning approaches.

Nevertheless, for our purposes here, we persistently argue in favor of the probabilistic LVMs as the usage of aforementioned methods may seem to be infeasible or at least very hard to motivate. In previous sections, we discussed the drawbacks of DNNs and other deterministic approaches, e.g., for the sake of interpretability and also the requirements of training on large amounts of datasets (or for some models, training on clean data or the need of

labels). It should be mentioned that we require our models to be trained on datasets with very few data points (because the number of medical data points at our perusal is limited), which motivates the usage of a more probabilistic approach (the usage of DNNs is consequently very challenging for such amount of data). Bayesian approaches are more flexible in that sense, but, as mentioned before, the scalability of such approaches is often limited. We again emphasize that we demand the application of our models also on images and other challenging datasets.

Amongst the approaches mentioned above, VAEs are of interest for many different tasks and have been actively researched in recent years. These are generative models which can successfully extract hidden information from a set of unlabelled data and describe the true process of data generation. Nonetheless, VAEs have been mainly developed for continuous latent variables which is in contrast to the assumption of binary latents investigated here. Although, generalization of VAEs to discrete latents or to the cases where a specific formulation is applied to feature discrete latents have been recently researched (Rolfe, 2016; Khoshaman and Amin, 2018; Roy et al., 2018; Sadeghi et al., 2019; Vahdat, Andriyash, and Macready, 2019), their optimization methods cannot be easily extended to binary latents. This is due to the fact that *backpropagation* (which is the method for parameter optimization of VAEs along with GANs and other DNNs) cannot be generally applied to binary latents (and in general to discrete latents) (Rolfe, 2016; Bengio, Léonard, and Courville, 2013). Because of this reason, and other obstacles that the assumption of binary latents poses to the model, the usage of VAEs is also not feasible. New studies have been done in order to tackle this issue and present an optimization algorithm for VAEs with binary latents (e.g., Guiraud, Drefs, and Lücke, 2020), but other generalizations that stated before are still essential and seemingly none of the aforementioned approaches can fulfil all of them. For instance, how to generalize the assumption of Gaussian distributed data in VAEs and in other similar networks to the distributions of the exponential family is currently under discussion.

We therefore argue in favor of probabilistic LVMs in the direction of our purposes in this study and further attempt to exploit each of the desired generalizations stated in Section 1.1.1. The summary of the main contributions that we undertake here and also the list of publications that are extracted from this thesis are then presented in the next two sections.

1.3.2 Summary of contributions

- We present a family of novel probabilistic generative models that encompass a large variety of exponential family distributions as the noise model.
- The proposed models exploit a maximum function to link the latents to the observables. The maximum is (a) a relatively canonical choice for a non-linear superposition, (b) suitable for the combination of the causes (e.g., diseases), and (c) will turn out to be particularly convenient in modelling the combination of symptom variances (e.g., for two-parameter distributions).
- Given the novel non-linear combination (using the maximum function), we show that a set of precise and concise update equations will be obtained (in the context of EM algorithm) that share the same functional forms for all well-behaved distributions of the exponential family.
- For any two-parameter distribution of the exponential family, the proposed approach learns two matrices as the parameters of the model: One to model the component means and another for the component variances. Hence, a component variance is learned for each cause that generalizes previous LVMs in this respect. This feature is also valid for any l-parameter distribution of the exponential family such that the number of l matrices will be learned corresponding to each of the model's parameters.

- Considering the Beta distribution, the proposed model generalizes the noisy-OR-based networks. We specifically show the application of such a Beta-distributed LVM in extracting diseases/symptoms relations using Common Audiological Functional Parameters (CAFPAs) (Buhl et al., 2020) which describe the human auditory system. The results will be then used to infer different types of causes for hearing losses or hearing deficits given certain symptoms.

- We employ the Truncated Variational EM (TV-EM) (Lücke, 2019) approach to show large-scale applications of the proposed models. Such variational approximations have been actively researched during the last years to allow large-scale applicability of LVMs, but the approach that we leverage here has the advantage that can be applied as a black-box. That is, importantly, with the change of noise distribution (we use different exponential family distributions), no additional derivations are required.

- We demonstrate the practical applications of the proposed models using different artificial and real datasets. For some experiments, we further compare the performance of our models with other state-of-the-art approaches.

1.3.3 List of publications extracted from this thesis

The content of this thesis has been submitted or published in various publications. The following list describes the representative papers extracted from this study:

- H. Mousavi*, J. Drefs* and J. Lücke.
  A Double Dictionary Approach Learns Component Means and Variances for V1 Encoding.
  The Sixth International Conference on Machine Learning, Optimization, and Data Science (LOD), pages 240–244, 2020.
  (* Mousavi and Drefs share the first authorship for this publication).

- H. Mousavi, J. Drefs, F. Hirschberger and J. Lücke.
  Maximal Causes for Exponential Family Observables.
  Submitted to Machine Learning, (currently under review).

- H. Mousavi, M. Buhl, E. Guiraud, J. Drefs and J. Lücke.
  Inference and Learning in a Latent Variable Model for Beta Distributed Interval Data.
  Entropy 23.5, 552, 2021.

1.4 Organization of the Thesis

- Starting point for our investigations will be the definition of probabilistic LVMs in Chapter 2 where we also describe some well-known models (namely SC, MCA and noisy-OR) that have been frequently used in similar studies.

- In Chapter 3, we will present a family of non-linear generative models and discuss their mathematical details.

- Chapter 4 will be devoted to the EM algorithm and the parameter optimization procedure. More specifically, we will discuss the details of how we train our proposed models given different probability distributions as the observation noise.

- Chapter 5 presents the experimental results where we assess the applicability of our models using both synthetic and real datasets. We will first validate the theoretical

findings presented in Chapter 4 and later point out some practical applications of the proposed models using more realistic datasets.

- In Chapter 6 we conclude the dissertation and discuss the final points and possible directions for the future studies.
- Finally, Appendices A and B present some extra details of our investigations: Appendix A will be devoted to further mathematical details of the proposed generative models and Appendix B will discuss further details of the experimental results.

1.5 Mathematical Notation

Throughout the thesis we will use the following mathematical notations:

- We preserve lower case letters to refer to scalars or vectors and capital letters to refer to matrices. For instance, y denotes a scalar random variable and in the case that the considered random variable is in multi-dimensional space, we distinguish it with an arrow above the letter. On the other hand, W and Σ, for e.g., denote matrices with appropriate dimensions. The elements of a matrix are also denoted, e.g., by W_{dh}.
- $\vec{y}$ denotes an array of observed variables.
- $\vec{s}$ denotes an array of hidden states.
- D denotes the dimension of an observed variable $\vec{y}$. We also use the index d to refer to one element of the observable $\vec{y}$ such as y_d. We also use $\sum_{d=1}^{D}$, $\sum_{d}$, $\sum\limits_{d}$ and $\sum\limits_{d=1}^{D}$ interchangeably to show a summation over different values with subscript d taking values from 1 to D. The same applies for $\prod\limits_{d=1}^{D}$ and $\prod\limits_{d}$ and also for the other subscripts.
- H denotes the dimension of a hidden state $\vec{s}$. We also use the index h to refer to one element of $\vec{s}$ such as s_h. Also, $\sum\limits_{\vec{s}}$ represents the summation over all possible hidden states $\vec{s}$ (the proposed models here use binary latents and thereby $s_h \in \{0,1\}$).
- $Y = \{\vec{y}^{(1)}, \ldots, \vec{y}^{(N)}\}$ denotes number of N samples drawn identically and independently from a probability distribution. We consequently use the superscript (n) to refer to the nth element of the set (i.e. $\vec{y}^{(n)}$).
- Θ represents the parameters of the model.
- $p(\cdot)$ and p are used to represent a probability density function.
- $q^{(n)}(\vec{s})$, for each $n = 1, \ldots, N$, represents a variational distribution defined over the latent parameter space (we use $\mathcal{S}$ to refer to the set of all hidden states).
- $\mathcal{K}^{(n)}$ denotes a set of finite hidden states $\vec{s}$ corresponding to each data point $\vec{y}^{(n)}$. We further use $\mathcal{K}$ to refer to the set of all $\mathcal{K}^{(n)}$, i.e. $\mathcal{K} = (\mathcal{K}^{(1)}, \ldots, \mathcal{K}^{(N)})$.
- $D_{KL}(p||q)$ is the Kullback–Leibler divergence between distributions p and q.
- λ and θ denote values in $\mathbb{R}$ that are used as specific thresholds. In some contexts, λ is also used to denote the parameter of a Poisson or an Exponential distribution.
- The superscript T denotes the transpose of a vector or matrix.

- Both $\frac{\partial}{\partial x}F$ and $\frac{\partial F}{\partial x}$ represent the partial derivative of the function F w.r.t. x.
- $p(a,b)$ represents the joint probability of a and b; and also $p(a|b)$ denotes the conditional probability of a given b.
- $\mathcal{N}(y;\mu,\sigma^2)$ demonstrates the probability density function of a Gaussian (normal) distribution with mean μ and variance σ^2.
- L represents the number of parameters for a probability density function where we use $l = 1,\ldots,L$ to refer to each of these parameters.
- $\vec{\eta} = (\eta_1,\ldots,\eta_L)^T$ denotes the natural parameters of an exponential family distribution.
- $\vec{T}(y) = (T_1(y),\ldots,T_L(y))^T$ is the sufficient statistics of an exponential family distribution.
- $h(y)$ is the base measure of an exponential family distribution.
- $\vec{w} = (w_1,\ldots,w_L)^T$ is used to defined the mean value parameters of an exponential family distribution.
- $\langle y\rangle_{p(y;\vec{\eta})}$ demonstrates the expected value of y w.r.t. the probability density function $p(y;\vec{\eta})$.

Chapter 2

Probabilistic Latent Variable Models

In this chapter, we will discuss the basic definitions of Latent Variable Models (LVMs) and review three well-known models that are closely related to our study here. The details for each of these models are taken from their corresponding papers and we refer the readers to the cited articles in each section for more information regarding these probabilistic models. Moreover, the content of Section 2.4 is taken from (Mousavi et al., 2021) which has been primarily written by Enrico Guiraud and later revised and completed by me.

2.1 Preliminary Definitions

Let us assume that we are given a set of observations where each of them is a D-dimensional vector (we denote an observed variable here by $\vec{y} = (y_1, \ldots, y_D)^T$). The aim is to obtain an intuition of the statistical properties between these different data points (or between the components of each data point), and also the procedure that has led to the generation of these data points (the distribution $p(\vec{y})$). In this direction, probabilistic LVMs are favorable models that have successfully been applied to accomplish the desired tasks.

In particular, LVMs describe the joint distribution of the observed and latent variables. This allows relatively complex distributions to be expressed in terms of more tractable joint distributions over the expanded variable space. Considering $\vec{s}$ to denote a latent variable with H entries and Θ as the parameters of the model, the joint distribution $p(\vec{y}, \vec{s}\,|\Theta)$ can be decomposed into the product of the prior distribution $p(\vec{s}\,|\Theta)$ of the latent variable and the conditional distribution $p(\vec{y}\,|\,\vec{s}, \Theta)$ of the observed variable given the latent $\vec{s}$. In short:

$$p(\vec{y}, \vec{s}\,|\Theta) = p(\vec{y}\,|\,\vec{s}, \Theta)p(\vec{s}\,|\Theta).$$

The structure of such probabilistic models that we study here can be more evident if we consider their graphical representation. In fact, we consider a directed acyclic graph known as a *Bayesian network*[1] (or simply a *Bayes net*) as presented in Figure 2.1 to illustrate the structure of a LVM.

The graphical model depicted in Figure 2.1 illustrates an underlying feature of the LVMs that we assume here: *Conditional independence* (Dawid, 1980). That is, corresponding to each observable $\vec{y}$, we assume a local latent variable $\vec{s}$ which represents the responsible causes for the generation of $\vec{y}$. Then, given the latent variable, we assume the observed variables y_d for $d = 1, \ldots, D$ to be conditionally independent; i.e. we have:

$$p(\vec{y}\,|\,\vec{s}, \Theta) = \prod_{d=1}^{D} p(y_d|\,\vec{s}, \Theta). \tag{2.1}$$

[1] Bayesian networks are statistical models with a probabilistic graphical structure that represents a set of variables and their conditional dependencies using a directed acyclic graph (Bishop, 2006).

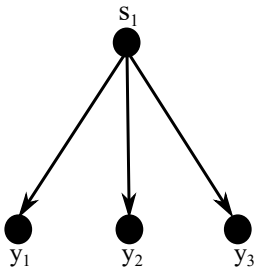


FIGURE 2.1: The nodes represent variables (s_1 is the variable of the latents and $y_1, \ldots, y_3$ are the observables; i.e., $H = 1$ and $D = 3$). Also, the edges represent conditional dependencies. Here, $y_1, \ldots, y_3$ are conditionally independent of each other as no edges connect these nodes to each other.

Furthermore, the corresponding probability of an observed variable, i.e. $p(\vec{y}\,|\Theta)$, can be obtained by marginalization over the latent variables:

$$p(\vec{y}\,|\Theta) = \sum_{\vec{s}} p(\vec{y},\vec{s}\,|\Theta) = \sum_{\vec{s}} p(\vec{y}\,|\,\vec{s},\Theta)p(\vec{s}\,|\Theta) \tag{2.2}$$

where $\sum_{\vec{s}}$ signifies a summation over all possible hidden states $\vec{s}$. It is important to mention that in this study, we are mainly concerned with discrete latents (in fact we consider binary latents and assume a distribution such as Bernoulli for the prior distribution $p(\vec{s}\,|\Theta)$). If we were to apply a LVM with continuous latent variables, we would use an integral instead of a sum in (2.2).

In recent years, LVMs have been actively researched and developed in the field of unsupervised machine learning. This can be seen as a direct consequence of advances in computer powers that enables analyzing big data, and the fact that the labelling procedure of these huge datasets is infeasible. Hence, the demands for unsupervised approaches capable of training such data have increased enormously. In the following, we will discuss three well-known LVMs that have been frequently used in the literature for unsupervised learning tasks. We will namely discuss sparse coding, maximal causes analysis and the noisy-OR models. We will then shortly discuss unsupervised learning as a common feature that all these three LVMs share.

2.2 Sparse Coding

The first model that we discuss here is sparse coding (SC) which, as mentioned before, has been actively researched in different areas of machine learning and computational neuroscience. The goal of the approach is to discover a set of *basis vectors* such as $\vec{M}_1, \ldots, \vec{M}_H \in \mathbb{R}^D$ (containing salient information of the data) such that each observable $\vec{y}$ can be approximately represented as a weighted linear combination of these basis vectors, i.e. $\vec{y} \approx \sum_h \vec{M}_h s_h$, where the latent variable $\vec{s}$ is called the *activation* of the observable $\vec{y}$. In fact, the expectation value of an observable $\vec{y}$ can be computed by $\sum_h \vec{M}_h s_h$ (we will later show how this summation will be used to define a link function connecting the latent variables to the observables). It should be also mentioned that in standard SC models, the activation $\vec{s}$ assumed to be a real number ($\vec{s} \in \mathbb{R}$) which contrasts the binary assumption of our investigations.

Importantly, the approach encourages each activation $\vec{s}$ to be sparse, i.e., enforcing many of the elements s_h to be equal to zero. Therefore, SCs can reliably learn a compact and succinct representation of the data. In another terminology, each basis vector $\vec{M}_h = (M_{1h}, \ldots, M_{Dh})^{\mathrm{T}}$ is also referred to as the *generative field* of unit h; and the $D \times H$ matrix containing all generative fields, $M = (\vec{M}_1, \ldots, \vec{M}_H)$, is commonly referred to as the model's *dictionary*.

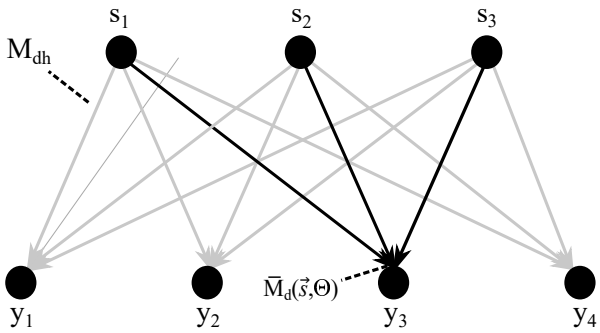


FIGURE 2.2: Graphical structure of a probabilistic SC model with $H = 3$ hidden variables and $D = 4$ observed variables. Given a vector $\vec{s}$, the value y_d of the observables is conditionally independent and drawn from a Gaussian distribution with mean parameter $\bar{M}_d(\vec{s},\Theta)$ determined by weights $M_{d1},\ldots,M_{d3}$ and a global variance σ^2.

We here use both terms 'basis function' and 'generative field' interchangeably. The same argument applies for the terms 'observable' and 'input' (referring to $\vec{y}$) that will be used interchangeably and also for the terms 'latent variable' and 'activation' (referring to $\vec{s}$). Moreover, note that the dictionary M, which is also known as the *weight matrix*, contains all the weight elements M_{dh} that describes the relation between cause h and observable d. Figure 2.2 then illustrates the structure of such a generative model.

In general, a standard generative model for SC assumes that the observables follow a Gaussian distribution (are Gaussian distributed) and the latents another distribution such as Laplace which enforces a sparse representation of the variables. In probabilistic SC, for instance, the set of D observed variables y_d can be modelled by a set of H latent variables s_h according to the following generative model:

$$p(\vec{s}\,|\Theta) = \prod_{h=1}^{H} p_{\text{sparse}}(s_h;\Lambda) \tag{2.3}$$

$$p(\vec{y}\,|\,\vec{s},\Theta) = \prod_{d=1}^{D} \mathcal{N}(y_d;\bar{M}_d(\vec{s},\Theta),\sigma^2), \quad y_d \in \mathbb{R} \tag{2.4}$$

$$\text{where} \quad \bar{M}_d(\vec{s},\Theta) = \sum_{h=1}^{H} M_{dh}s_h, \quad d = 1,\ldots,D \tag{2.5}$$

where $\Theta = (\Lambda, M, \sigma^2)$ denotes all parameters of the model (Λ denotes the parameters of the prior distribution and M and σ^2 are the parameters of the noise distribution). Furthermore, the term 'p_{sparse}' refers to a sparse distribution that is used as prior distribution $p(\vec{s}\,|\Theta)$. The canonical choice for such a distribution is the Laplace distribution (see, e.g., Olshausen and Field, 1996a; Tibshirani, 1996). Other choices include the Cauchy distribution (which is used by Olshausen and Field, 1996a, alongside Laplace), Student's t-distribution (Berkes, White, and Fiser, 2009), Bernoulli (Haft, Hofman, and Tresp, 2004; Henniges et al., 2010), Categorical (Exarchakis and Lücke, 2017), or spike-and-slab (Titsias and Lázaro-Gredilla, 2011; Goodfellow, Courville, and Bengio, 2012a; Sheikh, Shelton, and Lücke, 2014). In addition, note that if a Gaussian instead of a sparse distribution is used as prior, the data model is the one of probabilistic Principal Component Analysis (p-PCA) (Roweis, 1998; Tipping and Bishop, 1999).

Although all the aforementioned prior distributions enforce sparse activations, their corresponding SC models may differ substantially from each other. For instance, using the

Bernoulli distribution suggests a transition from continuous latents as it is used by conventional SCs to binary latents. Importantly, such a transition yields an elementary and yet very efficient generative model, known as Binary Sparse Coding (BSC), which models the absence or presence of a certain cause (activation) with a probability of $\pi \in [0,1]$. In the case of medical data, this indicates the existence of diseases with a probability of π. In particular, BSC replaces Equation (2.3) by:

$$p(\vec{s}\,|\Theta) = \prod_{h=1}^{H} \pi^{s_h}(1-\pi)^{1-s_h}, \quad s_h \in \{0,1\} \text{ and } \pi \in [0,1] \tag{2.6}$$

where the same activation probability π is considered for each hidden unit h. Consequently, the parameters of the model are $\Theta = (\pi, M, \sigma^2)$. The usage of binary latents has been motivated because of the inherent properties of some datasets. For instance, binary latent variables may better describe the state of being healthy or diseased in medical datasets and/or the existence or absence of an object in an image. As a consequence, the model has been repeatedly used to learn the hidden structures of many different datasets (such as images and natural image patches) and shown promising results (see, e.g., Henniges et al., 2010; Drefs, Guiraud, and Lücke, 2020 for different applications of the BSC model). Furthermore, binary latents have a useful property which allows us to obtain a general parameter optimization approach here. We will discuss the details of this property and specifically, the way we leverage such a feature to derive our parameter update equations in Chapter 4.

The predominant approaches to infer the generative fields M are deterministic algorithms which exploit a specific form of (2.3)-(2.5). The Lasso approach (Tibshirani, 1996), for instance, is based on a maximum-a-posteriori (MAP) estimate of the latent vector $\vec{s}$ (whose elements are taken to be Laplace distributed). Given a data point $\vec{y}$, the corresponding MAP estimate $\vec{s}$ can then be computed (approximately) by solving a convex optimization problem. Moreover, given the MAP estimates, the matrix M is updated using standard closed-form updates for M (see, e.g., Olshausen and Field, 1996a; Tibshirani, 1996). Nonetheless, MAP approaches are less suitable for SC models that use priors with richer structures (e.g., Berkes, White, and Fiser, 2009; Titsias and Lázaro-Gredilla, 2011) where approximate inference approaches such as sampling or variational optimizations are applied instead. Other studies follow a more probabilistic algorithm to estimate the parameters of the model. Namely, the Expectation Maximization (EM) algorithm (Dempster, Laird, and Rubin, 1977; Neal and Hinton, 1998) has been actively researched as a concrete method for the maximum likelihood estimation. The details of this algorithm will be further discussed in Chapter 4.

In addition, for a SC model, one can assume the basis set to be overcomplete (i.e., $H > D$). This important feature enables the model to capture a large variety of generative fields that are assumed to represent the structural primitives of the input data. One seminal study in this direction is the work by Olshausen and Field (Olshausen and Field, 1996a) where an overcomplete SC model is applied to natural image patches and the resultant generative fields are linked to the receptive fields of simple cells. In fact, it has been shown that the spatial receptive fields of simple cells contain higher-order statistics such as being localized, oriented, and bandpass (Olshausen and Field, 1996b; Hubel and Wiesel, 1968; Jones and Palmer, 1987). Such structures can also be found in natural images which cannot be characterized in terms of linear, pairwise correlations and therefore some preliminary models based on PCA are not able to efficiently describe them. Subsequently, Olshausen and Field have proposed the usage of an overcomplete SC model such that by maximizing the sparseness of the representation, an optimal sparse coding of the natural images succeeds in producing receptive fields (also see Olshausen and Field, 1997; Olshausen, Cadieu, and Warland, 2009; Olshausen and Field, 2004). The authors argue that an overcomplete model can learn a large number of basis functions containing interesting interactions which resemble receptive fields of neurons. We

will later discuss the details of such SC models in Sections 3.3 and 5.2.2. Besides, the theory of SC has close links to deep learning approaches (see, e.g., Patel, Nguyen, and Baraniuk, 2016 for a relatively recent discussion).

In general, however, the two assumptions that the generative model (2.3)-(2.5) undertakes (Gaussian distributed data and linear superposition) limit the application of the model to certain datasets. For instance, given the example of causes/symptoms in Chapter 1, one can easily infer that the model is less suitable for analysis of medical data as a linear superimposition of the active causes is considered. Thereby, the two assumptions should be modified in the direction of our purposes here. The next LVM that we discuss here replaces the linear superposition assumption of the BSC model with a non-linear superposition using a maximum function.

2.3 Maximal Causes Analysis

Another LVM which we will discuss here is known under the name of Maximal Causes Analysis (MCA) and has been investigated, e.g., by (Lücke and Sahani, 2008; Lücke and Eggert, 2010; Puertas, Bornschein, and Lücke, 2010; Bornschein, Henniges, and Lücke, 2013; Shelton et al., 2017). MCA models introduced in these contributions assume binary latent variables and let the cause with the largest weight determine the value of an observable $\vec{y}$; i.e., a non-linear superposition model is considered in contrast to the linear superposition of the BSC model. This has been incorporated to the model using a point-wise maximum function (we demand an observable $\vec{y}$ to be approximated by $\vec{y} \approx \max_h\{s_h\vec{M}_h\}$ for an activation $\vec{s}$ corresponding to the observable $\vec{y}$). The maximum superposition has been suggested, e.g., for acoustic data (Roweis, 2003; Lücke and Sahani, 2008; Sheikh et al., 2019) and for visual data (Bornschein, Henniges, and Lücke, 2013) where it aligns more closely with the actual data generating process. In the case of images, for instance, the maximum function enables the model to interpret a certain occlusion in the data which results in a more accurate encoding of the generative fields and ultimately an improved inference (see, e.g., Bornschein, Henniges, and Lücke, 2013). Such an occlusion can be seen in many different areas (specifically in natural images) and thus the application of the MCA models has been actively researched in many contributions (e.g., Bornschein, Henniges, and Lücke, 2013; Lücke et al., 2009; Henniges et al., 2014; Dai and Lücke, 2012).

For analysis of medical data, the considered non-linear superposition in the form of the maximum function seems to be an appropriate choice as it can better describe the combination of different diseases (causes) with different degrees of severity (we refer the readers to our discussion in Chapter 1). Given the maximum function, the dominant disease (cause) would set the value of each symptom. In other words, it can be stated that the dominant cause *occludes* the other causes and its connection value (the weight M_{dh}) would be the only value affecting the observable d. Figure 2.3 then depicts the differences between a maximum and a linear superposition of the latents using artificial bars.

The generative model of these MCA networks with Gaussian observation noise (it should be mentioned that the generative model described in (Lücke and Sahani, 2008) assumes a Poisson noise model) can be defined as follows:

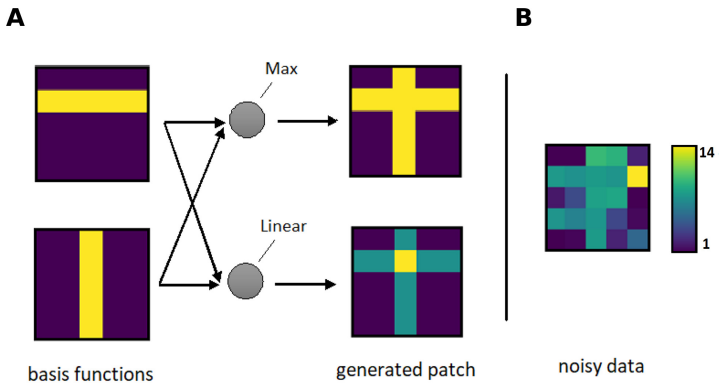


FIGURE 2.3: Illustration of non-linear versus linear combination of generative fields. **A** Two arbitrary basis functions that combined through a maximum or a linear combination to generate data. **B** A noisy data point obtained from the maximum superposition of the two basis functions plus noise.

$$p(\vec{s}\,|\Theta) = \prod_{h=1}^{H} \text{Bernoulli}(s_h; \pi), \quad s_h \in \{0,1\} \tag{2.7}$$

$$p(\vec{y}\,|\,\vec{s},\Theta) = \prod_{d=1}^{D} \mathcal{N}(y_d; \bar{M}_d(\vec{s},\Theta), \sigma^2), \quad y_d \in \mathbb{R} \tag{2.8}$$

$$\text{where} \quad \bar{M}_d(\vec{s},\Theta) = \max_h \{s_h M_{dh}\}, \quad h = 1,\ldots,H \text{ and } d = 1,\ldots,D. \tag{2.9}$$

For the given H hidden variables, similar to the BSC model, the Bernoulli distribution assumes a cause to be present or not ($s_h \in \{0,1\}$) and assigns a prior probability $\pi \in [0,1]$ to each cause h. Likewise, the set of all model parameters is $\Theta = (\pi, M, \sigma^2)$. In contrast to the linear summation in (2.5), here, the hidden variables (causes) determine the *mean* of an observable d through a maximum function: For a given observable d the active cause with the highest value of M_{dh} determines the mean. The data model (2.7)-(2.9) therefore amends the BSC model in the aspect that the linear superposition is replaced by a non-linear superposition. However, the model is still specific to the Gaussian noise which restricts the generality of the model. That is, the model may fail to appropriately encode other types of data points such as counts or interval data. This is specifically important as we wish to, for e.g., train medical datasets with values in the interval $[0,1]$ (as described in Chapter 1). Therefore, even though the maximum function replaces the linear summation, other generalizations are still required in the direction of our desires stated in Chapter 1.

2.4 The Noisy-OR Model

Finally, we discuss the probabilistic noisy-OR model as another LVM related to our study here. The model is a non-linear bipartite Bayesian network with all-to-all connectivity among hidden and observed variables. All variables take binary values. The model assumes a Bernoulli prior

for the latents, and active latents are then combined via the noisy-OR rule:

$$p(\vec{s}\,|\Theta) = \prod_{h=1}^{H} \pi_h^{s_h}(1-\pi_h)^{1-s_h}, \quad s_h \in \{0,1\} \tag{2.10}$$

$$p(\vec{y}\,|\,\vec{s},\Theta) = \prod_{d=1}^{D} \mathrm{NOR}_d(\vec{s})^{y_d}(1-\mathrm{NOR}_d(\vec{s}))^{1-y_d}, \quad y_d \in \{0,1\} \tag{2.11}$$

$$\text{where} \quad \mathrm{NOR}_d(\vec{s}) = 1 - \prod_{h=1}^{H}(1-M_{dh}s_h),\ d = 1,\ldots,D \text{ and } M_{dh} \in [0,1] \tag{2.12}$$

where $\Theta = (\vec{\pi}, M)$ is the set of model parameters. Moreover, parameter $\vec{\pi} = (\pi_1,\ldots,\pi_H)^T$ is the set of values $\pi_h \in [0,1]$ representing the prior activation probabilities for the hidden variables s_h, and M is a $D\times H$ matrix of values $M_{dh} \in [0,1]$ representing the probability that an active latent variable s_h activates the observable y_d.

Observe that the data model (2.10)-(2.12), in comparison to the previous SC models (BSC and MCA), assumes individual prior parameters (i.e. a single probability of π_h is considered corresponding to each cause h). In addition, the model assumes the observables to be distributed according to the Bernoulli distributions which compares with the Gaussian assumption of BSC and MCA. Such a binary-binary model has been successfully applied to automatically derive a causal graph, which relates diseases to the symptoms, from a set of Electronic Medical Record (EMR) data (Rotmensch et al., 2017). Analyzing the medical records of over 270,000 patients, the authors have further shown that a probabilistic noisy-OR model produces a high-quality knowledge graph and state-of-the-art results in terms of the obtained precision in the clinical evaluation (see Rotmensch et al., 2017 for detailed information). Furthermore, they have validated their method against Google's manually-constructed knowledge graph (Ramaswami, 2015) and against the expert physicians' opinions. Also they have compared the performance of noisy-OR with a naive Bayes net and with a logistic regression and showed that the probabilistic noisy-OR model can outperform the other two models.

The aforementioned contribution together with a large number of other studies (see, e.g., Šingliar and Hauskrecht, 2006; Jernite, Halpern, and Sontag, 2013; Xie et al., 2016; Arora et al., 2017; Rotmensch et al., 2017 and references therein for a variety of similar works) have demonstrated the effectiveness of the noisy-OR model (specifically) for medical data analysis. However, regardless of the high-quality results obtained by the noisy-OR model, we still argue that a binary encoding of the data misses essential features that are vital for the inference. Our running example of causes/symptoms relations have made this point clear that a continuous interval encoding of the data is more desirable. The point that we will attempt to address in the next chapter.

2.5 Unsupervised Learning

One important feature that all the three models mentioned above share is the fact that they can be trained unsupervised. In other words, no labelled data (or even non-noisy data) is required for the training of the models. This is in contrast to many of the machine learning algorithms, specially deep learning approaches, that should be trained either supervised or with a large number of clean (non-noisy) data points (this is an important property that some of the neural networks share; see, e.g., Remez et al., 2017; Kumwilaisak et al., 2020; DeGuchy et al., 2019). The importance of this feature will be noted when we realize how expensive the labelling procedure is. Importantly for medical data, this requires a huge amount of human work that should be done by the experts and specialists of the field (a machine learner cannot do this job

as it involves specific medical knowledge). Such difficulties have been further highlighted as labelling big data, which are nowadays at our perusal, is almost impossible. Self-labelling approaches may be further used (see, e.g., Lee, 2013; Triguero, García, and Herrera, 2015) but this is also an intricate procedure specifically for medical data.

Besides, unlabelled data are easily available and can be obtained without much effort. Examples are images, speeches, videos and texts which are available to anyone and, for e.g., can be obtained easily from the web. In fact, even though we may not have labels for a set of data points, there often exist rich structures in the data that can be learned by the algorithm. For example, given natural images, we may be able to discover a variety of different structures including both low-level structures (such as edges) and high-level structures (such as corners, local curvatures, and shapes). A desirable model would then be able to extract such information without any supervision. The same applies for many of the other unlabelled datasets.

We are subsequently interested in using unlabelled data and investigating the problem of learning feature representations in an unsupervised way (similar to the three LVMs described above). In practice, however, one can modify the proposed generative models here such that they leverage extra information from the labels. That is, for instance, one can exercise a semi-supervised (or even a supervised) approach (see, e.g., Forster, Sheikh, and Lücke, 2018) and adjust the learning procedure according to the desired task. Especially, in the absence of enough data points for training, such alternatives would be beneficial. This, however, is far from our goal in this study and we mainly restrict our investigations to unsupervised learning and unlabelled datasets.

Chapter 3

A Family of Non-Linear Latent Variable Models

In the previous chapter, we discussed probabilistic LVMs and briefly overviewed three important models that have repeatedly been used by experts to extract the latent structures of different datasets. Amongst the three models, MCA approaches are specifically substantial for our study here as they replace the linear superposition of the SC models by a maximum function. This has been shown to favor many datasets as it introduces a certain non-linearity to the model. Building upon these approaches, we here question how such non-linear LVMs can be generalized in order to exploit other observation noises. To our best knowledge, Gaussian and Poisson are the only two distributions that have successfully been applied in the framework of MCA models. In this study, however, we introduce a family of non-linear LVMs that encompasses a large variety of distributions as the noise model. More specifically, we generalize the previous MCA approaches and present a family of novel generative models in which the observation noise is a member of the exponential family distributions. This family includes many of the frequently used distributions such as Gaussian, Gamma, Bernoulli and Poisson as well as other complicated and less encountered distributions such as Beta, Multinomial, Categorical and so forth. The presented models will be therefore suitable for a variety of data types including binary, counts and intervals.

For this purpose, we will first define the exponential family of distributions and discuss some useful properties of them. Next, we will present the family of non-linear generative models which will be later used to learn the latent structures of different data types. The content of this chapter is mainly based on (Mousavi et al., 2020) (currently under review). The original idea of exploiting exponential family distributions was suggested by Jörg Lücke. Later, we both established the mathematical notations used here and also developed the general framework of the proposed LVMs. He was also involved in the writing of details presented in Sections 3.2 and 3.5 which have been primarily reported as parts of (Mousavi et al., 2020). Moreover, a literature review on previous LVMs (appeared in Section 3.2.2) carried out by Jörg Lücke which was later revised and completed by me. Nevertheless, as the first author of (Mousavi et al., 2020), I was responsible and also involved in most of the preparations and final revisions of the materials presented in the paper(s) and further adapted the texts and the details in order to fully describe them in this thesis. Also the idea of using a background model stated in Section 3.4.1 was jointly established and written together with Jörg Lücke. In addition, the content of Section 3.3 is based on (Mousavi, Drefs, and Lücke, 2020) which has been investigated, developed and written together with Jakob Drefs and Jörg Lücke.

3.1 Exponential Family Distributions

Exponential family of probability distributions subsumes a group of distributions $p(y;\vec{\eta})$ that share the following general form of density function:

$$p(y;\vec{\eta}) = h(y)\exp\big(\vec{\eta}^T\vec{T}(y) - A(\vec{\eta})\big), \quad y \in \mathcal{Y} \tag{3.1}$$

where $\mathcal{Y}$ is the domain of the observables which may denote binary $\{0,1\}$, integer $\mathbb{N}$, real $\mathbb{R}$ or any subset of real numbers. Moreover, $h(y)$ is the *base measure*, $\vec{T}(y)$ represents the *sufficient statistics* of the data, and $\vec{\eta}$ and $A(\vec{\eta})$ are *natural parameters* and *log-partition*, respectively. Each of the vectors $\vec{T}(y) = (T_1(y),\ldots,T_L(y))^T$ and $\vec{\eta} = (\eta_1,\ldots,\eta_L)^T$ has L elements when distribution $p(y;\vec{\eta})$ is an L-parameter distribution. Moreover, from (3.1) we can obtain:

$$\exp\big(A(\vec{\eta})\big)p(y;\vec{\eta}) = h(y)\exp\big(\vec{\eta}^T\vec{T}(y)\big) \tag{3.2}$$

which integrating over the observation space results in:

$$\exp\big(A(\vec{\eta})\big)\underbrace{\int p(y;\vec{\eta})dy}_{=1} = \int h(y)\exp\big(\vec{\eta}^T\vec{T}(y)\big)dy \tag{3.3}$$

and thus:

$$A(\vec{\eta}) = \log\Big(\int h(y)\exp\big(\vec{\eta}^T\vec{T}(y)\big)dy\Big). \tag{3.4}$$

The log-partition $A(\vec{\eta})$ has an important role in the exponential family as it carries salient features such as being convex with respect to the natural parameters. Importantly, one special case of the exponential families is the case when the log-partition is finite, i.e. $A(\vec{\eta}) < \infty$. Such distributions are then said to be *regular*. In this study, we will benefit from these properties and further restrict ourselves to the regular distributions of the family. The following theorem then presents another property of the log-partition function which will be later used to obtain the parameter updates of our proposed generative models:

Theorem 1 (Wainwright and Jordan, 2008)**.** *For any regular distribution of the exponential family, $A(\vec{\eta})$ satisfies:*

$$\frac{\partial A(\vec{\eta})}{\partial \eta_l} = \big\langle T_l(y)\big\rangle_{p(y;\vec{\eta})}, \quad l = 1,\ldots,L \tag{3.5}$$

where $\langle T_l(y)\rangle_{p(y;\vec{\eta})}$ denotes the expectation value of $T_l(y)$ w.r.t. the distribution $p(y;\vec{\eta})$ (evaluated per entry if y is a vector).

Proof. Let $A(\vec{\eta})$ to be given by (3.4) and further assume $A(\vec{\eta}) < \infty$, then we can obtain:

$$\begin{aligned}
\frac{\partial A(\vec{\eta})}{\partial \eta_l} &= \frac{\partial}{\partial \eta_l}\log\Big(\int h(y)\exp\big(\vec{\eta}^T\vec{T}(y)\big)dy\Big) \\
&= \frac{\int T_l(y)h(y)\exp\big(\vec{\eta}^T\vec{T}(y)\big)dy}{\int h(y)\exp\big(\vec{\eta}^T\vec{T}(y)\big)dy} \\
&= \frac{\int T_l(y)h(y)\exp\big(\vec{\eta}^T\vec{T}(y)\big)dy}{\exp\big(A(\vec{\eta})\big)} \\
&= \int T_l(y)h(y)\exp\big(\vec{\eta}^T\vec{T}(y) - A(\vec{\eta})\big)dy \\
&= \big\langle T_l(y)\big\rangle_{p(y;\vec{\eta})}
\end{aligned} \tag{3.6}$$

which completes the proof. □

Note that an alternative notation for $\langle T_l(y)\rangle_{p(y;\vec{\eta})}$ is $\mathbb{E}_p[T_l(y)]$ that we could use here. However, we will use the former notation in this study for the sake of consistency with previous works (Lücke and Sahani, 2008; Sheikh, Shelton, and Lücke, 2014). Furthermore, the notation which we used at (3.1) to describe the exponential family distributions is the form that has been introduced and used in many studies. This specific formulation is known as the *canonical* parametrization of the exponential family. Other forms of exponential families have also been used in the literature such as the *standard* form (see, e.g., Rockafellar and Wets, 2009). In the following, we will discuss one specific parametrization of the exponential family, the *mean value parameterization*, which sets the foundation for the investigated family of generative models.

3.1.1 The mean value parametrization

As an important tool to later derive parameter update equations, we use the mean value parametrization (a.k.a. the *mean parametrization*) of the exponential family. To this, we consider parameters $\vec{w} = (w_1, \ldots, w_L)^T$ defined by:

$$\vec{w} := \langle \vec{T}(y)\rangle_{p(y;\vec{\eta})} \tag{3.7}$$

and assume distributions (3.1) to be parameterized w.r.t. the mean (value) parameters $\vec{w}$. According to Theorem 1, the above definition defines a mapping from the natural parameter space to the mean value parameter space since the terms $\langle \vec{T}(y)\rangle_{p(y;\vec{\eta})}$ are well defined given the natural parameters; that is:

$$w_l = \frac{\partial A(\vec{\eta})}{\partial \eta_l}, \quad l = 1, \ldots, L. \tag{3.8}$$

The mapping (3.8) is bijective and its inverse is well defined if and only if the distribution has a *minimal representation* property (see Wainwright and Jordan, 2008 for a concrete proof; also see Cox, 2006). In particular, a distribution of the exponential family is said to be minimal if there are no linear constraints among components of the sufficient statistics, i.e., there is no coefficient $\vec{a} \in \mathbb{R}^L$, such that $\vec{a} \neq 0$ and

$$\sum_{l=1}^{L} a_l T_l(y) = constant, \quad \text{for any } y \in \mathcal{Y}.$$

In fact, for the case of minimality, the log-partition function $A(\vec{\eta})$ is strictly convex and therefore there is a one-to-one relation between natural parameters of the distributions and the defined mean value parameters. In this study and in order to define our generative models, we hereafter consider distributions of the exponential family with the minimal representation property that consequently results in the following: There exists a function $\vec{\Phi}$ which maps the mean value parameters to the natural parameters, i.e.

$$\vec{\eta} = \vec{\Phi}(\vec{w}), \quad \text{such that} \quad \vec{w} = \langle \vec{T}(y)\rangle_{p(y;\vec{\Phi}(\vec{w}))}. \tag{3.9}$$

It should be mentioned that by substituting the above equation into the exponential families (3.1), the distributions will be reformulated in terms of mean value parameters. We will later show how such a function will be used for the definition of our non-linear LVMs. Before that, however, let us now examine the existence of function $\vec{\Phi}$ for some example distributions of the exponential family.

3.1.2 A few examples of the exponential family

To give you an intuition of the exponential families and the relations between the corresponding natural and mean value parameters, we here present the details for six frequently used probability distributions of this family. We specifically chose the ones that will be later used for our experiments. Other distributions of the family can also be treated in a similar way.

The Bernoulli distribution

As the first example here, we consider the Bernoulli distribution. The corresponding probability mass function (PMF) of the Bernoulli distribution is given by (2.6) for each $\pi \in [0,1]$. Moreover, using the following equations, one can easily validate the fact that Bernoulli is a member of the exponential family distributions:

$$h(y) = 1, \quad A(\eta) = \log(1+\exp(\eta)), \quad \eta = \log\big(\frac{\pi}{1-\pi}\big), \quad T(y) = y.$$

Note that the trick to obtain above formulas is to compute $\exp(\log(p))$ for each of the considered probability mass (or density in the case of continuous random variables) functions and then simplify the resultant equations. Furthermore, according to Theorem 1, we have:

$$\langle y \rangle_{p(y;\vec{\Phi}(\vec{w}))} = \frac{d}{d\eta} A(\eta) = \frac{\exp(\eta)}{1+\exp(\eta)}.$$

Then, in order to compute the inverse mapping Φ, we write:

$$w = \langle y \rangle_{p(y;\vec{\Phi}(\vec{w}))} = \frac{\exp(\eta)}{1+\exp(\eta)} \quad \Longrightarrow \quad \Phi(w) = \log\big(\frac{w}{1-w}\big).$$

As it can be seen, the function Φ can be computed in closed-form (in this case) which represents the *logistic function*. Also, note that $\langle y \rangle_{p(y;\vec{\Phi}(\vec{w}))}$ denotes the mean of Bernoulli distribution which is equal to π.

The Poisson distribution

Next, we consider the Poisson distribution which is likewise a one-parameter distribution of the family. For $\lambda > 0$, the PMF of the Poisson is given by:

$$p(y;\lambda) = \frac{\lambda^y \exp(-\lambda)}{y!}, \quad y \in \mathbb{N}$$

that can be reparameterized in the form of exponential families, by setting:

$$h(y) = \frac{1}{y!}, \quad A(\eta) = \exp(\eta), \quad \eta = \log(\lambda), \quad T(y) = y.$$

Given the equations above, the inverse mapping Φ can be simply computed as follows:

$$w = \langle y \rangle_{p(y;\vec{\Phi}(\vec{w}))} = \exp(\eta) \quad \Longrightarrow \quad \Phi(w) = \log(w)$$

where we further used the results of Theorem 1.

The Exponential distribution

Similarly, the Exponential distribution is a one-parameter distribution of the family where, given a parameter $\lambda > 0$, its probability density function (PDF) can be written as:

$$p(y;\lambda) = \lambda \exp(-\lambda y), \quad y \in [0,\infty).$$

Further, to obtain the exponential family parametrization, we assume:

$$h(y) = 1, \quad A(\eta) = -\log(-\eta), \quad \eta = -\lambda, \quad T(y) = y.$$

Now, following the method used in previous cases, we write:

$$w = \langle y \rangle_{p(y;\vec{\Phi}(\vec{w}))} = -\frac{1}{\eta} \quad \Longrightarrow \quad \Phi(w) = -\frac{1}{w}$$

which likewise represents a closed-form solution for the inverse mapping Φ.

The Gaussian distribution

Next, as a two-parameter distribution, we consider the Gaussian which has been frequently used in different statistical models. The PDF of this distribution is defined as follows:

$$p(y;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y-\mu)^2}{2\sigma^2}\right), \quad y \in \mathbb{R}$$

where μ and σ^2 are the mean and variance parameters, respectively. In addition, the above equation can be reformulated as in (3.1) by considering the following relations:

$$h(y) = \frac{1}{\sqrt{2\pi}}, \qquad A(\vec{\eta}) = -\frac{\eta_1^2}{4\eta_2} - \frac{1}{2}\log(-2\eta_2)$$
$$\vec{\eta} = \begin{pmatrix} \mu/\sigma^2 \\ -1/(2\sigma^2) \end{pmatrix}, \qquad \vec{T}(y) = \begin{pmatrix} y \\ y^2 \end{pmatrix}.$$

By expressing the mean value parameters $\vec{w}$ in terms of the natural parameters $\vec{\eta}$, we can easily obtain (by using Theorem 1):

$$\vec{w} = \begin{pmatrix} \langle y \rangle_{p(y;\vec{\Phi}(\vec{w}))} \\ \langle y^2 \rangle_{p(y;\vec{\Phi}(\vec{w}))} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial \eta_1} A(\vec{\eta}) \\ \frac{\partial}{\partial \eta_2} A(\vec{\eta}) \end{pmatrix} = \frac{1}{4\eta_2^2} \begin{pmatrix} -2\eta_1\eta_2 \\ \eta_1^2 - 2\eta_2 \end{pmatrix}. \tag{3.10}$$

Note that for the Gaussian case, we have $\langle y \rangle_{p(y;\vec{\Phi}(\vec{w}))} = \mu$ and $\langle y^2 \rangle_{p(y;\vec{\Phi}(\vec{w}))} = \mu^2 + \sigma^2$. Now, based on (3.10), the inverse mapping $\vec{\Phi}$ can be computed in closed-form and is given by:

$$\vec{\Phi}(\vec{w}) = \frac{1}{2(w_2 - w_1^2)} \begin{pmatrix} 2w_1 \\ -1 \end{pmatrix}. \tag{3.11}$$

The Gamma distribution

The second two-parameter distribution that we investigate here is the Gamma distribution which its PDF, given the shape and rate parameters $\alpha, \beta > 0$, is given by:

$$p(y;\alpha,\beta) = \frac{\beta^\alpha y^{\alpha-1} e^{-\beta y}}{\Gamma(\alpha)}, \quad y \in (0,\infty).$$

The above equation can be further parameterized either w.r.t. the mean and variance parameters (μ and σ^2) of the Gamma distribution, or w.r.t. the natural parameters (η_1 and η_2), or even w.r.t. the mean value parameters (w_1 and w_2). For the case of natural parametrization, one can assume:

$$h(y) = 1, \qquad A(\vec{\eta}) = \log(\Gamma(\eta_2 + 1)) - (\eta_2 + 1)\log(-\eta_1)$$
$$\vec{\eta} = \begin{pmatrix} -\beta \\ \alpha - 1 \end{pmatrix}, \qquad \vec{T}(y) = \begin{pmatrix} y \\ \log(y) \end{pmatrix}.$$

Furthermore, based on the mean value parametrization in (3.7) and also Theorem 1, we have:

$$\vec{w} = \begin{pmatrix} \langle y \rangle_{p(y;\vec{\Phi}(\vec{w}))} \\ \langle \log(y) \rangle_{p(y;\vec{\Phi}(\vec{w}))} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial \eta_1} A(\vec{\eta}) \\ \frac{\partial}{\partial \eta_2} A(\vec{\eta}) \end{pmatrix} = \begin{pmatrix} -\frac{\eta_2+1}{\eta_1} \\ \psi(\eta_2 + 1) - \log(-\eta_1) \end{pmatrix} \tag{3.12}$$

where the Digamma function $\psi(\cdot)$ is defined by (Abramowitz and Stegun, 1972):

$$\psi(x) = \frac{d}{dx}\log(\Gamma(x)) = \log(x) - \frac{1}{2x} - \frac{1}{12x^2} + \dots. \tag{3.13}$$

Finally, in order to compute the function $\vec{\Phi}$, we require to solve the following system of equations:

$$\begin{cases} w_1 + \frac{\eta_2+1}{\eta_1} = 0 \\ w_2 - \psi(\eta_2 + 1) + \log(-\eta_1) = 0 \end{cases} \tag{3.14}$$

which is entangled by the Digamma function. Hence, as it can be seen, a closed-form analytic solution for the function $\vec{\Phi}$ is not available in this case. Nonetheless, in order to solve the above system of equations, we here approximate the Digamma function with its first two terms; i.e., we let $\psi(x) \approx \log(x) - \frac{1}{2x}$. In particular, this approximation yields reliable results and consequently the parameters of the Gamma distribution can be estimated with a good precision. However, one can also use more summations of (3.13) for a better approximation (see, e.g., Minka, 2002). For instance, it has been argued that the first six terms of the Digamma function can practically yield a very accurate approximation.
We then substitute $\eta_1 = -\frac{\eta_2+1}{w_1}$ from the first equation of (3.14) into the second one which results in:

$$\begin{aligned} w_2 &\approx \log(\eta_2 + 1) - \frac{1}{2(\eta_2 + 1)} - \log(\frac{\eta_2 + 1}{w_1}) \\ &= \log(\eta_2 + 1) - \frac{1}{2(\eta_2 + 1)} - \log(\eta_2 + 1) + \log(w_1) \end{aligned}$$

and thus

$$\frac{1}{\eta_2 + 1} \approx 2(\log(w_1) - w_2).$$

As a consequence, we obtain:

$$\eta_2 \approx \frac{1}{2(\log(w_1) - w_2)} - 1 \tag{3.15}$$

which by substituting into the equation $\eta_1 = -\frac{\eta_2+1}{w_1}$, yields:

$$\eta_1 \approx \frac{-1}{2w_1(\log(w_1) - w_2)}. \tag{3.16}$$

Therefore, the inverse mapping $\vec{\Phi}$ can be estimated by:

$$\vec{\Phi}(\vec{w}) \approx \frac{1}{2(\log(w_1) - w_2)} \begin{pmatrix} -1/w_1 \\ 1 - 2(\log(w_1) - w_2) \end{pmatrix}. \tag{3.17}$$

The Beta distribution

Finally, we discuss the details of the Beta distribution as another two-parameter distribution of the family. The PDF of the Beta, given the shape parameters $\alpha, \beta > 0$, is defined as follows:

$$p(y;\alpha,\beta) = \frac{y^{(\alpha-1)}(1-y)^{(\beta-1)}\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}, \quad y \in [0,1]. \tag{3.18}$$

Subsequently, we can use the following equations to obtain the natural parametrization of the Beta distribution:

$$h(y) = \frac{1}{y(1-y)}, \quad A(\vec{\eta}) = \log(\Gamma(\eta_1)) + \log(\Gamma(\eta_2)) - \log(\Gamma(\eta_1+\eta_2))$$
$$\vec{\eta} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \qquad \vec{T}(y) = \begin{pmatrix} \log(y) \\ \log(1-y) \end{pmatrix}.$$

Similar to the Gamma distribution, the PDF (3.18) can be reformulated in different ways resulting in the desired parametrization of the distribution. This includes parametrization w.r.t. the shape parameters (α and β) as in (3.18), w.r.t. the mean and variance parameters (μ and σ^2), the natural parameters (η_1 and η_2), or w.r.t. the mean value parameters (w_1 and w_2). It should be mentioned that the standard parametrization of the Beta distributions is to use shape parameters α and β, which is similar (mathematically) to the natural parametrization of the Beta (i.e. $\alpha = \eta_1$ and $\beta = \eta_2$). In general, the following relations exist between the mean and variance parameters of the Beta and its natural parameters:

$$\mu = \frac{\eta_1}{\eta_1+\eta_2}, \quad \sigma^2 = \frac{\eta_1\eta_2}{(\eta_1+\eta_2)^2(\eta_1+\eta_2+1)} \tag{3.19}$$

which subsequently results in:

$$\eta_1 = \frac{(1-\mu)\mu^2}{\sigma^2} - \mu, \quad \eta_2 = \frac{(1-\mu)^2\mu}{\sigma^2} - 1 + \mu. \tag{3.20}$$

Furthermore, based on the mean value parametrization in (3.7), one can define:

$$w_1 = \langle \log(y) \rangle_{p(y;\vec{\Phi}(\vec{w}))}, \quad w_2 = \langle \log(1-y) \rangle_{p(y;\vec{\Phi}(\vec{w}))} \tag{3.21}$$

where in the case that $\eta_1 \neq \eta_2$, there exists a bijective mapping $\vec{\Phi}$ as follows:

$$\vec{\Phi}(\vec{w}) = \vec{\eta} \quad \text{such that} \quad \vec{w} = (w_1, w_2)^T \text{ and } \vec{\eta} = (\eta_1, \eta_2)^T. \tag{3.22}$$

Observe that the assumption of $\eta_1 \neq \eta_2$ is essential for the existence of function $\vec{\Phi}$. This is because of the minimal representation property of the Beta distribution that mentioned above (see Wainwright and Jordan, 2008).

Now, to find the inverse mapping $\vec{\Phi}$, we should express the expected values $\langle \log(y) \rangle_{p(y;\vec{\Phi}(\vec{w}))}$ and $\langle \log(1-y) \rangle_{p(y;\vec{\Phi}(\vec{w}))}$ in terms of natural parameters $\vec{\eta}$. To this, using Theorem 1, we write:

$$\vec{w} = \begin{pmatrix} \langle \log(y) \rangle_{p(y;\vec{\Phi}(\vec{w}))} \\ \langle \log(1-y) \rangle_{p(y;\vec{\Phi}(\vec{w}))} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial \eta_1} A(\vec{\eta}) \\ \frac{\partial}{\partial \eta_2} A(\vec{\eta}) \end{pmatrix} = \begin{pmatrix} \psi(\eta_1) - \psi(\eta_1 + \eta_2) \\ \psi(\eta_2) - \psi(\eta_1 + \eta_2) \end{pmatrix}. \tag{3.23}$$

The above relation then results in solving the following system of equations:

$$\begin{cases} w_1 - \psi(\eta_1) + \psi(\eta_1 + \eta_2) = 0 \\ w_2 - \psi(\eta_2) + \psi(\eta_1 + \eta_2) = 0 \end{cases} \tag{3.24}$$

As it can be seen, a closed-form solution for the function $\vec{\Phi}$ is not available (similar to the Gamma distribution). Likewise, a straightforward approach would be to approximate the Digamma function ψ and then obtain η_1 and η_2 in terms of w_1 and w_2. However, for our experiments in Chapter 5, we will use another approach and directly solve the system of equations above using numerical methods. The details of this approach will be further discussed in Section A.3.

3.2 The Generative Model Description

Consider a set of N data points $Y = \{\vec{y}^{(1)}, \ldots, \vec{y}^{(N)}\}$ where each datum $\vec{y}^{(n)}$ is a vector with D entries. As latent variables $\vec{s}$ we consider H dimensional vectors with binary entries, $s_h \in \{0, 1\}$. The choice of discrete (here binary) latent variables can be seen as maintaining the previous research direction (Bornschein, Henniges, and Lücke, 2013; Lücke and Eggert, 2010; Puertas, Bornschein, and Lücke, 2010; Shelton et al., 2017; Lücke and Sahani, 2008). Concrete example algorithms will (for simplicity) assume these latents to be distributed according to H independent Bernoulli distributions. The analytical results derived in the following will, however, also apply for general binary latents $\vec{s}$. They could hence also be used in conjunction with more complex priors, e.g., priors given by deep models with binary variables such as SBNs, noisy-OR and so forth. In this aspect, the current study can extend the framework of previous works where mainly Bernoulli distributions are considered. Moreover, similar to the standard SC, we assume all observables to be independent and identically distributed (abbreviated as i.i.d.) but allow for any regular distribution of the exponential family according to the following generative model:

$$p(\vec{s}\,|\Theta) = \prod_{h=1}^{H} \pi_h^{s_h} (1-\pi_h)^{1-s_h} \tag{3.25}$$

$$p(\vec{y}\,|\,\vec{s}, \Theta) = \prod_{d=1}^{D} p\left(y_d; \vec{\eta}_d(\vec{s}, \Theta)\right) \tag{3.26}$$

$$\text{where} \quad p(y; \vec{\eta}) = h(y) \exp\left(\vec{\eta}^T \vec{T}(y) - A(\vec{\eta})\right) \tag{3.27}$$

and where Θ denotes the parameters of the model. For data model (3.25)-(3.27), the latent variable $\vec{s}$ couples to the observed variable y_d through the function $\vec{\eta}_d(\vec{s}, \Theta)$ which is defined further below (the function can be considered as a link function in a broader sense). In addition, for each $d = 1, \ldots, D$, we consider $\vec{\eta}_d(\vec{s}, \Theta)$ to be a vector with L entries, where L denotes the number of parameters corresponding to the noise distribution. For standard SCs, the latents set the *mean* of the observables using a weight matrix $M \in \mathbb{R}^{D \times H}$ (see Sections 2.2 and

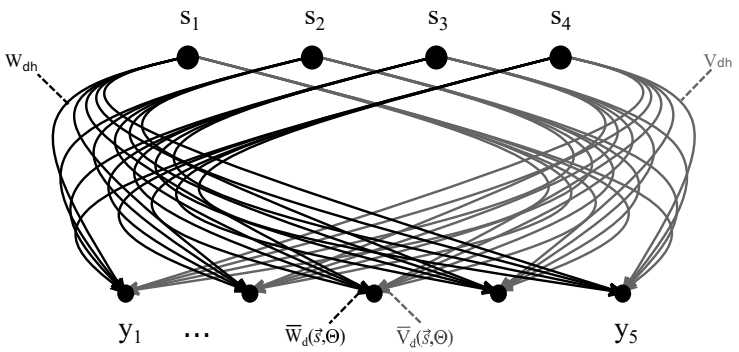


FIGURE 3.1: Proposed generative model with $H = 4$ hidden variables and $D = 5$ observed variables. Given a set of binary vector $\vec{s}$, the value y_d of the observables is conditionally independent and drawn from any regular distribution of the exponential family with mean value parameters $\bar{W}_d$ and $\bar{V}_d$ determined by weights $W_{d1}, \ldots, W_{d4}$ and $V_{d1}, \ldots, V_{d4}$, respectively.

2.3 for details). Here we will seek to couple latents and observables in an analog but more generally applicable way. To facilitate our notation, let us assume two-parameter distributions from now on (i.e. $L = 2$) as it applies, e.g., for Gaussian, Gamma or Beta distributions. Arbitrary L will be treated later on.

For the generative model presented above, we demand the distribution (3.26) to be parameterized w.r.t. the mean value parameters $\vec{w}$ (Equation 3.7), and therefore require to define the parameters of the distribution properly. For this purpose, we assume the $L = 2$ case and let $w_1 = \bar{W}$ and $w_2 = \bar{V}$. Parameters $\bar{W}$ and $\bar{V}$ depend on the latents $\vec{s}$ and *two* matrices W and V with $D \times H$ entries describing the mean value components (we will later show that the choice of a linear or non-linear superposition will be embedded to the model using the definitions of $\bar{W}$ and $\bar{V}$). The notation serves for gaining some intuition because, e.g. for Gaussian or Gamma distributions, $\bar{W}$ may denote the mean parameter and $\bar{V}$ the variance parameter (or the second moment of the distribution). Also Figure 3.1 illustrates the structure of the proposed model where, in the case of $L = 2$, two matrices connect the latent variables to the observables (compare it with Figure 2.2 where only one weight matrix is used). Then, using the function $\vec{\Phi}(\vec{w})$ given by (3.9), we define the link from latents to observables as follows:

$$\vec{\eta}_d(\vec{s}, \Theta) := \vec{\Phi}\big(\bar{W}_d(\vec{s}, \Theta), \bar{V}_d(\vec{s}, \Theta)\big), \quad d = 1, \ldots, D. \tag{3.28}$$

The mapping $\vec{\Phi}$, as mentioned before, is specific to the choice of noise distribution (3.26) which is a member of the exponential family distributions. Although a trivial closed-form formulation of the function is only available for some instances of the family, the function is definite as long as the noise distribution (3.26) has a minimal representation property (the assumption that we consider here). Thus, such a general link function is well-defined for the proposed data model using $\bar{W}(\vec{s}, \Theta)$ and $\bar{V}(\vec{s}, \Theta)$. For the case of arbitrary L, however, we define this link using L parameters and subsequently L matrices (see Section 3.5 further below). To complete the definition of our generative models, it only remains to define the functions $\bar{W}(\vec{s}, \Theta)$ and $\bar{V}(\vec{s}, \Theta)$ given a non-linear superposition. Before that, we elaborate our notation above using a preeminent example of the exponential family distributions:

Example 1. *To provide some intuition for the link defined by (3.28), consider the model with Gaussian observation noise. In the Gaussian case, based on* (3.10)-(3.11)*, we can write:*

$$\vec{\Phi}(\vec{w}) = \frac{1}{2(w_2 - w_1^2)} \begin{pmatrix} 2w_1 \\ -1 \end{pmatrix}. \tag{3.29}$$

By using definition (3.28), the coupling of latents to observables is consequently given by:

$$\begin{aligned}\vec{\bar{\eta}}_d(\vec{s},\Theta) &= \vec{\Phi}\big(\bar{W}_d(\vec{s},\Theta),\bar{V}_d(\vec{s},\Theta)\big) \\ &= \frac{1}{2\big(\bar{V}_d(\vec{s},\Theta)-\bar{W}_d^2(\vec{s},\Theta)\big)}\begin{pmatrix} 2\,\bar{W}_d(\vec{s},\Theta) \\ -1 \end{pmatrix}.\end{aligned} \tag{3.30}$$

In order to recover standard linear SC, we can complete the definition of $\vec{\bar{\eta}}_d(\vec{s},\Theta)$ using parameters $\sigma^2 \in \mathbb{R}^+$ and $W \in \mathbb{R}^{D\times H}$ and setting:

$$\bar{W}_d(\vec{s},\Theta) = \sum_h W_{dh}s_h \quad \textit{and} \quad \bar{V}_d(\vec{s},\Theta) = \sigma^2 + \bar{W}_d^2(\vec{s},\Theta) \tag{3.31}$$

which results in:

$$\vec{\bar{\eta}}_d(\vec{s},\Theta) = \begin{pmatrix} (W\vec{s})_d/\sigma^2 \\ -1/2\sigma^2 \end{pmatrix}. \tag{3.32}$$

That is, we recover the standard SC parametrization with $\mu_d = (W\vec{s})_d$ as the mean of observable d, and σ^2 as its variance (with σ^2 being the same for all d). Observe that by setting $W = M$ (and also $\pi_h = \pi$ for all h values), the current model represents the BSC model presented in Section 2.2.

3.2.1 A non-linear superposition model

In Example 1, given a matrix $W \in \mathbb{R}^{D\times H}$, the mean μ_d of observable d was given by $(W\vec{s})_d$. Our definition of the link from latents to observables (3.28) is sufficiently flexible to also allow for other types of superpositions that can be defined by the functions $\bar{W}_d(\vec{s},\Theta)$ or $\bar{V}_d(\vec{s},\Theta)$. In this respect, the question is what aspects of a given observable distribution these two functions shall determine. Most commonly the latents determine the *mean* of observable d using a linear superposition (e.g., for p-PCA, sparse coding etc). For many members of the exponential family, the first mean value parameter coincides with the mean of the corresponding exponential family distribution (that is we have $T_1(y) = y$ for a random variable y). Note that without loss of generality, we can always assume that the value of y appears at the first position of $\vec{T}(y)$ (if such a value exists among the elements of $\vec{T}$). Because otherwise, we can reorder the parameters of the model and let $T_1(y) = y$. If this is the case, the function $\bar{W}_d(\vec{s},\Theta)$ alone directly determines the observable mean:

$$\begin{aligned}\mu_d &= \big\langle y \big\rangle_{p(y;\vec{\bar{\eta}}_d(\vec{s},\Theta))} \\ &= \big\langle y \big\rangle_{p(y;\vec{\Phi}(\bar{W}_d(\vec{s},\Theta),\bar{V}_d(\vec{s},\Theta)))} \\ &= \big\langle T_1(y) \big\rangle_{p(y;\vec{\Phi}(\bar{W}_d(\vec{s},\Theta),\bar{V}_d(\vec{s},\Theta)))} \\ &= \bar{W}_d(\vec{s},\Theta)\end{aligned} \tag{3.33}$$

where the last step follows from (3.9). The function $\bar{W}_d(\vec{s},\Theta)$ can then be chosen as a linear superposition $\bar{W}_d(\vec{s},\Theta) = (W\vec{s})_d$ as in Example 1 or as a non-linear superposition. Note that in Sections 2.2 and 2.3, we used matrix M (denoting the weight matrix) to set the mean of observables with a linear (BSC) or a non-linear (MCA) superposition; i.e., we used the notation $\bar{M}_d(\vec{s},\Theta) = (M\vec{s})_d$ to refer to such a linear superposition. Although our definition of matrix W here is different from matrix M (we assumed W to denote the components of the first mean value parameter), the two matrices become identical if $T_1(y) = y$.

In the following, we will follow previous work (Lücke and Sahani, 2008; Bornschein, Henniges, and Lücke, 2013; Sheikh et al., 2019) and demand that the mean of a given observable

distribution is given by a maximum instead of the sum. More concretely, we will replace (e.g., for Gaussian or Gamma distributions):

$$\bar{W}_d(\vec{s},\Theta) = (W\vec{s})_d = \sum_h W_{dh}s_h \quad \text{by} \quad \bar{W}_d(\vec{s},\Theta) = \max_h\{W_{dh}s_h\}. \tag{3.34}$$

Our running example of the causes/symptoms relations for different diseases made the use of a maximum function to link the latents and observables relatively obvious. At the same time and importantly for the purposes of this study, the specific properties of the maximum function enable derivations of generally applicable update equations that can be used for any regular distribution of the exponential family.

Equation (3.34) would cover members of the exponential family such as Gaussian, Gamma etc (as $T_1(y) = y$ holds for these distributions). However, to be applicable to the whole exponential family, the definition of the maximum superposition has to be generalized (also the parameter $\bar{V}(\vec{s},\Theta)$ should be defined here). The challenge is that, in general, the sufficient statistics $\vec{T}(y)$ may not have an element that is proportional to y (the Beta distribution is one example where $T_1(y) = \log(y)$ and $T_2(y) = \log(1-y)$). As a consequence, the definition of the maximum superposition has to involve both functions $\bar{W}_d(\vec{s},\Theta)$ and $\bar{V}_d(\vec{s},\Theta)$. Nevertheless, it will be possible to define $\bar{W}_d(\vec{s},\Theta)$ and $\bar{V}_d(\vec{s},\Theta)$ such that the mean of observables is determined by a maximum superposition. For this purpose, consider a weight matrix $M(\Theta)$ with $D \times H$ entries which can potentially be a non-trivial function of the parameters Θ. Given a latent vector $\vec{s}$, we now demand that the mean μ_d of an observable d is given by:

$$\mu_d = \langle y\rangle_{p(y;\vec{\Phi}(\bar{W}_d(\vec{s},\Theta),\bar{V}_d(\vec{s},\Theta)))} \stackrel{!}{=} \max_h\{M_{dh}(\Theta)\, s_h\}. \tag{3.35}$$

Clearly, if $T_1(y) = y$, we can satisfy the demand by choosing $M_{dh}(\Theta) = W_{dh}$ and further setting $\bar{W}_d(\vec{s},\Theta) = \max_h\{W_{dh}s_h\}$. From (3.33), we can consequently obtain:

$$\begin{aligned}\mu_d &= \langle y\rangle_{p(y;\vec{\Phi}(\bar{W}_d(\vec{s},\Theta),\bar{V}_d(\vec{s},\Theta)))}\\ &= \bar{W}_d(\vec{s},\Theta)\\ &= \max_h\{W_{dh}s_h\}\\ &= \max_h\{M_{dh}(\Theta)\, s_h\}\end{aligned} \tag{3.36}$$

which fulfils (3.35). For the general case, nonetheless, the definition of $M(\Theta)$ which is applicable for the whole exponential family has to be more elaborate. So far, our derivation relied on the sufficient statistic $T_1(y)$ being equal to y. As mentioned before, we consider two $D \times H$ matrices W and V as part of the model parameters Θ, i.e. $\Theta = (\vec{\pi}, W, V)$. We then define $M(\Theta)$ as follows:

$$\forall d,h: \quad M_{dh}(\Theta) := F(W_{dh},V_{dh}) \quad \text{where} \quad F(w,v) = \langle y\rangle_{p(y;\vec{\Phi}(w,v))}. \tag{3.37}$$

Using matrix $M(\Theta)$ we now define our functions $\bar{W}_d(\vec{s},\Theta)$ and $\bar{V}_d(\vec{s},\Theta)$ as follows:

$$\begin{aligned}&\bar{W}_d(\vec{s},\Theta) := W_{dh(d,\vec{s},\Theta)}, \quad \bar{V}_d(\vec{s},\Theta) := V_{dh(d,\vec{s},\Theta)}\\ \text{where} \quad &h(d,\vec{s},\Theta) := \text{argmax}_h\{M_{dh}(\Theta)\, s_h\}.\end{aligned} \tag{3.38}$$

That is, we define functions $\bar{W}_d(\vec{s},\Theta)$ and $\bar{V}_d(\vec{s},\Theta)$ via an index $h(d,\vec{s},\Theta)$ which selects for a given observable d the latent with the maximal value of $M_{dh}(\Theta)$. Definitions (3.37) and (3.38) represent a generalization of the superposition model (3.36) suitable for the whole exponential family. To see this, we show that these definitions in fact fulfil (3.35):

$$\begin{aligned}
\mu_d &= \langle y \rangle_{p(y;\vec{\eta}_d(\vec{s},\Theta))} \\
&= \langle y \rangle_{p(y;\bar{\Phi}(\bar{W}_d(\vec{s},\Theta),\bar{V}_d(\vec{s},\Theta)))} \\
&= F(\bar{W}_d(\vec{s},\Theta),\bar{V}_d(\vec{s},\Theta)) \\
&= F(W_{dh(d,\vec{s},\Theta)},V_{dh(d,\vec{s},\Theta)}) \\
&= M_{dh(d,\vec{s},\Theta)}(\Theta) \\
&= \max_h\{M_{dh}(\Theta)\, s_h\}. \qquad (3.39)
\end{aligned}$$

In virtue of (3.37) and (3.38), we do *not* have to require that there exists a sufficient statistic $T_1(y) = y$, so the definition applies for all exponential family distributions. If $T_1(y) = y$, we get $F(w,v) = w$. Consequently, we drop back to the more trivial case of $M_{dh}(\Theta) = W_{dh}$. To our best knowledge, such a general formulation for a link function has not been considered before. While the dominance of one cause has been modelled by the maximum function as for previous MCA approaches (Lücke and Sahani, 2008; Lücke and Eggert, 2010; Bornschein, Henniges, and Lücke, 2013), modelling the dominance of the same cause also for the variance (or for the second moment of the distribution) has not been used (to our best knowledge) neither for the maximum combination nor for linear or any other combinations. Hence, definitions (3.37) and (3.38) can be considered as the first to link both mean and variance components based on the maximal cause.

The definition of functions $\bar{W}_d(\vec{s},\Theta)$ and $\bar{V}_d(\vec{s},\Theta)$ are relatively technical here such that we can accurately define the family of generative models and subsequently a set of update equations for the models' parameters which will be discussed in the next chapter. They define a link $\vec{\bar{\eta}}_d(\vec{s},\Theta)$ from latents to observables for the proposed models which is a consistent generalization of the maximum non-linearity (3.36) used, e.g., for Poisson (Lücke and Sahani, 2008) and Gaussian (Lücke and Eggert, 2010) observables. In this case, nevertheless, the proposed link remains consistently defined for all noise distributions of the exponential family. Independently of the choice of observables' distribution (the noise distribution), the link $\vec{\bar{\eta}}_d(\vec{s},\Theta)$ defined by (3.37) and (3.38) ensures that the latents change the *mean* of the observables always according to the maximum superposition model. This is in contrast to the previous LVMs for exponential family distributions such as EF-PCA (Collins, Dasgupta, and Schapire, 2002), Bayesian EF-PCA (Mohamed, Ghahramani, and Heller, 2008) and EF-SC (Lee et al., 2009) where the latents set the natural parameters of the observables using a linear superposition model (we will further elaborate on these studies in the upcoming section).

Equations (3.37) and (3.38) finalize the definition of the proposed family of generative models. The complete model family is defined by (3.25)-(3.27) and link (3.28) with (3.37) and (3.38). In analogy to previous models defined using the maximum non-linearity (Lücke and Sahani, 2008), we will refer to the family of models as *exponential family MCA* (ef-MCA). While the ef-MCA data models are very general, we will later see that the chosen parametrization results in generic equations for parameter updates. Before that, let us again consider the Gaussian distribution and explain a little more.

Example 2. *Consider the Gaussian case of Example 1. For the corresponding ef-MCA model (Gaussian-MCA in this case), the matrix $M(\Theta)$ is (because of $T_1(y) = y$) given by $M_{dh}(\Theta) = W_{dh}$ and*

$$\begin{aligned}
\bar{W}_d(\vec{s},\Theta) &= W_{dh(d,\vec{s},\Theta)} = \max_h\{W_{dh}s_h\} \qquad (3.40) \\
\bar{V}_d(\vec{s},\Theta) &= V_{dh(d,\vec{s},\Theta)}.
\end{aligned}$$

Using (3.9) with $\vec{\Phi}$ given by (3.11), we again obtain $\vec{\tilde{\eta}}_d(\vec{s},\Theta)$ as given by (3.30) but this time with $\bar{W}_d(\vec{s},\Theta)$ and $\bar{V}_d(\vec{s},\Theta)$ as in (3.40). In terms of the standard Gaussian parametrization, we would thus obtain as noise model:

$$p(y_d; \vec{\tilde{\eta}}_d(\vec{s},\Theta)) = \mathcal{N}(y_d; \bar{W}_d(\vec{s},\Theta), \bar{V}_d(\vec{s},\Theta) - \bar{W}_d^2(\vec{s},\Theta)).$$

The matrix V thus allows for the latents to also parameterize the variance. We will later see how this parametrization can also be changed back to the standard parameterization using a variance matrix $\Sigma^2_{dh}(\Theta) = V_{dh} - W^2_{dh}$ instead of V_{dh} (which is a more familiar parametrization). Note, however, the transition from a scalar variance (as usually used) to a matrix variance. Such a generalization, to the best of our knowledge, has not been considered before and the data model presented in (3.25)-(3.27) together with (3.28), (3.37) and (3.38) is the first non-linear LVM which allows for training of two matrices (one corresponding to the mean of observables, W, and another which parameterized the variance, V). This feature makes the data model more flexible compared to previous approaches (e.g. Lücke and Eggert, 2010; Puertas, Bornschein, and Lücke, 2010) and provides more information about the given dataset. If we wanted to enforce scalar variances (as conventionally used), we would define $\bar{V}_d(\vec{s},\Theta)$ as in (3.31).

3.2.2 Relation to previous work

Gaussian distributions do not model many types of data well. It has therefore been the goal of many previous LVMs to generalize learning algorithms developed for the Gaussian case to other distributions or to exponential family distributions (Collins, Dasgupta, and Schapire, 2002; Mohamed, Ghahramani, and Heller, 2008; Lee et al., 2009; Mohamed, Heller, and Ghahramani, 2010; Mohamed, Heller, and Ghahramani, 2011; Giryes and Elad, 2014; Lu, Huang, and Qian, 2016; Zhou et al., 2012; Salmon et al., 2014). As an important reference for this study, the approach of exponential family sparse coding (EF-SC) (Lee et al., 2009) defines a SC approach for noise distributions of the exponential family. The work chooses a link function that assumes the natural parameters of an exponential family distribution to be given by a linear superposition of the generative fields. Essentially, if $\tilde{\eta}_d(\vec{s},\Theta)$ is the natural parameter of a one-parameter distribution, then $\tilde{\eta}_d(\vec{s},\Theta) = (W\vec{s})_d$ (in Lee et al., 2009 W is denoted by B). With a link defined in this way and a standard Laplacian prior, the posteriors of the model maintain a mono-modal shape such that a maximum a-posteriori (MAP) approximation can be applied (Lee et al., 2009). However, MAP-based training of SC is usually restricted to the generative fields; like for standard SC, neither the prior parameters nor other parameters of the noise model are inferred. Cross-validation may be used in addition but is only feasible for very few additional parameters. Furthermore, setting the natural parameters of a given exponential family distribution using a linear superposition may (while mathematically convenient) not capture the true data generating process well. Depending on the distribution, a linear superposition may divert substantially from the standard choice of setting the distribution's mean. If we use Poisson noise as an elementary one-parameter example, then the natural parameter is given by $\eta = \log(\lambda)$, where λ is the mean (see Section 3.1.2 for details). Exponential family SC according to (Lee et al., 2009) would then use $\lambda_d = \exp\left(\sum_h W_{dh} s_h\right)$ for an observable d. The same choice of coupling was used more recently for the Poisson PCA approach (Salmon et al., 2014) which does focus on Poisson noise and reported state-of-the-art denoising performance when it was first suggested (also see Giryes and Elad, 2014). Other distributions of the exponential family will result in other types of non-linearities if the natural parameters are assumed to be set by a linear summation.

Exponential family sparse coding (EF-SC) (Lee et al., 2009) and Poisson PCA (Salmon et al., 2014) both borrow their link from latents to observables from the previously investigated

exponential family PCA (EF-PCA) approach (Collins, Dasgupta, and Schapire, 2002). In addition to the same link, EF-PCA uses (like Poisson PCA) a deterministic optimization for parameter learning. The original EF-PCA work only considers one-parameter distributions of the exponential family with sufficient statistics proportional to y (Collins, Dasgupta, and Schapire, 2002, Equation 2), which would exclude the Pareto, Laplace and other one-parameter distributions. A fully Bayesian approach to exponential family PCA (Mohamed, Ghahramani, and Heller, 2008) also uses the linear superposition to set the natural parameters but replaces MAP-based learning with a Hybrid Monte Carlo approach (which also allows for learning of other parameters than the weight matrix). Bayesian EF-PCA is formulated for distributions of the exponential family with one and several parameters but the concrete algorithm uses distributions of the family with a sufficient statistic equal to the identity function (Mohamed, Ghahramani, and Heller, 2008, Equation 7). EF-SC is more general in its initial formulation (Lee et al., 2009). However, the used generative fields are all defined to set one-parameter distributions (their matrix B sets the natural parameters of an exponential family distribution). Also for the Gaussian case, which is used as an introductory example, a single-parameter Gaussian (with known variance) is chosen. Notably, none of the above mentioned previous approaches (i.e., Poisson PCA (Salmon et al., 2014), Poisson factor analysis (Zhou et al., 2012), EF-PCA (Collins, Dasgupta, and Schapire, 2002), Bayesian EF-PCA (Mohamed, Ghahramani, and Heller, 2008) and EF-SC (Lee et al., 2009)) used any other than one-parameter distributions of the exponential family in their contributions. Moreover, and to our best knowledge, the numerical experiments of those contributions (including EF-SC) only used two specific one-parameter distributions: Bernoulli and Poisson. Although, e.g., EF-SC would in principle be able to treat other distributions, the meaning of setting all natural parameters using a linear superposition of latents is arguably unclear. For instance, even for a Gaussian (as a standard two-parameter distribution) the second moment would be determined by two linear sums. Such a case (and more intricate cases resulting from other two-parameter distributions) have not been discussed or treated by these approaches. One example of studies which did investigate other distributions than Bernoulli and Poisson is (Khan et al., 2010) where a generalized mixture of factor analyzers is presented that cope with both continuous and discrete data (Gaussian and Multinomial distributions have been treated). Further, a new variational EM algorithm is established in this contribution based on a lower bound of the Multinomial likelihood for fitting the model to the considered mixed data. Although the model can, in principle, deal with binary and categorical data along with the continuous case, it is not defined to subsume all distributions of the exponential family.

In addition, note that approaches such as EF-SC are, furthermore, restricted to other aspects of learning. As a deterministic MAP-based learning is used, only weight parameters are updated (and neither additional noise nor prior parameters). It may thus be argued that no LVM has been applied so far (at least for practical optimizations) to the whole exponential family. Once a model is defined, the challenge is, of course, the derivation of a general parameter optimization procedure. In our study here, we therefore attempt to address the goal of model definition and parameter optimization using a non-linear superposition model to couple latents to observables. All approaches discussed above maintain a linear superposition assumption; and when a general formulation for the exponential family is aimed at, the Gaussian case is generalized by using the linear superposition to set the natural parameters. The non-linear superposition we introduce here can also be seen as a generalization of the Gaussian case but we will maintain that the used superposition sets the *mean* of a given observable distribution. Notably, for the specific choice of the Poisson distribution, approaches such as Poisson factor analysis (Zhou et al., 2012) have also chosen to couple the latents to the mean of distribution rather than to its natural parameter. For our non-linear superposition presented above, we also keep such a coupling to the distribution mean but the coupling is defined for the general case.

3.3 A Double-Dictionary Approach: SC with Mean and Variance Dictionaries

The presented family of data models (3.25)-(3.27) together with the link (3.28) and definitions (3.37) and (3.38) represent a non-linear SC model (due to the usage of Bernoulli distributions as the prior) which couples latents and observables using a novel maximum superposition. Given such a definition, the proposed models follow a double-dictionary approach (L-dictionary if the considered noise distribution has L parameters). Meaning that training an ef-MCA model results in having component means alongside component variances for a two-parameter distribution like Gaussian or Beta. That is, for each cause (e.g. disease), we will compute both mean and variance components corresponding to one specific observation (e.g. symptom). For the case of L-parameter distributions, this will be generalized to using L matrices corresponding to the parameters of the noise distribution. The model is consequently capable of extracting more statistical information from data which can be used, for instance, to estimate the distribution of the symptoms (observables) given each of the diseases (causes). We will later assess the performance of our models given such extra information on medical data of hearing impairments in Section 5.2.1.

Analysis of medical data is only one specific application of the proposed SC models. Originated from computational neuroscience, SC approaches have further been developed for many different applications including but not restricted to feature extraction, data decomposition, denoising, inpainting and video or audio processing. As a practical application, SCs have been successfully used for finding succinct representations of stimuli. In fact, it has been shown that the learned generative fields, obtained from applying a SC model to natural images, resemble the receptive fields of neurons in the primary visual cortex (V1) (see, e.g., Olshausen and Field, 1997; Olshausen and Field, 2004). In the seminal paper (Olshausen and Field, 1997), a conventional SC model with a linear superposition has been used to relate neural response properties of area V1 to the statistical properties of images. The SC approach studied in this work models the dependency between latent and observed variables using one weight matrix that contains the latents' generative fields. Another statistical model which has also been used to link the response properties of simple cells in V1 to the view of sensory systems as an optimal information encoder is Independent Component Analysis (ICA) (e.g., Hyvärinen and Oja, 1997; Hyvärinen and Oja, 2000). Likewise, the approach uses only one weight matrix to couple the latents and observables. Indeed, a scalar variance and a linear superposition are two important features of these two models which we argued to be statistically suboptimal.

Although both SC and ICA models achieved prominent results in analyzing the response properties of V1, the models are restricted to certain assumptions. Since their introduction, it has repeatedly been pointed out that generalizations of the original model assumptions are required. Such generalizations include, e.g., extensions of ICA which include an encoding of dependencies between latent activities, addition of intensity variables and latents, hierarchical features and many more (e.g. Karklin and Lewicki, 2003; Karklin and Lewicki, 2009; Wainwright and Simoncelli, 1999). Also the encoding of image components' variances has been observed to be important for image processing but has so far only been realized for mixture models (e.g., Zoran and Weiss, 2011).

In (Bornschein, Henniges, and Lücke, 2013), the impact of occlusion-like non-linearities on predicted simple cell responses is studied by exploiting a maximum superposition (in contrast to the linear superposition of the standard SCs). Moreover, the study leverages binary latents since a MCA model (presented in Section 2.3) is employed. In particular, binary latents suggest themselves for probabilistic neural encoding (Shivkumar et al., 2018), and further the maximum non-linearity of MCA can be motivated by statistical properties of stimuli processed by primary visual cortex (Puertas, Bornschein, and Lücke, 2010; Bornschein, Henniges, and Lücke, 2013) and auditory cortex (Sheikh et al., 2019). The investigated MCA approach has

then resulted in a variety of Gabor-like functions together with many globular generative fields after training on natural image patches. Nevertheless, the assumption of a single variance parameter (that is considered by the model) is suboptimal which suggests that the primary visual cortex may, likewise, represent component variances as well as component means. The proposed LVMs here can thus put one step forward and establish a new approach for coupling the latents and observables using two matrices.

Therefore, the data models (3.25)-(3.27) will be a suitable choice that can capture first- and second-order statistics of natural image patches. If response properties of V1 simple cells are not restricted to first-order statistics, the proposed models are likely to be more closely aligned with neural responses than standard SC or ICA models. We will further examine the performance of these generative models when applied to natural image patches for two different noise models: Gaussian and Beta. See Section 5.2.2 for the details of these two experiments.

3.4 Two Special Cases of the Proposed Link Function

Let us now briefly discuss two technical issues that one may encounter when applying the proposed non-linear link function defined in (3.38). We will explain the two problems and also the way we approach them as our proposed solutions. Each of these solutions will be then exploited and further examined in our experiments in Chapter 5.

3.4.1 The background: An extra cause for MCA models

One technical problem that may occur for the link function (3.38) is the case where none of the causes are active, i.e. $\vec{s} = (0,\ldots,0)^T$. Given our medical reasoning example, this represents the case where an examined patient is healthy (no specific disease is active). In particular, the linear summation of SC models computes the mean value to be equal to zero for this case. This is acceptable for the Gaussian distribution as the mean of Gaussian could be any real number (note that for the Gaussian case we have $M_{dh}(\Theta) \in \mathbb{R}$ for each d and h). For other distributions of the exponential family, however, this is not a valid argument. For instance, the mean of Beta distribution must be in the interval $(0,1)$ (i.e. $M_{dh}(\Theta) \in (0,1)$) and the mean of Gamma or Exponential is any value in $(0,\infty)$ (i.e. $M_{dh}(\Theta) \in (0,\infty)$). In addition, given the proposed link function (3.38), the maximum function would search for the highest value amongst different zeros (all $s_h M_{dh}(\Theta)$ values will be equal to zero). Therefore, when implementing the algorithm, any programming language would choose one arbitrary h as the maximum argument (argmax) which may not be desirable. This could cause certain anomalies in the code that affect the final results. The defined link function should thus be amended such that it can consume a zero vector as an activation.

To address this issue, we here consider a background cause (in addition to the H considered hidden causes) which combines with the corresponding active causes and provides (if all $s_h = 0$) non-zero mean and variance values. The background thus prevents degeneracies and models, in the case of medical data, the symptom statistics of healthy patients. We denote this cause by $s_B = 1$ that is much like how bias terms are usually modelled. Given latent variable $\vec{s}$, we then demand each of the matrices $M(\Theta), W$ and V to accept $D \times (H+1)$ entries such that the last column denotes the corresponding values of the background. For instance, $M(\Theta) = (\vec{M}_1(\Theta),\ldots,\vec{M}_H(\Theta),\vec{M}_B(\Theta))$ where the basis function $\vec{M}_B(\Theta)$ contains background parameters and corresponds to $s_B = 1$. We subsequently let $\vec{s} = (s_1,\ldots,s_H,s_B)^T$. Hence, we never face the case where all the causes are zero as the background is always active. Consequently, the link function (3.38) can be used without any difficulties because this time index $h(d,\vec{s},\Theta)$ will select the background value (B) for a given observable d if all the other causes are zero. Also observe that the background introduced here does not make

any changes for the combination of other causes. Because the corresponding background $M_{dB}(\Theta)$ assumed to learn the lowest values from the data and when other causes are active, the maximum function (3.38) chooses the one with the highest $M_{dh}(\Theta)$ value. Therefore, the proposed solution will only address the case where none of the causes are active and do not make any other changes. Figure 3.2 then illustrates the structure of the proposed LVMs (3.25)-(3.27) when the background model is also considered.

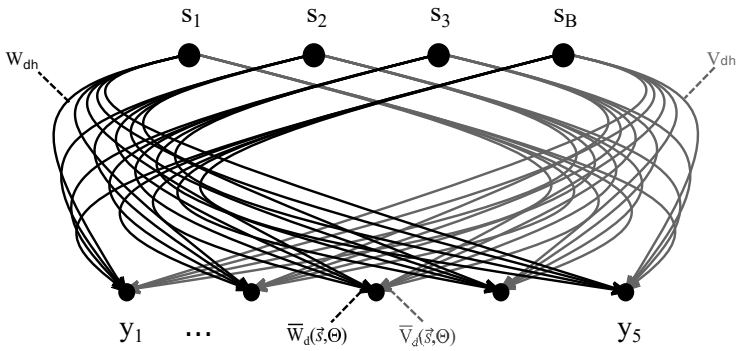


FIGURE 3.2: A generative model with $H = 3$ hidden variables together with background s_B and $D = 5$ observed variables. Given a set $\vec{s} = (s_1, s_2, s_3, s_B)$, the values y_d of the observed variables are then conditionally independent. Each value y_d is drawn from one of the distributions of the exponential family with the mean value parameters determined by parameters W_{d1}, W_{d2}, W_{d3} and W_{dB}, and also V_{d1}, V_{d2}, V_{d3} and V_{dB}. For a given binary vector $\vec{s}$, these parameters combine competitively according to the maximum function defined in (3.38).

This novel definition here not only eliminates the raised issue, but also improves the performance of the models to some extent. Importantly, when applied to medical data, the considered ef-MCA model can further learn a new generative field (corresponding to the background) which captures the essential information of a patient being healthy. For other datasets, the corresponding model can learn a common background which appears as the basis of all other generative fields. This could, for instance, correspond to the background of natural images. We will later show in Chapter 5 how such a background can be used in practice for different data types.

3.4.2 The maximum magnitude combination rule

Another issue that may occur when applying the maximum function (3.38) is subject to the ef-MCA models where the mean components ($M_{dh}(\Theta)$) accept both positive and negative values. Example is the Gaussian distribution where $M_{dh}(\Theta) \in \mathbb{R}$ for each d and h. For such cases, when the corresponding causes with both positive and negative weights are active, the maximum function seeks the maximal amongst positive values as they are mathematically higher and subsequently the cause which corresponds to such a positive value will be chosen by the link function (3.38). As a result, the model fails to correctly extract all statistical dependencies between different causes.

To address this issue, we here follow (Bornschein, Henniges, and Lücke, 2013) and use a *maximum magnitude* link function when the considered noise model encompasses both positive and negative mean values. Such a link function can be defined as:

$$h(d, \vec{s}, \Theta) := \text{argmax}_h \{ |M_{dh}(\Theta)\, s_h| \} \tag{3.41}$$

and the index $h(d, \vec{s}, \Theta)$ will be used for the definition of $\bar{W}_d(\vec{s}, \Theta)$ and $\bar{V}_d(\vec{s}, \Theta)$ in (3.38). This specific link function has been shown to accurately model the true non-linear relations

between the generative fields and yield reliable results (see, e.g., Bornschein, Henniges, and Lücke, 2013). We will also examine the effectiveness of this link function for the proposed generative models (e.g., when a Gaussian distribution is used) in Chapter 5.

3.5 Parametrization of the Proposed Generative Models – General Case

We now generalize the definition of the proposed family of generative models presented in (3.25)-(3.27) to any arbitrary L-parameter distribution. For this purpose, first note that the prior parameter $\vec{\pi}$ does not change by the definition of the noise distribution. That is, we always keep the independent Bernoulli distributions in (3.25). For the noise distribution given by (3.26) and (3.27), however, instead of two matrices W and V, we can in general consider L matrices (corresponding to the L parameters of the distribution). Let us denote these matrices by $W^{(1)}$ to $W^{(L)}$, i.e. $\Theta = (\vec{\pi}, W^{(1)}, \ldots, W^{(L)})$. Therefore, we can generalize our definition of $\vec{\bar{\eta}}_d(\vec{s}, \Theta)$, which is now given by:

$$\vec{\bar{\eta}}_d(\vec{s}, \Theta) := \vec{\Phi}\big(\bar{W}_d^{(1)}(\vec{s}, \Theta), \ldots, \bar{W}_d^{(L)}(\vec{s}, \Theta)\big). \tag{3.42}$$

Moreover, given matrices $W^{(1)}$ to $W^{(L)}$ with $D \times H$ entries, we have to define the mappings $\bar{W}_d^{(1)}(\vec{s}, \Theta)$ to $\bar{W}_d^{(L)}(\vec{s}, \Theta)$ for each observable d. In analogy to the $L = 2$ case, we do so by defining a matrix $M(\Theta)$ as follows:

$$\forall d, h: \quad M_{dh}(\Theta) := F(W_{dh}^{(1)}, \ldots, W_{dh}^{(L)}) \;\text{ with }\; F(\vec{w}) = \langle y \rangle_{p(y; \vec{\Phi}(\vec{w}))}. \tag{3.43}$$

Using the matrix $M(\Theta)$, we now define our mappings $\bar{W}_d^{(l)}(\vec{s}, \Theta)$ for $l = 1, \ldots, L$:

$$\forall l, d, h: \bar{W}_d^{(l)}(\vec{s}, \Theta) := W_{dh(d,\vec{s},\Theta)}^{(l)} \;\text{ with }\; h(d, \vec{s}, \Theta) := \operatorname{argmax}_h\{M_{dh}(\Theta)\, s_h\}. \tag{3.44}$$

Likewise, in the general case, definitions (3.43) and (3.44) guarantee that the *mean* of observable d is always given by $\mu_d = \max_h\{M_{dh}(\Theta)\, s_h\}$. The above equations then finalize the definition of ef-MCA data models where the noise distribution can now be any arbitrary regular distribution of the exponential family.

As the function $\vec{\bar{\eta}}_d(\vec{s}, \Theta)$ models the influence of latent variables on observed variables, it is reminiscent of the link functions as, e.g., defined for generalized linear regression. Furthermore, our definitions (3.42)-(3.44) ensure that the means μ_d of the observables are always given by a superposition defined by matrix $M(\Theta)$, which is likewise reminiscent of the link functions for non-linear regression. Our definition of $\vec{\bar{\eta}}_d(\vec{s}, \Theta)$ via $M(\Theta)$ is less direct as we use a maximum superposition, which notably differs from usual definitions of link functions (alongside other technical differences). The general role of coupling observables to latent variables (or response variables in regression) is also played by $\vec{\bar{\eta}}_d(\vec{s}, \Theta)$ such that it may also, in a broader sense, be referred to as a link function.

Chapter 4

Parameter Optimization

In the preceding chapter, we defined a family of non-linear generative models denoted by ef-MCA which considers a large variety of probability distributions as the noise model. In this manner, we first considered the two-parameter distributions ($L = 2$) of the exponential family and formulated the family of generative models using parameters $\Theta = (\vec{\pi}, W, V)$; and later generalized our models to L-parameter distributions of the family by considering $\Theta = (\vec{\pi}, W^{(1)}, \ldots, W^{(L)})$. We also introduced a non-linear superposition model that is appropriate for any regular distribution of the exponential family with a minimal representation property. We are now interested in a generally applicable approach for training the proposed family of generative models. In other words, we seek parameters Θ^* that, given a set of data points $Y = \{\vec{y}^{(1)}, \ldots, \vec{y}^{(N)}\}$, maximizes the likelihood of the data under the data models (3.25)-(3.27). To this, we first introduce the *Expectation-Maximization* (EM) algorithm and discuss its basic features. We then derive a set of update equations for the parameters Θ and present a reliable approach for training the family of generative models. Similar to the previous chapter, the two-parameter distributions are first treated as they are more intuitive.

The main part of this chapter, which is the derivation of parameter updates, is based on (Mousavi et al., 2020) (currently under review). This corresponds to Theorems 2, 3 and 4 which will be discussed and elaborated in Sections 4.2 and 4.3. I did the first preparations and proved the elementary version of the theorems and later, together with Jörg Lücke, we generalized and validated the proofs. Moreover, the content of Section 4.4 is based on (Lücke, 2019) which sets the foundation for the truncated variational EM algorithm and is further developed by, e.g., Guiraud et al. in (Guiraud, Drefs, and Lücke, 2018). Thanks to the black-box algorithm developed and introduced in these two contributions, we are able to train our models at large-scales without much effort. In addition, in Section 4.5, I will present some further analysis of the derived update equations which are not reported in our papers. This section can therefore be seen as the description of detailed discussions that I had with Jörg Lücke regarding the possible relations between the investigated parameter optimization algorithm and other approaches for parameter estimation.

4.1 Maximum Likelihood

We assume the set of data points $Y = \{\vec{y}^{(1)}, \ldots, \vec{y}^{(N)}\}$ to be drawn independently and identically from one of the regular distributions of the exponential family (to be i.i.d.). We then follow a standard maximum likelihood approach and seek parameters Θ^* that optimize the log-likelihood function given by:

$$\mathcal{L}(\Theta) = \sum_{n=1}^{N} \log\left(p(\vec{y}^{(n)}|\Theta)\right) = \sum_{n=1}^{N} \log\left(\sum_{\vec{s}} p(\vec{y}^{(n)}, \vec{s}\,|\Theta)\right) \tag{4.1}$$

where $\sum_{\vec{s}}$ denotes summation over all possible hidden states $\vec{s}$. Given a regular distribution p of the exponential family, the first step for finding the optimal parameters Θ^* is to compute

the derivative of the considered objective (the log-likelihood (4.1)) w.r.t. the parameters Θ. Afterwards, by equating the results to zero, we can obtain the critical points and desirably the parameters Θ^*. This is, however, a complicated task and produces intricate terms because of the summation over hidden states that appears within the log function. To address this issue, the well-known EM algorithm (Dempster, Laird, and Rubin, 1977) has been introduced which maximizes the log-likelihood function and has been widely used in previous studies. In contrast to, e.g., backpropagation which is based on gradient descent, the EM algorithm has been successfully applied for a large variety of LVMs including models with binary latents. Hence, the algorithm is suitable for our investigations here and we likewise follow previous works in using the EM algorithm to optimize the objective (4.1). To do so, let us first define the standard procedure of the EM algorithm in the following:

4.1.1 The expectation maximization algorithm

We apply an EM framework introduced by, e.g., (Neal and Hinton, 1998) and consider a variational distribution $q^{(n)}(\vec{s})$ corresponding to each data point $y^{(n)}$. We then restate (4.1) as follows:

$$\mathcal{L}(\Theta) = \sum_n \log\Big(\sum_{\vec{s}} q^{(n)}(\vec{s}) \frac{p(\vec{y}^{(n)}, \vec{s}\,|\Theta)}{q^{(n)}(\vec{s})}\Big). \tag{4.2}$$

Note that we take the assumption that $q^{(n)}(\vec{s})$ is in fact a probability density function defined over the latent parameter space, i.e., for each hidden state $\vec{s}$ we have $q^{(n)}(\vec{s}) \geq 0$ and $\sum_{\vec{s}} q^{(n)}(\vec{s}) = 1$. Now, using these assumptions and the fact that the log function is concave, we apply the Jensen's inequality[1] (Jensen, 1906) to get:

$$\begin{aligned}
\mathcal{L}(\Theta) &= \sum_n \log\Big(\sum_{\vec{s}} q^{(n)}(\vec{s}) \frac{p(\vec{y}^{(n)}, \vec{s}\,|\Theta)}{q^{(n)}(\vec{s})}\Big) \\
&\geq \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\Big(\frac{p(\vec{y}^{(n)}, \vec{s}\,|\Theta)}{q^{(n)}(\vec{s})}\Big) \\
&= \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \Big(\log\big(p(\vec{y}^{(n)}, \vec{s}\,|\Theta)\big) - \log\big(q^{(n)}(\vec{s})\big)\Big) \\
&= \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\big(p(\vec{y}^{(n)}, \vec{s}\,|\Theta)\big) - \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\big(q^{(n)}(\vec{s})\big) \\
&= \mathcal{F}(q, \Theta)
\end{aligned} \tag{4.3}$$

where $\mathcal{F}(q, \Theta)$ is known as the *free energy* function (Dempster, Laird, and Rubin, 1977; Neal and Hinton, 1998), and element q here denotes the set of all variational distributions $q^{(n)}$. Also note that in other terminologies, it is customary to name the log (marginal) probability of the observations as the *evidence* (see, e.g. Mcauliffe and Blei, 2008), and accordingly refer to $\mathcal{F}(q, \Theta)$ as the *Evidence Lower Bound* (ELBO). In this study, however, we mainly use the free energy term in order to be consistent with previous works but the term ELBO can also be used without any further assumption. In particular, the free energy $\mathcal{F}(q, \Theta)$ can be formally

[1] The Jensen's inequality states that for positive weights $\lambda_i, i = 1, \ldots, K$ where $K \in \mathbb{N}$ and $\sum_i \lambda_i = 1$, we have:

$$\log\Big(\sum_{i=1}^{K} \lambda_i x_i\Big) \geq \sum_{i=1}^{K} \lambda_i \log(x_i), \quad \forall x_i \neq 0.$$

defined by:

$$\mathcal{F}(q,\Theta) := \sum_n \langle \log\left(p(\vec{y}^{(n)},\vec{s}\,|\Theta)\right)\rangle_{q^{(n)}} - \sum_n \langle \log\left(q^{(n)}(\vec{s})\right)\rangle_{q^{(n)}} \tag{4.4}$$

where the second term is the so-called *Shannon entropy* $\mathcal{H}(q)$; that is:

$$\mathcal{H}(q) := -\sum_n \langle \log\left(q^{(n)}(\vec{s})\right)\rangle_{q^{(n)}}. \tag{4.5}$$

In the EM algorithm we consider here, instead of maximizing the log-likelihood directly, we maximize the free energy (4.4) as a lower bound of the log-likelihood. In practice, we have observed that this approach consistently maximizes both free energy and log-likelihood functions to at least a local optima. Given an ef-MCA data model presented in (3.25)-(3.27), the free energy can then be restated as follows:

$$\mathcal{F}(q,\Theta) = \sum_{n,\vec{s}} q^{(n)}(\vec{s})\{\sum_d \log\left(p(y_d^{(n)};\vec{\eta}_d(\vec{s},\Theta))\right) + \sum_h \log(p(s_h|\Theta))\} + \mathcal{H}(q) \tag{4.6}$$

which should be maximized w.r.t. the two sets of parameters q and Θ. We do so by first maximizing the free energy function (4.6) w.r.t. the distributions q while keeping parameters Θ fixed and next, w.r.t. the model parameters Θ while keeping distributions q fixed. The former step is known as the *Expectation step* (or simply the E-step) and the latter referred to as the *Maximization step* (or the M-step). To give you an insight of the considered algorithm, let us briefly describe the complete procedure of the EM algorithm: We initialize parameters Θ and given these parameters, compute the optimal distributions q in the E-step. Then having the distributions q, we update parameters Θ of the model in the M-step. These two steps will be iteratively applied since convergence and, at the convergence point, we consider the computed parameters Θ as the optimal parameters of the model (i.e., Θ^*). Consequently, the terms 'E-step' and 'M-step' refer to the two optimization steps of the expectation maximization algorithm (Dempster, Laird, and Rubin, 1977; Neal and Hinton, 1998).

The E-step

Given the model parameters Θ, we aim at finding distributions q that maximizes the free energy (4.6). For this purpose, we first define the well-known *Kullback–Leibler (KL) divergence*.

Definition 1. *The KL divergence (D_{KL}), also known as the relative entropy, is a function that measures the difference between two probability distributions. In discrete case, for instance, the KL divergence can be defined as:*

$$D_{KL}(Q\|P) := \sum_{x\in\mathcal{X}} Q(x)\log\left(\frac{Q(x)}{P(x)}\right) \tag{4.7}$$

where P and Q are probability distributions over the same probability space $\mathcal{X}$.

One important property of the KL-divergence is the non-negativity, i.e. $D_{KL}(Q\|P) \geq 0$. In fact, using the *Gibbs' inequality* [2](Brémaud, 2012) one can easily show that $D_{KL}(Q\|P) \geq 0$

[2]For any two probability distributions $\{p_1,\ldots,p_k\}$ and $\{q_1,\ldots,q_K\}$ (for $K \in \mathbb{N}$), the Gibbs' inequality states that the following inequality holds:

$$-\sum_{i=1}^{K} p_i \log(p_i) \leq -\sum_{i=1}^{K} p_i \log(q_i)$$

and with equality if and only if $p_i = q_i$ for all i.

and the equality holds ($D_{KL}(Q\|P) = 0$) if and only if $P(x) = Q(x)$ for all $x \in \mathcal{X}$. In other words, the KL-divergence achieves its minimum at the point $P(x) = Q(x)$. Now, for each n, we replace Q and P by the two probability distributions $q^{(n)}(\vec{s})$ and $p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)$ (the posterior), respectively. We can consequently obtain:

$$\begin{aligned} D_{KL}\big(q^{(n)}(\vec{s})\|p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)\big) &= \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\Big(\frac{q^{(n)}(\vec{s})}{p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)}\Big) \\ &= \sum_{\vec{s}} q^{(n)}(\vec{s})\Big(\log\big(q^{(n)}(\vec{s})\big) - \log\big(p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)\big)\Big) \\ &= \sum_{\vec{s}} q^{(n)}(\vec{s})\Big(\log\big(q^{(n)}(\vec{s})\big) - \log\big(\frac{p(\vec{s},\vec{y}^{(n)}|\Theta)}{\sum_{\vec{s}'} p(\vec{s}',\vec{y}^{(n)}|\Theta)}\big)\Big) \\ &= \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\big(q^{(n)}(\vec{s})\big) - \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\big(p(\vec{s},\vec{y}^{(n)}|\Theta)\big) \\ &+ \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\big(p(\vec{y}^{(n)}|\Theta)\big) \end{aligned} \tag{4.8}$$

where in the last equation we used the fact that $\sum_{\vec{s}'} p(\vec{s}',\vec{y}^{(n)}|\Theta) = p(\vec{y}^{(n)}|\Theta)$. Then, since the above equation is true for each $n = 1,\ldots,N$, we can write:

$$\begin{aligned} \sum_n D_{KL}\big(q^{(n)}(\vec{s})\|p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)\big) &= \sum_n \Big(\sum_{\vec{s}} q^{(n)}(\vec{s}) \log\big(q^{(n)}(\vec{s})\big) \\ &- \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\big(p(\vec{s},\vec{y}^{(n)}|\Theta)\big) + \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\big(p(\vec{y}^{(n)}|\Theta)\big)\Big) \\ &= \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\big(q^{(n)}(\vec{s})\big) - \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\big(p(\vec{s},\vec{y}^{(n)}|\Theta)\big) \\ &+ \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\big(p(\vec{y}^{(n)}|\Theta)\big) \\ &= \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\big(q^{(n)}(\vec{s})\big) - \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \log\big(p(\vec{s},\vec{y}^{(n)}|\Theta)\big) \\ &+ \sum_n \log\big(p(\vec{y}^{(n)}|\Theta)\big) \underbrace{\sum_{\vec{s}} q^{(n)}(\vec{s})}_{=1} \\ &= -\mathcal{F}(q,\Theta) + \sum_n \log\big(p(\vec{y}^{(n)}|\Theta)\big). \end{aligned} \tag{4.9}$$

Hence, considering $\mathcal{L}(\Theta)$ to be given by (4.1), the free energy $\mathcal{F}(q,\Theta)$ can be written as:

$$\mathcal{F}(q,\Theta) = \mathcal{L}(\Theta) - \sum_n D_{KL}\big(q^{(n)}(\vec{s})\|p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)\big). \tag{4.10}$$

As the function $\mathcal{L}(\Theta)$ is independent of q, finding a variational distribution $q^{(n)}$ that maximizes the free energy $\mathcal{F}(q,\Theta)$ is equivalent to finding $q^{(n)}$ that minimizes the KL divergence, that is:

$$\text{argmax}_{q^{(n)}}\mathcal{F}(q,\Theta) = \text{argmin}_{q^{(n)}} \sum_n D_{KL}\big(q^{(n)}(\vec{s})\|p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)\big). \tag{4.11}$$

On the other hand, we mentioned above that (using the Gibbs' inequality) D_{KL} achieves its minimum (at zero) if and only if $q^{(n)}(\vec{s}) = p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)$. Therefore, we can conclude that the E-step is to compute the posteriors $p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)$ for each hidden state $\vec{s}$ and each data point $\vec{y}^{(n)}$ and further equating them to distributions $q^{(n)}(\vec{s})$. For tractable models, computing

these posteriors are feasible and we can easily perform our E-steps. However, for large-scale models, such computations become very expensive (because of the summations over all hidden states $\vec{s}$ which by increasing the value of H becomes intractable). Therefore, we require to exploit an approximation method to compute the posteriors $p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)$ (for each n and $\vec{s}$) or equivalently the variational distributions $q^{(n)}(\vec{s})$ in each E-step. We will later discuss in Section 4.4 how truncated variational schemes can be applied for such cases.

Finally, observe that at the point of maximum, i.e. $q^{(n)}(\vec{s}) = p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)$, the D_{KL} function is zero and thus the free energy will be equal to the log-likelihood of the data. Hence, we can infer that the E-step will always increase the free energy (4.6) and consequently the log-likelihood function.

The M-step

After computing the posteriors in the E-step, we aim at maximizing the free energy (4.6) w.r.t. the model parameters Θ in the M-step. To this, we set the derivatives of the free energy w.r.t. all parameters to zero (i.e. we let $\nabla_\Theta \mathcal{F} = 0$) and obtain the corresponding update equations for the set of parameters Θ. These equations will be then used to update parameters Θ in each M-step.

Nonetheless, the central challenge we have to address here is the derivation of parameter update equations which is applicable to the family of ef-MCA data models (3.25)-(3.27). We have seen that the E-step procedure can be generally applied to any distribution of the exponential family. For the M-step, however, obtaining such a comprehensive approach for parameter updates seems to be infeasible. So far, generative models with Gaussian (Bornschein, Henniges, and Lücke, 2013) or Poisson (Lücke and Sahani, 2008) noise distribution have been studied (in the context of MCA models) and a set of concise update equations has been introduced accordingly. But they cannot be extended to the exponential family distributions. Besides, other LVMs that use exponential family noise models such as Bayesian EF-PCA (Mohamed, Ghahramani, and Heller, 2008) and EF-SC (Lee et al., 2009) do not provide a set of generalized update equations for the model parameters and only use Poisson and Bernoulli distributions in their numerical experiments. Notably, approaches such as EF-SC (Lee et al., 2009) employ a MAP-based approximation and do not learn all parameters of the model (for the Gaussian, for e.g., usually the variance parameter is not learned). In contrast, we will present a set of update equations that is suitable for any regular distribution of the exponential family with minimal representation property. We will specifically show that the same formulation can be obtained for the parameter updates of any regular exponential family distribution. The only difference that exists in using different noise distributions (that we will discuss it later) is the way we should compute the function $\vec{\Phi}(\vec{w})$ (3.9) which is, of course, pertinent to the choice of distribution.

4.2 Parameter Update Equations

Following the standard procedure, we will set the derivatives of $\mathcal{F}(q,\Theta)$ w.r.t. all model parameters to zero, and derive parameter update rules from the resulting equation systems. As mentioned before, we initially consider the two-parameter distributions, i.e., $\Theta = (\vec{\pi}, W, V)$ and discuss the general case later on. Among these parameters, W and V are the parameters of the noise distribution and $\vec{\pi}$ is the prior parameter. We will first investigate the updates for parameters W and V, and the corresponding update equation for the prior parameter $\vec{\pi}$ will be presented afterwards.

Doing so, one should observe that the derivatives of $\mathcal{F}(q,\Theta)$ contain derivatives of $\bar{W}_d(\vec{s},\Theta)$ and $\bar{V}_d(\vec{s},\Theta)$ w.r.t. the dictionary elements W_{dh} and V_{dh} for $d = 1,\ldots,D$ and

$h = 1, \ldots, H$. For these specific derivatives the following applies:

$$\frac{\partial}{\partial W_{dh}} \bar{W}_d(\vec{s}, \Theta) = \mathcal{A}_{dh}(\vec{s}, \Theta), \qquad \frac{\partial}{\partial W_{dh}} \bar{V}_d(\vec{s}, \Theta) = 0 \tag{4.12}$$

$$\frac{\partial}{\partial V_{dh}} \bar{V}_d(\vec{s}, \Theta) = \mathcal{A}_{dh}(\vec{s}, \Theta), \qquad \frac{\partial}{\partial V_{dh}} \bar{W}_d(\vec{s}, \Theta) = 0 \tag{4.13}$$

where

$$\mathcal{A}_{dh}(\vec{s}, \Theta) := \begin{cases} 1 & h = h(d, \vec{s}, \Theta) \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

which simply follows from considering the cases $h = h(d, \vec{s}, \Theta)$ and $h \neq h(d, \vec{s}, \Theta)$ separately. Function $\mathcal{A}_{dh}(\vec{s}, \Theta)$ has a useful property that we will exploit further below. For special cases of the ef-MCA data models, this property has been observed before (e.g., Lücke and Sahani, 2008; Lücke and Eggert, 2010 for Poisson and Gaussian noise models), but the following Lemma represents the required generalization for the exponential family observables:

Lemma 1. *Consider $\bar{W}_d(\vec{s}, \Theta)$ and $\bar{V}_d(\vec{s}, \Theta)$ that are defined in (3.38). Then, for any well-behaved function $g(\cdot)$ and any arbitrary $\vec{s} \in \{0, 1\}^H$, we have*

$$\mathcal{A}_{dh}(\vec{s}, \Theta) g(\bar{W}_d(\vec{s}, \Theta)) = \mathcal{A}_{dh}(\vec{s}, \Theta) g(W_{dh}) \tag{4.15}$$

and likewise

$$\mathcal{A}_{dh}(\vec{s}, \Theta) g(\bar{V}_d(\vec{s}, \Theta)) = \mathcal{A}_{dh}(\vec{s}, \Theta) g(V_{dh}). \tag{4.16}$$

Proof. We first prove the relation stated in (4.15). To this, note that for each pair (d_o, h_o) either of the following applies:

$$h_o = h(d_o, \vec{s}, \Theta) \quad \text{or} \quad h_o \neq h(d_o, \vec{s}, \Theta).$$

Now let $h_o = h(d_o, \vec{s}, \Theta)$, then it follows from (3.38) that

$$\mathcal{A}_{d_o h_o}(\vec{s}, \Theta) g(\bar{W}_{d_o}(\vec{s}, \Theta)) = \mathcal{A}_{d_o h_o}(\vec{s}, \Theta) g(W_{d_o h(d_o, \vec{s}, \Theta)}) = \mathcal{A}_{d_o h_o}(\vec{s}, \Theta) g(W_{d_o h_o}).$$

On the other hand, it follows from $h_o \neq h(d_o, \vec{s}, \Theta)$ and (4.14) that $\mathcal{A}_{d_o h_o}(\vec{s}, \Theta) = 0$ which trivially satisfies the claim. The proof of (4.16) is analog. □

Using the aforementioned lemma, we can now derive concise update equations for dictionaries W and V of the proposed ef-MCA data models which guarantee that all derivatives of $\mathcal{F}(q, \Theta)$ w.r.t. W_{dh} and V_{dh} vanish. These equations can then be used for parameter updates in an EM algorithm.

Theorem 2. *Consider an ef-MCA data model (3.25)-(3.27) with $p(y; \vec{\eta})$ being a regular exponential family distribution with $L = 2$. Furthermore, let the parameters Θ contain the matrices W and V with $D \times H$ entries and $\vec{\eta}_d(\vec{s}, \Theta)$ be defined as in Equations (3.28) and (3.37)-(3.38). Then, the derivatives of the free energy (4.6) w.r.t. all matrix elements W_{dh} and V_{dh} are zero if for all d and h applies:*

$$W_{dh} = \frac{\sum_{n=1}^{N} \langle \mathcal{A}_{dh}(\vec{s}, \Theta) \rangle_{q^{(n)}} T_1(y_d^{(n)})}{\sum_{n=1}^{N} \langle \mathcal{A}_{dh}(\vec{s}, \Theta) \rangle_{q^{(n)}}} \tag{4.17}$$

and

$$V_{dh} = \frac{\sum_{n=1}^{N} \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}} T_2(y_d^{(n)})}{\sum_{n=1}^{N} \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}} \tag{4.18}$$

where $\mathcal{A}_{dh}(\vec{s},\Theta)$ *is given by* (4.14).

Proof. Consider a single dictionary element W_{dh} and let us abbreviate $\bar{W}_d(\vec{s},\Theta)$ and $\bar{V}_d(\vec{s},\Theta)$ by $\bar{W}_d$ and $\bar{V}_d$, respectively. Then using the chain rule and Equations (3.28) and (4.12)-(4.13), we obtain:

$$\begin{aligned}
\frac{\partial}{\partial W_{dh}} \log(p(y_d;\vec{\eta}_d(\vec{s},\Theta))) &= \frac{\partial}{\partial W_{dh}} \log(p(y_d;\vec{\Phi}(\bar{W}_d,\bar{V}_d))) \qquad (4.19)\\
&= \sum_{l=1}^{2} \Big(\frac{\partial}{\partial W_{dh}} \Phi_l(\bar{W}_d,\bar{V}_d)\Big)\Big(\frac{\partial}{\partial \eta_l} \log(p(y_d;\vec{\eta}))\big|_{\vec{\eta}=\vec{\Phi}(\bar{W}_d,\bar{V}_d)}\Big)\\
&= \sum_{l=1}^{2} \Big(\frac{\partial}{\partial W_{dh}} \bar{W}_d\Big)\Big(\frac{\partial}{\partial w} \Phi_l(w,\bar{V}_d)\big|_{w=\bar{W}_d}\Big)\Big(\frac{\partial}{\partial \eta_l} \log(p(y_d;\vec{\eta}))\big|_{\vec{\eta}=\vec{\Phi}(\bar{W}_d,\bar{V}_d)}\Big)\\
&+ \sum_{l=1}^{2} \Big(\underbrace{\frac{\partial}{\partial W_{dh}} \bar{V}_d}_{=0}\Big)\Big(\frac{\partial}{\partial v} \Phi_l(\bar{W}_d,v)\big|_{v=\bar{V}_d}\Big)\Big(\frac{\partial}{\partial \eta_l} \log(p(y_d;\vec{\eta}))\big|_{\vec{\eta}=\vec{\Phi}(\bar{W}_d,\bar{V}_d)}\Big)\\
&= \sum_{l=1}^{2} \Big(\mathcal{A}_{dh}(\vec{s},\Theta)\Big)\Big(\frac{\partial}{\partial w} \Phi_l(w,\bar{V}_d)\big|_{w=\bar{W}_d}\Big)\Big(T_l(y_d^{(n)}) - \frac{\partial A(\vec{\eta})}{\partial \eta_l}\Big)\Big|_{\vec{\eta}=\vec{\Phi}(\bar{W}_d,\bar{V}_d)}.
\end{aligned}$$

Moreover, from Theorem 1 we know that for any regular distribution (i.e. with finite $A(\vec{\eta})$) of the exponential family, $A(\vec{\eta})$ satisfies:

$$\frac{\partial A(\vec{\eta})}{\partial \eta_1} = \langle T_1(y)\rangle_{p(y;\vec{\eta})} \quad \text{and} \quad \frac{\partial A(\vec{\eta})}{\partial \eta_2} = \langle T_2(y)\rangle_{p(y;\vec{\eta})}. \tag{4.20}$$

Thus, we can further simplify:

$$\begin{aligned}
\frac{\partial}{\partial W_{dh}} \log(p(y_d;\vec{\eta}_d(\vec{s},\Theta))) &= \sum_{l=1}^{2} \Big(\mathcal{A}_{dh}(\vec{s},\Theta)\Big)\Big(\frac{\partial}{\partial w} \Phi_l(w,\bar{V}_d)\big|_{w=\bar{W}_d}\Big) \qquad (4.21)\\
&\quad \times \Big(T_l(y_d^{(n)}) - \langle T_l(y)\rangle_{p(y;\vec{\eta})}\Big)\Big|_{\vec{\eta}=\vec{\Phi}(\bar{W}_d,\bar{V}_d)}\\
&= \sum_{l=1}^{2} \Big(\mathcal{A}_{dh}(\vec{s},\Theta)\Big)\Big(\frac{\partial}{\partial w} \Phi_l(w,\bar{V}_d)\big|_{w=\bar{W}_d}\Big)\Big(T_l(y_d^{(n)}) - \langle T_l(y)\rangle_{p(y;\vec{\Phi}(\bar{W}_d,\bar{V}_d))}\Big).
\end{aligned}$$

Now, using Lemma 1 we obtain:

$$\begin{aligned}
\frac{\partial}{\partial W_{dh}} \log(p(y_d;\vec{\eta}_d(\vec{s},\Theta))) &= \sum_{l=1}^{2} \Big(\mathcal{A}_{dh}(\vec{s},\Theta)\Big)\Big(\frac{\partial}{\partial w} \Phi_l(w,V_{dh})\big|_{w=W_{dh}}\Big) \qquad (4.22)\\
&\quad \times \Big(T_l(y_d^{(n)}) - \langle T_l(y)\rangle_{p(y;\vec{\Phi}(W_{dh},V_{dh}))}\Big).
\end{aligned}$$

Note that the above equation depends on parameter $\vec{s}$ of the hidden states only through the function $\mathcal{A}_{dh}(\vec{s},\Theta)$. This is an important property of Lemma 1 that alleviates the complexity of the aforementioned equations and enables us to extract a set of concise update equations

for dictionaries W and V. To see this, we derive (using Equation (4.6)):

$$\begin{aligned}
\frac{\partial}{\partial W_{dh}}\mathcal{F}(q,\Theta) &= \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \frac{\partial}{\partial W_{dh}} \Big(\sum_{d'} \log\left(p(y_{d'}^{(n)}; \vec{\eta}_{d'}(\vec{s},\Theta))\right)\Big) \\
&= \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \Big(\sum_{d'} \frac{\partial}{\partial W_{dh}} \log(p(y_{d'}^{(n)}; \vec{\eta}_{d'}(\vec{s},\Theta)))\Big) \\
&= \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \Big(\sum_{d'} \sum_{l=1}^{2} \Big(\mathcal{A}_{d'h}(\vec{s},\Theta)\Big) \Big(\frac{\partial}{\partial w}\Phi_l(w, V_{d'h})\big|_{w=W_{d'h}}\Big) \\
&\quad \times \Big(T_l(y_{d'}^{(n)}) - \langle T_l(y)\rangle_{p(y;\vec{\Phi}(W_{d'h},V_{d'h}))}\Big)\delta_{dd'}\Big) \\
&= \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \sum_{l=1}^{2} \Big(\mathcal{A}_{dh}(\vec{s},\Theta)\Big) \Big(\frac{\partial}{\partial w}\Phi_l(w, V_{dh})\big|_{w=W_{dh}}\Big) \\
&\quad \times \Big(T_l(y_{d}^{(n)}) - \langle T_l(y)\rangle_{p(y;\vec{\Phi}(W_{dh},V_{dh}))}\Big) \\
&= \sum_{l=1}^{2} \Big(\frac{\partial}{\partial w}\Phi_l(w, V_{dh})\big|_{w=W_{dh}}\Big) \sum_n \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}} \\
&\quad \times \Big(T_l(y_{d}^{(n)}) - \langle T_l(y)\rangle_{p(y;\vec{\Phi}(W_{dh},V_{dh}))}\Big) \\
&= \Big(\frac{\partial}{\partial w}\Phi_1(w, V_{dh})\big|_{w=W_{dh}}\Big) \sum_n \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}} \Big(T_1(y_{d}^{(n)}) - W_{dh}\Big) \\
&\quad + \Big(\frac{\partial}{\partial w}\Phi_2(w, V_{dh})\big|_{w=W_{dh}}\Big) \sum_n \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}} \Big(T_2(y_{d}^{(n)}) - V_{dh}\Big)
\end{aligned} \tag{4.23}$$

where in the last equation we exploited our mean value parameterization defined by $\vec{w} := \langle \vec{T}(y)\rangle_{p(y;\vec{\Phi}(\vec{w}))}$. Now, independently of the functions $\frac{\partial}{\partial w}\Phi_l(w, V_{dh})\big|_{w=W_{dh}}$ for $l = 1, 2$, derivative of the free energy (the ELBO) w.r.t. W_{dh} is zero, i.e. $\frac{\partial \mathcal{F}}{\partial W_{dh}} = 0$, if applies:

$$\sum_n \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}} \left(T_1(y_{d}^{(n)}) - W_{dh}\right) = 0 \tag{4.24}$$

and

$$\sum_n \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}} \left(T_2(y_{d}^{(n)}) - V_{dh}\right) = 0\,. \tag{4.25}$$

Rearranging terms yields Equations (4.17) and (4.18) which completes the proof. The proof proceeds along the same lines for $\frac{\partial \mathcal{F}}{\partial V_{dh}}$ which results in the same set of equations. □

Fulfilling Equations (4.17) and (4.18) guarantees vanishing derivatives and provides a generally applicable approach for updating W and V in each M-step. We do remark, however, that we have not strictly proven that (4.17) and (4.18) correspond to a maxima (and not minima or saddle points). In fact, one can compute the second derivatives and perform the required tests to examine the considered critical points closely. But this is far from our goal in this study and we hereafter follow previous work in order to update the dictionaries W and V using these equations. In practice, nonetheless, we observed that the updates obtained by the above equations do increase the free energy function to a (possibly local) maxima. Furthermore, we here emphasize that Equations (4.17) and (4.18) are valid for any regular two-parameter distribution of the exponential family which includes Gaussian, Gamma, Beta and many more (this results in a variety of noise models that we can consider). Importantly, the above theorem reveals that the same mathematical formulation is obtained for the parameters of all these

distributions under the generative model (3.25)-(3.27) (note that we here considered the $L = 2$ case but the general case will be discussed further below). Hence, the same update rules can be applied for the parameter optimization of different ef-MCA data models. Such a general case, to our best knowledge, has not been reported before for the LVMs and the current study is the first to present a set of generally applicable updates based on the results of Theorem 2.

In addition, observe that a straightforward outcome of the foregoing theorem is when the distribution does contain a sufficient statistic proportional to y, i.e. $T_1(y) = y$. This specific form yields a further simplification of our algorithm which is presented in the following:

Corollary 1. *Prerequisites as for Theorem 2, if the distribution $p(y;\vec{\eta})$ has sufficient statistic $T_1(y) = y$, then the condition for W_{dh} is given by:*

$$W_{dh} = \frac{\sum_{n=1}^{N} \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}} y_d^{(n)}}{\sum_{n=1}^{N} \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}}. \tag{4.26}$$

The corollary finally explains why the update equation for the original MCA data model (Lücke and Sahani, 2008) (which used Poisson noise) and the update equation for later MCA models (Lücke and Eggert, 2010; Bornschein, Henniges, and Lücke, 2013; Sheikh et al., 2019) (which used Gaussian noise) are identical and given by (4.26). More importantly, this update satisfies many other distributions of the exponential family such as Gamma, Exponential, Bernoulli etc.

Finally observe that Equations (4.17) and (4.18) do not represent closed-form solutions for W and V because their right-hand-sides also depend on W and V through the function $\mathcal{A}_{dh}(\vec{s},\Theta)$. Thereby, following previous work (Lücke and Sahani, 2008; Lücke and Eggert, 2010), we use (4.17) and (4.18) in the fixed-point sense; i.e., we update:

$$W_{dh}^{\text{new}} = \frac{\sum_{n=1}^{N} \langle \mathcal{A}_{dh}(\vec{s},\Theta^{\text{old}})\rangle_{q^{(n)}} T_1(y_d^{(n)})}{\sum_{n=1}^{N} \langle \mathcal{A}_{dh}(\vec{s},\Theta^{\text{old}})\rangle_{q^{(n)}}} \tag{4.27}$$

and

$$V_{dh}^{\text{new}} = \frac{\sum_{n=1}^{N} \langle \mathcal{A}_{dh}(\vec{s},\Theta^{\text{old}})\rangle_{q^{(n)}} T_2(y_d^{(n)})}{\sum_{n=1}^{N} \langle \mathcal{A}_{dh}(\vec{s},\Theta^{\text{old}})\rangle_{q^{(n)}}} \tag{4.28}$$

where also $q^{(n)} = q^{(n)}(\vec{s};\Theta^{\text{old}})$ depends on the old parameters. If repeated updates result in the values of W and V to converge, then the converged values fulfill Equations (4.17) and (4.18), respectively.

To complete the M-step parameter updates, we now derive the update equation for the prior parameter $\vec{\pi}$. This derivation, however, does not involve the specific form of the observables' distribution and can be applied in general for any ef-MCA data model. Following theorem then presents the corresponding update of the parameter $\vec{\pi}$:

Theorem 3. *Consider an ef-MCA data model (3.25)-(3.27) with $p(y;\vec{\eta})$ being a regular exponential family distribution. Then, the derivatives of the free energy function (4.6) w.r.t. π_h are zero if for all h applies:*

$$\pi_h = \frac{1}{N}\sum_{n=1}^{N} \langle s_h\rangle_{q^{(n)}}. \tag{4.29}$$

Proof. Consider a single prior element π_h. We assume $\pi_h \in (0,1)$ so that values $\log(\pi_h)$ and $\log(1-\pi_h)$ are well-defined. Then, for the free energy function stated in (4.6), we can

write:

$$\begin{aligned}
\frac{\partial}{\partial \pi_h}\mathcal{F}(q,\Theta) &= \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \frac{\partial}{\partial \pi_h}\Big(\sum_{h'} \log(p(s_{h'} \mid \Theta))\Big) \\
&= \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \Big(\frac{\partial}{\partial \pi_h}\sum_{h'} s_{h'}\log(\pi_{h'}) + (1-s_{h'})\log(1-\pi_{h'})\Big) \\
&= \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \sum_{h'} \Big(\frac{s_{h'}}{\pi_{h'}} - \frac{1-s_{h'}}{1-\pi_{h'}}\Big)\delta_{hh'} \\
&= \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \Big(\frac{s_h}{\pi_h} + \frac{s_h}{1-\pi_h} - \frac{1}{1-\pi_h}\Big) \\
&= \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) s_h \Big(\frac{1}{\pi_h} + \frac{1}{1-\pi_h}\Big) - \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \frac{1}{1-\pi_h} \\
&= \Big(\frac{1}{\pi_h(1-\pi_h)} \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) s_h\Big) - \Big(\frac{1}{1-\pi_h} \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s})\Big).
\end{aligned} \tag{4.30}$$

Since $q^{(n)}(\vec{s})$ is a valid probability distribution, we can insert $\sum_{\vec{s}} q^{(n)}(\vec{s}) = 1$ in the last equation above. Furthermore, by setting $\frac{\partial \mathcal{F}}{\partial \pi_h} = 0$ we can derive:

$$\frac{1}{\pi_h(1-\pi_h)} \sum_n \langle s_h \rangle_{q^{(n)}} = \frac{N}{1-\pi_h}. \tag{4.31}$$

Now, multiplying the above equation by $\pi_h(1-\pi_h)$, we obtain (note that $\pi_h(1-\pi_h) \neq 0$ since $\pi_h \in (0,1)$ for each h):

$$\sum_n \langle s_h \rangle_{q^{(n)}} = N\pi_h \tag{4.32}$$

which satisfies our claim (4.29). □

Therefore, we can now update prior parameter $\vec{\pi}$ in each M-step as follows:

$$\pi_h^{\text{new}} = \frac{1}{N}\sum_{n=1}^{N} \langle s_h \rangle_{q^{(n)}}, \quad h = 1, \ldots, H. \tag{4.33}$$

Equations (4.27)-(4.28) and (4.33) present a complete set of update equations for parameters $\Theta = (\vec{\pi}, W, V)$ of the proposed generative models. If the considered probability distribution has only one parameter (i.e. $L = 1$), then $\Theta = (\vec{\pi}, W)$ and we only require to use (4.27) and (4.33) in the M-step. For other cases where $L > 2$, the same procedure can be applied to obtain the update rules. We will later investigate such a general case further below. But, before that, let us summarize the exact EM algorithm presented for training our family of generative models.

4.2.1 The EM Algorithm for training the proposed ef-MCA data models

The concrete EM algorithm for training the proposed generative models is given by Algorithm 1 for a two-parameter distribution ($L = 2$) of the exponential family. Note that for one-parameter distributions ($L = 1$), dependencies are just on $\bar{W}(\vec{s}, \Theta)$.

Algorithm 1: The exact EM algorithm for training ef-MCA data models ($L = 2$)

initialize model parameters $\Theta = (\vec{\pi}, W, V)$;
repeat
  compute the inverse mapping $\vec{\Phi}$ defined in (3.9);
  if *the distribution* $p(y;\vec{\eta})$ *has sufficient statistic* $T_1(y) = y$ **then**
    $M(\Theta) = W$;
  else
    compute the function F in (3.37) based on the relation of the natural parameters $\vec{\eta}$ and the first moment of $p(y;\vec{\eta})$;
    $M(\Theta) = F(W, V)$;
  end
  for *each vector* $\vec{s}$ *of the latent space* **do**
    for $d = 1 : D$ **do**
      $h(d, \vec{s}, \Theta) = \text{argmax}_h\{M_{dh}(\Theta)s_h\}$;
      $\bar{W}_d = W_{dh(d,\vec{s},\Theta)}$ and $\bar{V}_d = V_{dh(d,\vec{s},\Theta)}$;
      $\vec{\bar{\eta}}_d = \vec{\Phi}(\bar{W}_d, \bar{V}_d)$;
    end
    for $n = 1 : N$ **do**
      $q^{(n)}(\vec{s}) = p(\vec{s} \,|\, \vec{y}^{(n)}, \Theta)$;
      for $h = 1 : H$ *and* $d = 1 : D$ **do**
        compute $q^{(n)}(\vec{s})s_h$;
        compute $q^{(n)}(\vec{s})\mathcal{A}_{dh}(\vec{s}, \Theta)$ where $\mathcal{A}_{dh}(\vec{s}, \Theta)$ is defined in (4.14);
      end
    end
  end
  update parameters Θ using (4.27)-(4.28) and (4.33);
until *parameters* Θ *have sufficiently converged*;

4.3 Parameter Update Equations – General Case

We derived Theorem 2 for the case of $L = 2$, i.e., for distributions of the exponential family with sufficient statistics $\vec{T}(y)$ of length two. This choice was for notational convenience only. Considering the proof of Theorem 2, it can be easily inferred that it applies for any $L \in \mathbb{N}$. We state this more formally in this section.

To this, first recall the details presented in Section 3.5. Then, similar to the $L = 2$ case, observe that the derivatives of $\mathcal{F}(q, \Theta)$ contain derivatives of $\bar{W}_d^{(l)}(\vec{s}, \Theta)$ for $l = 1, \ldots, L$ w.r.t. the dictionary elements $W_{dh}^{(l)}$ for $d = 1, \ldots, D$ and $h = 1, \ldots, H$. For these derivatives we have:

$$\frac{\partial}{\partial W_{dh}^{(l)}} \bar{W}_d^{(l')}(\vec{s}, \Theta) = \begin{cases} \mathcal{A}_{dh}(\vec{s}, \Theta) & \text{if } l = l' \\ 0 & \text{otherwise} \end{cases} \tag{4.34}$$

where $\mathcal{A}_{dh}(\vec{s}, \Theta)$ is given by (4.14). Similar to the $L = 2$ case, the proof of the above equation is trivial and can be easily obtained by considering the cases $h = h(d, \vec{s}, \Theta)$ and $h \neq h(d, \vec{s}, \Theta)$ separately. In short, we can write:

$$\frac{\partial}{\partial W_{dh}^{(l)}} \bar{W}_d^{(l')}(\vec{s}, \Theta) = \mathcal{A}_{dh}(\vec{s}, \Theta)\delta_{ll'} \tag{4.35}$$

where $\delta_{ll'}$ denotes the delta Kronecker. Then, for the Equations (3.25)-(3.27) with (3.42)-(3.44) that generally define the family of ef-MCA data models, the following general theorem applies:

Theorem 4. *Consider an ef-MCA data model (3.25)-(3.27) with $p(y;\vec{\eta})$ being a regular exponential family distribution with sufficient statistics $\vec{T}(y)$ of length $L \in \mathbb{N}$. Moreover, let the parameters Θ contain L matrices $W^{(1)},\ldots,W^{(L)}$ with $D \times H$ entries and let $\vec{\bar{\eta}}_d(\vec{s},\Theta)$ be defined as in (3.42)-(3.44). Then, the derivatives of the free energy (4.6) w.r.t. all $W^{(l)}_{dh}$ are zero if for all d, h, and l applies:*

$$W^{(l)}_{dh} = \frac{\sum_{n=1}^{N}\langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}\, T_l(y^{(n)}_d)}{\sum_{n=1}^{N}\langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}} \tag{4.36}$$

where $\mathcal{A}_{dh}(\vec{s},\Theta)$ is given by (4.14).

Proof. Consider a single parameter $W^{(l)}_{dh}$ for an arbitrary $1 \leq l \leq L$ and, for the sake of readability, let us abbreviate $\bar{W}^{(l)}_d(\vec{s},\Theta)$ by $\bar{W}^{(l)}_d$. Then using the chain rule and Equations (3.28) and (4.35), we get:

$$\begin{aligned}
\frac{\partial}{\partial W^{(l)}_{dh}}\log\left(p(y_d;\vec{\bar{\eta}}_d(\vec{s},\Theta))\right) &= \frac{\partial}{\partial W^{(l)}_{dh}}\log\left(p(y_d;\vec{\Phi}(\bar{W}^{(1)}_d,\ldots,\bar{W}^{(L)}_d))\right) \qquad (4.37)\\
&= \sum_{l'=1}^{L}\left(\frac{\partial}{\partial W^{(l)}_{dh}}\Phi_{l'}(\bar{W}^{(1)}_d,\ldots,\bar{W}^{(L)}_d)\right)\left(\frac{\partial}{\partial \eta_{l'}}\log(p(y_d;\vec{\eta}))\big|_{\vec{\eta}=\vec{\Phi}(\bar{W}^{(1)}_d,\ldots,\bar{W}^{(L)}_d)}\right)\\
&= \sum_{l'=1}^{L}\left\{\sum_{l''=1}^{L}\left(\frac{\partial}{\partial W^{(l)}_{dh}}\bar{W}^{(l'')}_d\right)\left(\frac{\partial}{\partial w}\Phi_{l'}(\bar{W}^{(1)}_d,\ldots,w,\ldots,\bar{W}^{(L)}_d)\big|_{w=\bar{W}^{(l'')}_d}\right)\right\}\\
&\quad\times\left(\frac{\partial}{\partial \eta_{l'}}\log(p(y_d;\vec{\eta}))\big|_{\vec{\eta}=\vec{\Phi}(\bar{W}^{(1)}_d,\ldots,\bar{W}^{(L)}_d)}\right)\\
&= \sum_{l'=1}^{L}\left\{\sum_{l''=1}^{L}\left(\mathcal{A}_{dh}(\vec{s},\Theta)\delta_{ll''}\right)\left(\frac{\partial}{\partial w}\Phi_{l'}(\bar{W}^{(1)}_d,\ldots,w,\ldots,\bar{W}^{(L)}_d)\big|_{w=\bar{W}^{(l'')}_d}\right)\right\}\\
&\quad\times\left(\frac{\partial}{\partial \eta_{l'}}\log(p(y_d;\vec{\eta}))\big|_{\vec{\eta}=\vec{\Phi}(\bar{W}^{(1)}_d,\ldots,\bar{W}^{(L)}_d)}\right)\\
&= \sum_{l'=1}^{L}\left(\mathcal{A}_{dh}(\vec{s},\Theta)\right)\left(\frac{\partial}{\partial w}\Phi_{l'}(\bar{W}^{(1)}_d,\ldots,w,\ldots,\bar{W}^{(L)}_d)\big|_{w=\bar{W}^{(l)}_d}\right)\\
&\quad\times\left(T_{l'}(y^{(n)}_d)-\frac{\partial A(\vec{\eta})}{\partial \eta_{l'}}\right)\Big|_{\vec{\eta}=\vec{\Phi}(\bar{W}^{(1)}_d,\ldots,\bar{W}^{(L)}_d)}.
\end{aligned}$$

Moreover, from Theorem 1 we know that for any regular distribution (i.e. with finite $A(\vec{\eta})$) of the exponential family, $A(\vec{\eta})$ satisfies:

$$\frac{\partial A(\vec{\eta})}{\partial \eta_{l'}} = \langle T_{l'}(y)\rangle_{p(y;\vec{\eta})}. \tag{4.38}$$

Therefore, we can write:

$$\begin{aligned}\frac{\partial}{\partial W_{dh}^{(l)}}\log(p(y_d;\vec{\eta}_d(\vec{s},\Theta))) &= \sum_{l'=1}^{L}\Big(\mathcal{A}_{dh}(\vec{s},\Theta)\Big)\Big(\frac{\partial}{\partial w}\Phi_{l'}(\bar{W}_d^{(1)},\ldots,w,\ldots,\bar{W}_d^{(L)})\big|_{w=\bar{W}_d^{(l)}}\Big)\\ &\quad\times\Big(T_{l'}(y_d^{(n)}) - \langle T_{l'}(y)\rangle_{p(y;\vec{\eta})}\Big)\Big|_{\vec{\eta}=\vec{\Phi}(\bar{W}_d^{(1)},\ldots,\bar{W}_d^{(L)})}\\ &= \sum_{l'=1}^{L}\Big(\mathcal{A}_{dh}(\vec{s},\Theta)\Big)\Big(\frac{\partial}{\partial w}\Phi_{l'}(\bar{W}_d^{(1)},\ldots,w,\ldots,\bar{W}_d^{(L)})\big|_{w=\bar{W}_d^{(l)}}\Big)\\ &\quad\times\Big(T_{l'}(y_d^{(n)}) - \langle T_{l'}(y)\rangle_{p(y;\vec{\Phi}(\bar{W}_d^{(1)},\ldots,\bar{W}_d^{(L)}))}\Big).\end{aligned} \tag{4.39}$$

Now, using Lemma 1 we obtain:

$$\begin{aligned}\frac{\partial}{\partial W_{dh}^{(l)}}\log(p(y_d;\vec{\eta}_d(\vec{s},\Theta))) &= \sum_{l'=1}^{L}\Big(\mathcal{A}_{dh}(\vec{s},\Theta)\Big)\Big(\frac{\partial}{\partial w}\Phi_{l'}(W_{dh}^{(1)},\ldots,w,\ldots,W_{dh}^{(L)})\big|_{w=W_{dh}^{(l)}}\Big)\\ &\quad\times\Big(T_{l'}(y_d^{(n)}) - \langle T_{l'}(y)\rangle_{p(y;\vec{\Phi}(W_{dh}^{(1)},\ldots,W_{dh}^{(L)}))}\Big).\end{aligned} \tag{4.40}$$

Note that the above equation depends on parameter $\vec{s}$ of the hidden states only through function $\mathcal{A}_{dh}(\vec{s},\Theta)$. Using Equation (4.6), we then have:

$$\begin{aligned}\frac{\partial}{\partial W_{dh}^{(l)}}\mathcal{F}(q,\Theta) &= \sum_n\sum_{\vec{s}} q^{(n)}(\vec{s})\frac{\partial}{\partial W_{dh}^{(l)}}\Big(\sum_{d'}\log\big(p(y_{d'}^{(n)};\vec{\eta}_{d'}(\vec{s},\Theta))\big)\Big) \qquad (4.41)\\ &= \sum_n\sum_{\vec{s}} q^{(n)}(\vec{s})\Big(\sum_{d'}\frac{\partial}{\partial W_{dh}^{(l)}}\log\big(p(y_{d'}^{(n)};\vec{\eta}_{d'}(\vec{s},\Theta))\big)\Big)\\ &= \sum_n\sum_{\vec{s}} q^{(n)}(\vec{s})\Big(\sum_{d'}\sum_{l'=1}^{L}\Big(\mathcal{A}_{d'h}(\vec{s},\Theta)\Big)\Big(\frac{\partial}{\partial w}\Phi_{l'}(W_{d'h}^{(1)},\ldots,w,\ldots,W_{d'h}^{(L)})\big|_{w=W_{d'h}^{(l)}}\Big)\\ &\quad\times\Big(T_{l'}(y_{d'}^{(n)}) - \langle T_{l'}(y)\rangle_{p(y;\vec{\Phi}(W_{d'h}^{(1)},\ldots,W_{d'h}^{(L)}))}\Big)\delta_{dd'}\Big)\\ &= \sum_n\sum_{\vec{s}} q^{(n)}(\vec{s})\sum_{l'=1}^{L}\Big(\mathcal{A}_{dh}(\vec{s},\Theta)\Big)\Big(\frac{\partial}{\partial w}\Phi_{l'}(W_{dh}^{(1)},\ldots,w,\ldots,W_{dh}^{(L)})\big|_{w=W_{dh}^{(l)}}\Big)\\ &\quad\times\Big(T_{l'}(y_d^{(n)}) - \langle T_{l'}(y)\rangle_{p(y;\vec{\Phi}(W_{dh}^{(1)},\ldots,W_{dh}^{(L)}))}\Big)\\ &= \sum_{l'=1}^{L}\Big(\frac{\partial}{\partial w}\Phi_{l'}(W_{dh}^{(1)},\ldots,w,\ldots,W_{dh}^{(L)})\big|_{w=W_{dh}^{(l)}}\Big)\sum_n\langle\mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}\\ &\quad\times\Big(T_{l'}(y_d^{(n)}) - W_{dh}^{(l')}\Big)\end{aligned}$$

where in the last equation we exploited the mean value parameterization defined in (3.7), that is:

$$\vec{w} := \langle\vec{T}(y)\rangle_{p(y;\vec{\Phi}(\vec{w}))}. \tag{4.42}$$

Therefore, independently of the choice of functions $\frac{\partial}{\partial w}\Phi_{l'}(W_{dh}^{(1)},\dots,w,\dots,W_{dh}^{(L)})\big|_{w=W_{dh}^{(l)}}$ for each l', we have $\frac{\partial \mathcal{F}}{\partial W_{dh}^{(l)}} = 0$ if, for $l = 1,\dots,L$, applies:

$$\sum_n \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}} \left(T_l(y_d^{(n)}) - W_{dh}^{(l)}\right) = 0 \tag{4.43}$$

that yields Equations (4.36) and completes the proof. □

The general case does, of course, include the case $L = 1$ and consequently Bernoulli, Exponential or the Poisson distribution. Similar to the two-parameter case, the updates (4.36) do not provide closed-form solutions for the dictionaries $W_{dh}^{(l)}$; and we therefore employ the same approach as before and for each d, h and l let:

$$(W_{dh}^{(l)})^{\text{new}} = \frac{\sum_{n=1}^N \langle \mathcal{A}_{dh}(\vec{s},\Theta^{\text{old}})\rangle_{q^{(n)}} T_l(y_d^{(n)})}{\sum_{n=1}^N \langle \mathcal{A}_{dh}(\vec{s},\Theta^{\text{old}})\rangle_{q^{(n)}}}. \tag{4.44}$$

For the proposed EM algorithm, we initialize dictionaries $W_{dh}^{(l)}$ for $l = 1,\dots,L$ and update their elements in each M-step. After the convergence, the converged dictionaries will satisfy (4.36) and therefore represent the optimal parameters of the model. Moreover, note that the prior parameter update would be exactly the same and given by (4.33) as it is independent of the choice of noise distribution.

4.4 Truncated Variational Expectation Maximization

So far, we have presented the details of the EM algorithm that can be used for training the proposed family of generative models (3.25)-(3.27). The main focus of the previous section, however, was the derivation of the parameter update equations (the M-step). We specifically showed how the use of a maximum function defined in (3.38) as a non-linear superposition results in concise and elegant update equations for the model parameters. Moreover, in each E-step, we proved that variational distributions $q^{(n)}(\vec{s})$ should be equated, for each $n = 1,\dots,N$, to the exact posteriors; i.e., in each E-step we let:

$$q^{(n)}(\vec{s}) = p(\vec{s}\,|\,\vec{y}^{(n)},\Theta) = \frac{p(\vec{s},\vec{y}^{(n)}\,|\Theta)}{\sum_{\vec{s}'} p(\vec{s}',\vec{y}^{(n)}\,|\Theta)} \tag{4.45}$$

and then compute the following expected values for a well-behaved function $g(\vec{s})$ (in the updates (4.27)-(4.28) and (4.33), $g(\vec{s})$ should be replaced by either $\mathcal{A}_{dh}(\vec{s},\Theta)$ or s_h):

$$\langle g(\vec{s})\rangle_{q^{(n)}} = \sum_{\vec{s}} q^{(n)}(\vec{s}) g(\vec{s}) = \frac{\sum_{\vec{s}} p(\vec{s},\vec{y}^{(n)}\,|\Theta) g(\vec{s})}{\sum_{\vec{s}'} p(\vec{s}',\vec{y}^{(n)}\,|\Theta)}. \tag{4.46}$$

The expected values above can be computed for small-scale ef-MCA models and the corresponding results will be used in Equations (4.27)-(4.28) and (4.33) accordingly. For large-scale applications, however, we require approximations in our optimization procedure as computing (4.46) becomes intractable. This is simply because of the fact that the computations grow exponentially by the number of H (all different combinatorics of the bit vectors yields amounts of 2^H distinct hidden states $\vec{s}$ that we have to sum over for each posterior). Therefore, approximations of the variational distributions $q^{(n)}(\vec{s})$ (or equivalently the posteriors) are

required: Approximations of the EM meta-algorithm. Such approximations include different areas of research such as sampling-based approximations (Booth and Hobert, 1999; Wei and Tanner, 1990; Lindsten, 2013), maximum a-posterior or *hard* EM approaches (Tibshirani, 1996; Olshausen and Field, 1996a; Celeux and Govaert, 1992; Mairal et al., 2010), Laplace approximations (Kass and Steffey, 1989; Friston et al., 2007) and *variational* EM approaches (Jordan et al., 1999; Neal and Hinton, 1998; Opper and Winther, 2005; Seeger, 2008; Kingma and Welling, 2013). In our study here, we are specifically concerned with the latter case and seek a class of variational approximations that are based on truncations of the full posteriors (Lücke and Sahani, 2008; Lücke and Eggert, 2010). Consequently, we refer to the case where the exact posteriors (4.45) are used in the E-step as the *exact* EM or *full* EM, and refer to the case where approximations are used as the variational EM.

Presumably, the two prominent instances of the variational EM approaches are based on Gaussian variational distributions (Opper and Winther, 2005; Seeger, 2008; Opper and Archambeau, 2009; Kingma and Welling, 2013) and factored variational distributions (Jordan et al., 1999). The former method uses mono-modal Gaussian distributions to approximate the posteriors, and the latter, which is also known as the *mean-field* approximation, uses factored variational distributions. Truncated approximations can be seen as another types of variational EM approaches which attempt to approximate the variational distribution $q^{(n)}(\vec{s})$, corresponding to a data point $\vec{y}^{(n)}$, as a proportion of the full posterior $p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)$ (see, e.g., Lücke and Sahani, 2008; Lücke and Eggert, 2010; Sheikh, Shelton, and Lücke, 2014; Dai and Lücke, 2014; Shelton et al., 2017; Guiraud, Drefs, and Lücke, 2018; Sheikh et al., 2019; Lücke, 2019 for a variety of approaches developed in this direction).

In contrast to the Gaussian or mean-field approaches that use a parametric function to approximate the variational distributions, the truncated approaches attempt to find a subset of hidden states by a preselection mechanism (this could be based on a sampling method like (Lücke, Dai, and Exarchakis, 2017), or a selection function such as (Lücke and Eggert, 2010; Henniges et al., 2010), or most recently an evolutionary algorithm as presented in (Guiraud, Drefs, and Lücke, 2018; Drefs, Guiraud, and Lücke, 2020)) in order to reduce the amount of summations in (4.46). These methods are importantly tailored to generative models with discrete latents and can therefore be exploited directly in our study here (we use binary latents drawn from independent Bernoulli distributions).

The most recent approach in this direction is *Truncated Variational Expectation Maximization* (TV-EM) introduced by Lücke in (Lücke, 2019) which, for $n = 1,\ldots,N$, defines:

$$q^{(n)}(\vec{s}\,|\,\mathcal{K},\Theta) := \frac{p(\vec{s},\vec{y}^{(n)}\,|\Theta)}{\sum\limits_{\vec{s}'\in\mathcal{K}^{(n)}} p(\vec{s}',\vec{y}^{(n)}\,|\Theta)}\,\delta(\vec{s}\in\mathcal{K}^{(n)}) \tag{4.47}$$

where $\delta(\vec{s}\in\mathcal{K}^{(n)})$ is 1 if $\vec{s}$ is in the subset $\mathcal{K}^{(n)}$ and zero otherwise. The set $\mathcal{K}^{(n)}$ contains a finite number of hidden states $\vec{s}$ corresponding to each data point $\vec{y}^{(n)}$, and the set of all $\mathcal{K}^{(n)}$ is denoted by $\mathcal{K}$; i.e. $\mathcal{K} = (\mathcal{K}^{(1)},\ldots,\mathcal{K}^{(N)})$. The subset $\mathcal{K}^{(n)}$ then contains the states that best describe the generation of $\vec{y}^{(n)}$ given the parameters Θ (see, e.g., Lücke, 2019; Guiraud, Drefs, and Lücke, 2018 for details).

Equations (4.47) approximate full posteriors by truncating the sum over the whole latent space to sum over those subsets $\mathcal{K}^{(n)}$ which accumulate most of the posterior mass. This has been shown to produce an accurate approximation in the case that subsets $\mathcal{K}^{(n)}$ are well chosen (i.e., if the contribution of hidden states that are not included in $\mathcal{K}^{(n)}$ is indeed negligible compared to the contribution of others). In detail, we require a selection of hidden states for each n to apply *variational E-steps* and further approximate the posteriors in order to obtain a tractable learning algorithm. Concretely, we compute the following expected values in each E-step:

$$\langle g(\vec{s})\rangle_{q^{(n)}} = \sum_{\vec{s}} q^{(n)}(\vec{s} \mid \mathcal{K}, \Theta) g(\vec{s}) = \frac{\sum_{\vec{s}} p(\vec{s}, \vec{y}^{(n)} \mid \Theta) g(\vec{s})}{\sum\limits_{\vec{s}' \in \mathcal{K}^{(n)}} p(\vec{s}', \vec{y}^{(n)} \mid \Theta)} \delta(\vec{s} \in \mathcal{K}^{(n)}) \tag{4.48}$$
$$= \frac{\sum\limits_{\vec{s} \in \mathcal{K}^{(n)}} p(\vec{s}, \vec{y}^{(n)} \mid \Theta) g(\vec{s})}{\sum\limits_{\vec{s}' \in \mathcal{K}^{(n)}} p(\vec{s}', \vec{y}^{(n)} \mid \Theta)}.$$

In each iteration of the corresponding variational EM, we now require to find a set $\mathcal{K}$ such that the hidden states preserved in each $\mathcal{K}^{(n)}$ can provide a good approximation of the full posteriors $p(\vec{s} \mid \vec{y}^{(n)}, \Theta)$ in (4.47). This raises the question of how such hidden states should be chosen (based on which objective), and how many of them should be considered to approximate the full posteriors. Trivially, if we include all hidden states to a subset $\mathcal{K}^{(n)}$, then the full posteriors can be obtained accordingly. In addition, if we choose a small number of hidden states for subsets $\mathcal{K}^{(n)}$, then fewer amounts of computations are required in each variational EM, but at the same time, we might end up with worse approximations. Hence, it is important to find a trade-off between the number of states stored in each subset $\mathcal{K}^{(n)}$ and the amount of computations that we can afford. This is in general a parameter that can be tuned empirically for each specific experiment (we have tried out different values to find a suitable setting; for e.g., we used 60 variational states for our experiments in Section 5.2). Indeed, it has been shown, e.g. by (Guiraud, Drefs, and Lücke, 2018), that even a naive choice of subsets $\mathcal{K}^{(n)}$ can in practice yield a concrete approach to perform tractable EM approximations. However, the aim is to find the subsets $\mathcal{K}^{(n)}$ in each variational E-step that maximizes the free energy (in this case the *variational free energy* or *variational lower bound*) (Lücke, 2019). In this direction, it has been proved for TV-EM that the variational lower bound can be written as the following compact form:

$$\mathcal{F}(\mathcal{K}, \Theta) = \sum_{n} \log\Big(\sum_{\vec{s} \in \mathcal{K}^{(n)}} p(\vec{s}, \vec{y}^{(n)} \mid \Theta) \Big) \tag{4.49}$$

which will be maximized w.r.t. the parameters $\mathcal{K}$ in each variational EM.
To do so and to update the subsets $\mathcal{K}^{(n)}$ for our experiments here, we use an approach that uses evolutionary algorithms with fitness defined to be a monotonic function of the model joint $p(\vec{s}, \vec{y} \,|\Theta)$. The method, known as *Evolutionary Expectation Maximization* (EEM) introduced in (Guiraud, Drefs, and Lücke, 2018), does not require new derivations for any new generative model, and thus can be directly applied as a 'black-box' to any regular distribution of the exponential family. We therefore exploit EEM in each variational EM of our proposed approach in order to obtain a tractable algorithm. We will later elaborate in Chapter 5 if we use an exact EM (computing full posteriors in each iteration) or if we apply a variational EM (exploiting approximations using EEM) for each of the experiments.

4.5 Relation to Other Approaches for Parameter Estimation

In the previous sections, we presented an EM algorithm for *Maximum Likelihood Estimation* (MLE) (Bishop, 2006) as a common approach for training probabilistic generative models. In detail, we showed how to obtain optimal parameters Θ^* that maximize the log-likelihood of the data or accurately, the free energy function as a lower bound of the log-likelihood. Now, in this section, we want to closely investigate the updates (4.44) for each of the dictionaries $W^{(l)}$, $l = 1, \ldots, L$ in order to obtain an intuition about the proposed M-step update equations. We are specifically interested in finding possible similarities between the current approach and other

methods for estimating the model's parameters (see, e.g., Wonnacott and Wonnacott, 1990 for some details regarding the parameter estimation of the exponential family distributions).

To this, consider the family of generative models presented in (3.25)-(3.27) together with the link function (3.38), and also assume the noise distribution to be an arbitrary and regular L-parameter distribution of the exponential family. Therefore, we have dictionaries $W^{(l)}$ for $l = 1, \ldots, L$ as parameters of the model. Analytically, obtaining a closed-form solution for each of the dictionaries $W^{(l)}$ (which is in general applicable to all distributions of the family) is infeasible (or at least very complicated). Nevertheless, we proved that Equations (4.36) can provide a condition for the stationary point of the free energy (4.4). Regardless of the other theoretical tests that we could apply in order to further investigate the details of these stationary points (to find out if they represent a maxima or a minima or even a saddle point), we used these equations in a fixed-point sense (as presented in (4.44)) to iteratively update our dictionaries $W^{(l)}$. This has been shown to work in practice and monotonically increase the free energy and also the log-likelihood functions to at least a local optima. Now, let us here have a close look into the Equations (4.44) for an arbitrary $1 \leq l \leq L$; that is:

$$\begin{aligned}(W_{dh}^{(l)})^{\text{new}} &= \frac{\sum_n \langle \mathcal{A}_{dh}(\vec{s}, \Theta^{\text{old}}) \rangle_{q^{(n)}} T_l(y_d^{(n)})}{\sum_n \langle \mathcal{A}_{dh}(\vec{s}, \Theta^{\text{old}}) \rangle_{q^{(n)}}} \\ &= \frac{\sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \mathcal{A}_{dh}(\vec{s}, \Theta^{\text{old}}) T_l(y_d^{(n)})}{\sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \mathcal{A}_{dh}(\vec{s}, \Theta^{\text{old}})}.\end{aligned} \tag{4.50}$$

For each data point $y^{(n)}$ and indices $d = 1, \ldots, D$ and $h = 1, \ldots, H$, we now define the weight $\mathcal{B}_{dh}^{(n)}(\Theta)$ as follows:

$$\mathcal{B}_{dh}^{(n)}(\Theta) := \sum_{\vec{s}} q^{(n)}(\vec{s}) \mathcal{A}_{dh}(\vec{s}, \Theta). \tag{4.51}$$

Therefore, we can write:

$$(W_{dh}^{(l)})^{\text{new}} = \frac{\sum_n \mathcal{B}_{dh}^{(n)}(\Theta^{\text{old}}) T_l(y_d^{(n)})}{\sum_n \mathcal{B}_{dh}^{(n)}(\Theta^{\text{old}})}. \tag{4.52}$$

As it can be seen, the above equation represents the weighted sample mean of the sufficient statistics $T_l(y)$ that is computed in each M-step given the old parameters Θ. In other words, it can be inferred that the maximum likelihood estimator used here simply sets the dictionary $W^{(l)}$ to be equal to a specific weighted sample mean of the sufficient statistics $T_l(y)$. To further illustrate, assume indices h and d to be given. For each vector $\vec{s}$, if the cause h has the highest weight value amongst the other causes, i.e. $\mathcal{A}_{dh}(\vec{s}, \Theta) = 1$ (note that in this case $\mathcal{A}_{dh'}(\vec{s}, \Theta) = 0$, $\forall h' \neq h$), then the corresponding value of $q^{(n)}(\vec{s})$ will be added to the function $\mathcal{B}_{dh}^{(n)}(\Theta)$ in (4.51) which later affects the actual weight on $T_l(y_d^{(n)})$. We finally sum the amounts of $\mathcal{B}_{dh}^{(n)}(\Theta) T_l(y_d^{(n)})$ for each n and compute the average (4.52) which will be inserted at the position of the matrix with indices h and d. In addition, observe that the weight $\mathcal{B}_{dh}^{(n)}(\Theta)$ depends on the index matrix $\mathcal{A}_{dh}(\vec{s}, \Theta)$ which, in a broader sense, can be seen as a masking operator with values equal to 0 and 1. In fact, for each index h and for each hidden state $\vec{s}$, the vector $(\mathcal{A}_{1h}(\vec{s}, \Theta), \ldots, \mathcal{A}_{Dh}(\vec{s}, \Theta))^T$ can be described as the *mask* of cause (object) h. Such a concept has been frequently used before, e.g., for *occlusive* models and specifically for images to distinguish the order of different objects in depth (see, e.g., Lücke et al., 2009; Dai, Exarchakis, and Lücke, 2013; Henniges et al., 2014 for a detailed study).

This is one straight property of the maximum function defined in (3.38) (as the combination rule) where the dominant cause (e.g., object) occludes the other causes and sets the mean of

observable d (for the images this would be the pixel value of d). Here, we observe a similar masking effect for the updates (4.44) as the previously used MCA models, but with some small differences. In studies like (Lücke et al., 2009; Henniges et al., 2014), masks are considered as a parameter of the model (describing the contribution that each cause or object makes to each pixel or observable). This parameter can then be learned using an EM algorithm. In contrast, given the parameters Θ, we here compute a masking operator $\mathcal{A}_{dh}(\vec{s}, \Theta)$ (which yields values 0 and 1) for each hidden state $\vec{s}$ and for each cause h and observable d. Moreover, the link function (3.38) defined here can be seen to put one step forward and introduce an *exclusive* property to our generative models. That is, the dominant cause not only sets the mean parameter for each observable d, but also sets the variance (or other moments of the noise distribution). The motivation behind such an exclusive model was suggested by medical data analysis and our running example of analyzing the causes/symptoms relations necessitated the existence of such a combination.

One important case is when the noise distribution has sufficient statistics proportional to y, i.e. $T_1(y) = y$. In Corollary 1, we mentioned such commonly encountered distributions of the exponential family and showed that the following update equation for the weight matrix (component means) can be obtained accordingly:

$$W_{dh}^{\text{new}} = \frac{\sum_n \langle \mathcal{A}_{dh}(\vec{s}, \Theta^{\text{old}}) \rangle_{q^{(n)}} y_d^{(n)}}{\sum_n \langle \mathcal{A}_{dh}(\vec{s}, \Theta^{\text{old}}) \rangle_{q^{(n)}}}. \tag{4.53}$$

Consequently, using definition (4.51), we can rewrite the above equation as:

$$W_{dh}^{\text{new}} = \frac{\sum_n \mathcal{B}_{dh}^{(n)}(\Theta^{\text{old}}) y_d^{(n)}}{\sum_n \mathcal{B}_{dh}^{(n)}(\Theta^{\text{old}})}. \tag{4.54}$$

For this case, let us now consider one special approach where, e.g., the set of hidden states $\mathcal{S}$ has only one vector $\vec{s}$. Then, summation over all hidden states will reduce to only one term. This could be either of the following two cases:
(A) If we set $|\mathcal{K}^{(n)}| = 1$ (for each n) in our truncated approximations (in TV-EM presented in Section 4.4), where $|\cdot|$ denotes the number of elements in the subset $\mathcal{K}^{(n)}$. This can also be seen as a specific type of MAP approximations that have been widely researched in many studies. A relevant study in this direction is the work by Lücke and Forster (Lücke and Forster, 2019) that shows applying the TV-EM for Gaussian mixture models (GMMs) with isotropic Gaussians yields the well-known k-means clustering algorithm (Lloyd, 1982).
(B) If we train our model supervised. That is, we let $\vec{s}_{true}^{(n)}$ to be the true label that indicates the active causes for data point $y^{(n)}$. Consequently, we assume $q^{(n)}(\vec{s}_{true}^{(n)}) = 1$ for each n and let $q^{(n)}(\vec{s})$ for the other hidden states to be equal to zero (i.e., $\mathcal{B}_{dh}^{(n)}(\Theta) := \mathcal{A}_{dh}(\vec{s}_{true}^{(n)}, \Theta)$).
For $d = 1, \ldots, D$ and $h = 1, \ldots, H$, we then define (note that this definition is valid for both cases A and B above):

$$I_{dh} := \{n \mid 1 \leq n \leq N,\ \mathcal{A}_{dh}(\vec{s}_{true}^{(n)}, \Theta) = 1\}. \tag{4.55}$$

Hence, we can write:

$$\sum_n \mathcal{B}_{dh}^{(n)}(\Theta^{\text{old}}) y_d^{(n)} = \sum_n \mathcal{A}_{dh}(\vec{s}_{true}^{(n)}, \Theta^{\text{old}}) y_d^{(n)} = \sum_{n' \in I_{dh}} y_d^{(n')}$$

and also

$$\sum_n \mathcal{B}_{dh}^{(n)}(\Theta^{\text{old}}) = \sum_n \mathcal{A}_{dh}(\vec{s}_{true}^{(n)}, \Theta^{\text{old}}) = \sum_{n' \in I_{dh}} 1 = |I_{dh}|.$$

Therefore, the update for the weight matrix W (in a matrix form) can be restated as:

$$W^{\text{new}} = \frac{\begin{pmatrix} \sum_{n' \in I_{11}} y_1^{(n')} & \cdots & \sum_{n' \in I_{1H}} y_1^{(n')} \\ \vdots & \ddots & \vdots \\ \sum_{n' \in I_{D1}} y_D^{(n')} & \cdots & \sum_{n' \in I_{DH}} y_D^{(n')} \end{pmatrix}_{D \times H}}{\begin{pmatrix} |I_{11}| & \cdots & |I_{1H}| \\ \vdots & \ddots & \vdots \\ |I_{D1}| & \cdots & |I_{DH}| \end{pmatrix}_{D \times H}}. \tag{4.56}$$

As it can be seen, for each W_{dh}, the sample mean is computed along the subset I_{dh} in the right hand side of the equation above. For a set of i.i.d. random variables $\vec{y}^{(n)}$, we know that the sample mean converges to the distribution mean as the number of observations increases (i.e. $N \longrightarrow \infty$). However, for a large value of N, the distribution mean should be well approximated by the sample mean. This assumption is the basis of approaches such as the *method of moments* (Bowman and Shenton, 2014). In such methods, to estimate the parameters of a L-parameter distribution, the number of L moments, whose probability limits are known functions of the L parameters of the distribution, are computed (note that the sample mean represents the first moment of the distribution). Next, these L moments are equated to the L functions and these functions are then inverted to express the parameters as functions of the moments (see, e.g., Bowman and Shenton, 2014; Mátyás, 1999; Hall, 2005 for further details). Here, only the parameters of the first moment (the component means) are estimated as in (4.56). In general, however, a combination of dictionaries, e.g. W and V, should be considered to estimate the other moments of the distribution (similar to the Gaussian distribution stated in Examples 1 and 2 and Section A.1).

In addition, we here observed that the update (4.53) can indeed be used for many of the exponential family distributions including Gamma, Gaussian, Bernoulli, Poisson, Exponential etc. As stated in Corollary 1, the common factor of all these distributions is the fact that $T_l(y) = y$ (for at least one l) satisfies them. For these distributions, it can be shown that MLE is similar to the method of moments as, in both cases, a specific sample mean is used to estimate the parameters of the mean. Such a close relationship, to our best knowledge, has not been discussed before for LVMs with the general case of exponential family distributions. In this section, nonetheless, we took the first steps to find some similarities between the generally applicable update equations presented here and other statistical approaches for parameter estimation. In addition, we know that regardless of the choice of distribution, the sample mean is approved to represent a good estimation of the distribution mean for sufficiently large amounts of data (see, e.g., Wonnacott and Wonnacott, 1990). We therefore conjecture that the update (4.53) can be applied for updating the component means (elements of our matrix $M(\Theta)$) of any considered noise distribution. In other words, the update (4.53) can, in practice, be used to estimate the matrix $M(\Theta)$ of the Beta distribution and other complex distributions of the family which $T_l(y) = y$ does not apply for them. However, to our best knowledge, there is no theoretical proof (nor will we provide such a proof here) stating that this update can maximize the free energy or log-likelihood functions for the proposed generative model with Beta noise. Such investigations are far from our goal here and we only opened this discussion to show the potential for further studies that can be done in this direction.

Chapter 5

Experimental Results

In this chapter, we examine the performance of proposed generative models in extracting important patterns, statistical correlations and causal structures from a set of unlabelled data. We will investigate our models using both artificial and real datasets and further point out some practical applications for which the presented models can be used. Artificial data will be first employed to numerically verify the reliability of the proposed parameter optimization approach (the derived update equations (4.27)-(4.28) and (4.33)) using different well-known distributions of the exponential family as examples. Concretely, we will use Bernoulli, Poisson, Exponential, Gaussian, Gamma and Beta distributions and optimize the corresponding ef-MCA models with the EM algorithm presented by Algorithm 1. In other words, the data we study here includes binary (corresponding to the Bernoulli distribution), count (corresponding to the Poisson distribution), real (corresponding to the Gaussian distribution), positive (corresponding to the Exponential and Gamma distributions) and continuous interval (corresponding to the Beta distribution). We refer to each of the considered models as the Bernoulli-MCA, Poisson-MCA, Exponential-MCA, Gaussian-MCA, Gamma-MCA and Beta-MCA generative models. Bernoulli, Poisson and Exponential are one-parameter distributions and are therefore subject to one dictionary W (which corresponds to the component means since $T(y) = y$ holds for all these three distributions). Besides, Gaussian, Gamma and Beta are two-parameter distributions and require a double-dictionary approach. We discussed the details of all these six distributions in Section 3.1.2 and will further provide extra details for the parametrization of the Gaussian, Gamma and Beta distributions in Appendix A. In addition, we use the results of Theorems 2 and 3 to update the parameters of the considered ef-MCA models in each M-step. For E-steps, we use the full posteriors when the used models are sufficiently small (for the bars tests and also for medical data analysis we used exact EM) and apply truncated variational approximations (see Section 4.4) otherwise.

After verifying the presented update equations, we will exploit a few example applications of the proposed non-linear LVMs using real datasets. To this, we will first assess the performance of the models in estimating the causes/symptoms relations for a collected medical data of hearing impairments. For this task, we will use Beta-MCA as a model to encode continuous interval data and compare the performance of this model with the probabilistic noisy-OR presented in Section 2.4. Also the details of M-step update equations for the used noisy-OR model can be found in Section A.5. Thereafter, we will study feature extraction, noise type estimation and denoising tasks. These experiments are specifically dedicated to demonstrate the scalability of the proposed non-linear LVMs here using truncated variational approximation in the form of EEM. In this context, we train our models with very high values of H (e.g., $H = 1000$), which is an intractable task for many of the similar approaches.

In short, in all the upcoming experiments, we consistently observed an efficient performance of the investigated generative models in producing interpretable and high-quality results. In some cases, e.g. denoising, the used models can outperform many of the similar methods and obtain competitive results compared to the state-of-the-art approaches. In addition, the investigated ef-MCA models observed to reliably express the underlying generating process

of the data (for both artificial and real datasets). Such information enables researchers to build representations of the observables that can be used for reasoning, predicting, decision making and ultimately optimal inference. Also we will state whenever a background model is used for the corresponding ef-MCA models and/or if a maximum magnitude superposition is applied. Moreover, for some technical details of the experiments or some extra results we will refer the readers to Appendix B.

The content of the following sections is based on (Mousavi, Drefs, and Lücke, 2020), (Mousavi et al., 2020) (currently under review) and (Mousavi et al., 2021). As the first author of these contributions, I was mainly responsible for the design and preparation of the experimental results in consultation with Jörg Lücke and other co-authors. In the following, I detail the contributions of all co-authors:

I implemented, tested and validated the used ef-MCA models with the help of Jakob Drefs. To this, I used a general implementation platform which has been developed by Jakob Drefs and Enrico Guiraud. This corresponds to the EEM algorithm and the TV-EM implementation package which I embedded my ef-MCA models to and used the codes for large-scale experiments. Moreover, I performed the bars test experiments with Exponential-, Gamma and Beta-MCA models and produced the corresponding figures; and the experiments with Bernoulli-MCA were performed by Florian Hirschberger. The trick for avoiding local optima was suggested by Jörg Lücke and I implemented and examined the idea. The corresponding Section 5.1.3 was then prepared and written by me in consultation with Jörg Lücke. Florian Hirschberger further employed the experiments for reliability of the Bernoulli-MCA model in Section 5.1.4. Regarding the medical data analysis, Mareike Buhl prepared the data and also wrote the parts regarding the CAFPA values (these details are taken from Mousavi et al., 2021). Further, experiments with the noisy-OR model were performed by Enrico Guiraud and I performed the experiments with the Beta-MCA model. In addition, the idea of producing disease profiles and also generating simulated CAFPAs for data augmentation was suggested by Jörg Lücke which I further exploited. Analysis of the ROC curves have been further investigated by me and Jörg Lücke. For natural image patches and the task of feature extraction, Jakob Drefs performed the experiments with the Gaussian-MCA model and the experiments with the Bata-MCA model were performed by me. Also, the experiments with Gaussian-MCA for the noise type estimation section were performed by Jakob Drefs while I performed the experiments with the Gamma-MCA model. Further preparations and analyses carried out jointly by me, Jakob Drefs and Jörg Lücke. Also Jakob Drefs designed the tests on acoustic data and applied the basic preprocessing methods. Moreover, I conducted all the denoising experiments presented in this chapter, which were conceived and analyzed in consultation with Jörg Lücke and Jakob Drefs.

5.1 Numerical Verification of the Proposed Update Equations

In order to validate the efficacy of the presented updates (4.27)-(4.28) and (4.33) (based on the results of Theorems 2 and 3), we here exploit standard bars tests using artificial datasets. The bars test (Földiák, 1990; Hinton et al., 1995; Hoyer, 2004; Spratling, 2006; Lücke and Sahani, 2008) is a well-known task for non-linear component extraction that has become a popular benchmark in the field. We consider Exponential-, Bernoulli-, Gamma- and Beta-MCA models in this section and evaluate their performances. In detail, we use $H = 10$ basis functions $\vec{M}_h(\Theta)$ (for Exponential, Bernoulli and Gamma distributions $M(\Theta) = W$) in the form of horizontal and vertical bars each occupying 5 pixels on a $D = 5 \times 5$ grid (i.e., each of the inputs $\vec{y}^{(n)}$ is a 25-dimensional vector). We here consider the same intensity for the bars (to be uniform and equal) as it is a common procedure for many of the similar tasks. The latents are then sampled independently (according to the prior distribution (3.25)) with probability

π_h for $h = 1, \ldots, H$, and the corresponding generative fields are superimposed non-linearly according to the maximum function defined in (3.38).

Having the ground-truth parameters Θ for each of the MCA models, we artificially generate data points according to one of the MCA models considered above (this results in datasets with Exponential, Bernoulli, Gamma or Beta noise). The corresponding ef-MCA models are then optimized (using the exact EM presented in Algorithm 1) for each of the generated datasets. We are particularly interested in how well the updates can recover the true generating parameters, and to further observe how robust the Algorithm 1 is (how much different conditions such as initialization, sparsity, noise level etc may affect the convergence of the algorithm). In general, for data generated by the corresponding data model itself, parameters should, for sufficiently many data points and modulo symmetries, be recovered with very high accuracy.

5.1.1 One-parameter distributions

We first consider Exponential and Bernoulli as one-parameter distributions of the exponential family to assess the efficacy of updates (4.27) and (4.33) in increasing the log-likelihood of the data using Exponential- and Bernoulli-MCA models.

The Exponential-MCA model

We generated $N = 1000$ i.i.d. data points (drawn independently per pixel) according to the Exponential-MCA (E-MCA) model (the data is corrupted with Exponential noise). We assumed $\pi_h^{gen} = 0.2$ for $h = 1, \ldots, H$ (two active bars on average per data point), and the ground-truth parameter W (note that here $M(\Theta) = W$) is set to a value of 10 for the bars and 1 for other pixels (non-bar pixels). We then trained the E-MCA model on the generated data and updated the parameters $\Theta = (\vec{\pi}, W)$ using Equations (4.27) and (4.33). Note that after choosing the Exponential distribution as our example, Equation (4.27) can be used directly in each M-step and no additional derivations are required. That is, we only require the evaluation of the joint probabilities in each E-step which are directly given by the prior and the used noise model (here Exponential distribution).

We initialized W by randomly choosing H data points and appending them to each other in the form of a $D \times H$ matrix; also the values π_h were initialized at 0.3 for $h = 1, \ldots, H$. We then performed 50 full EM iterations where, in each iteration, the M-step fixed-point (4.27) was (for simplicity) applied just once. In this manner, one can also perform the fixed-point equations of the M-step multiple times to assess the convergence rate of the algorithms (this can be examined for any of the ef-MCA models). Here, we mainly observed similar behaviours of the algorithms when running the fixed-point equations multiple times compared to running it only once.

We repeated this procedure 100 times and closely examined the behaviour of the log-likelihood function for each of the runs. The evolution of the log-likelihood for these 100 runs is then depicted in Figure 5.1-A. As it can be observed, the update equations of Theorem 2 and Theorem 3 give rise to a robust and reliable algorithm that monotonically increases the log-likelihood of the data to at least a local log-likelihood optimum (note that we used full posteriors here). Figure 5.1-A further illustrates that the algorithm converges to the ground-truth log-likelihood value (the red line) in many of the runs. For these cases, the learned parameters are very close to the generating ones (with small differences due to finite sample size effects). Nevertheless, local optima can be observed for some cases where the learned log-likelihood values differ from the ground-truth value. For these runs, the model did not restore the ground-truth parameters (learned generative fields contain, e.g., multiple bars).

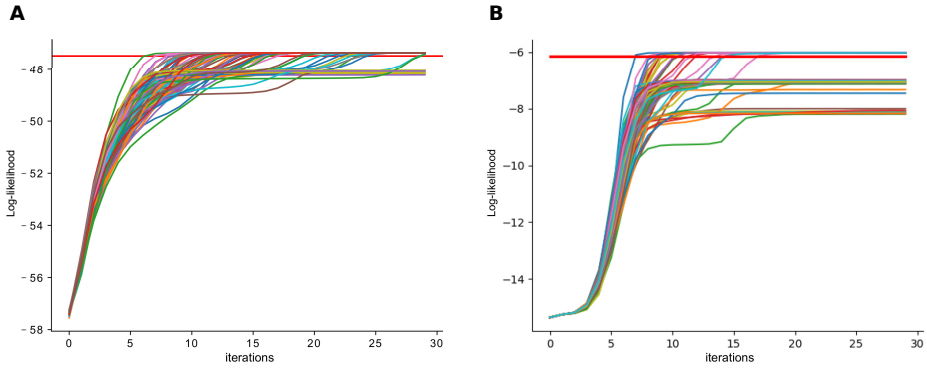


FIGURE 5.1: Behaviour of the log-likelihood function corresponding to (**A**) Exponential-MCA and (**B**) Bernoulli-MCA for 100 runs using artificial bars tests. Here only the first 30 iterations are depicted. Also, different colors denote different runs of the algorithms.

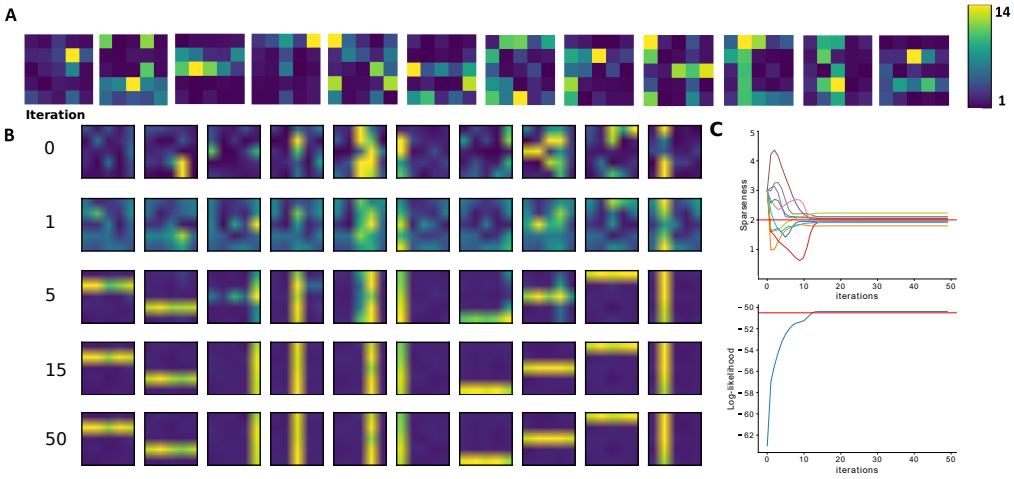


FIGURE 5.2: **A** 12 examples of input data. **B** Learned generative fields using the E-MCA model where the number of iterations illustrated on the left hand side. **C** Behaviour of the log-likelihood (bottom) and sparseness ($\pi_h H$; top) for $h = 1, \ldots, H$; the generating parameter is $\pi_h H = 2$ for all h. See text for further details.

Finally, Figure 5.2 displays the model parameters learned in one specific run of the E-MCA algorithm. As illustrated, the generative fields are learned quickly after approximately 15 iterations and then remain at the convergence point for the rest of iterations (in fact we still observe a monotonic increase of the log-likelihood values, but the changes after convergence are very little). The prior parameters π_h are also learned with a good precision. In addition, the log-likelihood value corresponding to the learned parameters is observed to be a little higher than the ground-truth value, which can be explained as the effect of overfitting. With increasing the amount of data points ($N \longrightarrow \infty$), we assume that both of these inaccuracies will vanish.

The Bernoulli-MCA model

For the second experiment, we considered the Bernoulli-MCA model and generated $N = 1000$ i.i.d. data points according to this model (meaning that the data is Bernoulli distributed). In

detail, we assumed a matrix W with values 0.99 for the bars and 0.01 for the other pixels. The ground-truth value of π_h for each h is similarly set to be 0.2. We then applied the same procedure as above but this time we used the Bernoulli-MCA model (for amount of 50 EM iterations) to recover the ground-truth parameters Θ. Note that the two models differ from each other only in their E-steps (the M-steps are analogous).

We repeated the experiment 100 times by randomly initializing the parameters and closely examined the evolution of the model's parameters. In short, we observed the Bernoulli-MCA to perform similar to the E-MCA where in most cases, the original parameters are obtained with a good precision. However, the effect of local optima is observed to be more severe in this case. Figure 5.1-B then shows the behaviour of the log-likelihood function for this model given the corresponding 100 runs.

5.1.2 Two-parameter distributions

In this section, we consider Beta and Gamma as two-parameter distributions of the exponential family and examine their corresponding MCA models. For these two models (Beta- and Gamma-MCA) we have two matrices that we should update in the M-steps together with the prior parameter $\vec{\pi}$. We consider $\Theta = (\vec{\pi}, W, V)$ as parameters of the two models, but specifically, set the component means ($M(\Theta)$) and component variances ($\Sigma^2(\Theta)$) as the ground-truth parameters and learn them using the two generative models. That is, we use the results of Theorems 2 and 3 to train the models and in each iteration compute matrices $M(\Theta)$ and $\Sigma^2(\Theta)$ using the updated matrices W and V. It should be mentioned that for the specific case of Beta-MCA, we require the function F defined in (3.37) in order to compute the weight matrix $M(\Theta)$, while for the Gamma-MCA $M(\Theta) = W$. Further details regarding the parameter updates of the Gamma- and Beta-MCA models are presented in Sections A.2 and A.3, respectively. Moreover, for these two models, we additionally consider an extra cause as the background which will be explained further below (also see Section 3.4.1 for more details).

The Beta-MCA model

To generate data for this task, we likewise assumed $H = 10$ basis functions $\vec{M}_h$ in the form of horizontal and vertical bars. Here we used values of 0.9 for each observable representing a bar (and a bar occupies 5 observables/pixels in a $D = 5 \times 5$ image) and zero for the other observables (non-bar pixels). Together with these basis functions, a background $\vec{M}_B$ with pixel values of 0.08 and 0.22 (a checkerboard) is assumed, which ensures the lowest value of the observation mean to be greater than zero (see Figure 5.3-B). Moreover, we here deliberately chose two basis functions with the same mean values but with different variances (causes 1 and 5 in Figure 5.3-B) in order to assess the ability of the model in distinguishing between the two causes. Such a task is formidable for most of previously established generative models as they learn a global variance parameter. On the contrary, the proposed Beta-MCA model (as one example of the family of generative models (3.25)-(3.27)) is capable of training component variances alongside component means.

For $\vec{\Sigma}_h$ values, we considered different horizontal bars with pixel values ranging from 0.1 to 0.2. The ground-truth patterns for variances were deliberately chosen to be different from the mean patterns (to better illustrate the modelling capabilities). Also, the background $\vec{\Sigma}_B$ was chosen to also (like the mean) have a checkerboard shape (with low pixel values of 0.01 and 0.05) to distinguish the background variance pattern from other patterns. Furthermore, similar to the previous experiments, each of the basis functions (except the background π_B) are appeared independently with the probability of $\pi_h = 0.2$ (meaning two bars, in average, are active for each data point) and then superimposed exclusively according to the maximum

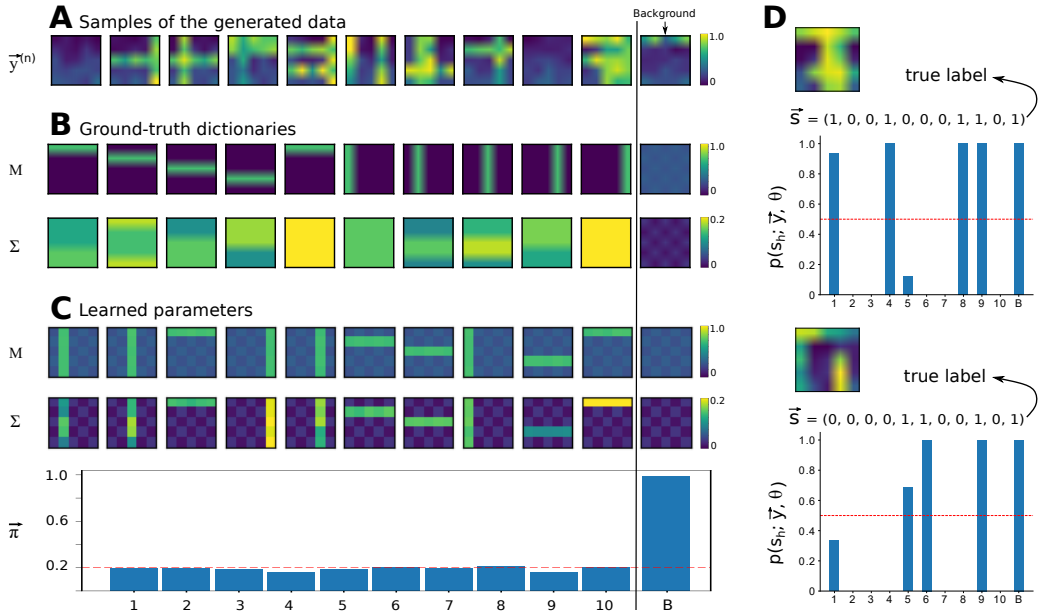


FIGURE 5.3: **A** 11 examples of input data. **B** Ground-truth mean and standard deviation (SD) dictionaries (M and Σ) used to generate data. As illustrated, generative fields s_1 and s_5 have the same mean values but different variances. **C** Learned mean and SD dictionaries together with learned π_h values after 50 EM iterations. As observed, the model is able to learn all parameters with a fairly good precision. Importantly, the learned background patterns and also learned SD dictionaries reveal the robustness of the model in training interval data. **D** Two example data points that are chosen such that cause s_1 is active for one of them (the upper datum) and s_5 is active for the other (the lower datum). The posterior values $p(\vec{s};\vec{y},\Theta)$ given the learned parameters are then computed for these two data points and the results are depicted accordingly. As it can be seen, the model successfully distinguishes between the two causes with the same mean values but with different variances, and infers the active causes correctly. Observe that the model still assigns a low probability to cause s_5 for the upper case and to cause s_1 for the lower case.

function defined in (3.38). Note that for $d = 1,\ldots,D$, we demand the mean and variance of the Beta distribution to be defined as follows:

$$\mu_d := \bar{M}_d(\vec{s},\Theta) = M_{dh(d,\vec{s},\Theta)}(\Theta) \quad \text{and} \quad \sigma_d^2 := \bar{\Sigma}_d^2(\vec{s},\Theta) = \Sigma^2_{dh(d,\vec{s},\Theta)}(\Theta) \tag{5.1}$$

where $h(d,\vec{s},\Theta)$ is given by (3.38).

We generated $N = 1000$ i.i.d. data points according to the Beta-MCA model (see Figure 5.3-A for the illustration of such data points). Next, we used the exact EM procedure presented in Algorithm 1 to recover the ground-truth parameters (50 EM iterations were observed to be sufficient). To initialize matrices W and V, we randomly chose $H+1$ data points and then computed the values of $\log(y)$ and $\log(1-y)$ and rearranged the results into two different matrices with $D\times(H+1)$ entries. The matrix obtained from the values of $\log(y)$ used for the initialization of W and the other for the initialization of V. We also initialized π_h values at 0.3. The results are then presented in Figure 5.3.

As it can be seen, the model extracted all bar patterns of the mean. All basis functions of the mean M have also learned traces of the checkerboard background. The reason for this is the maximum combination and the fact that the background is always present. Note that the likelihood or free energy does not change for values between zero (the non-bar pixels' values) and the background values since the maximum combination (3.38) always chooses the highest

value. Therefore, the learning stops once the value of the background pixel is reached and there is no reason for the algorithm to converge to any other non-bar pixels' values (this is only the case where a background exists). The variance basis functions (in the figure we presented the standard deviation basis functions $\vec{\Sigma}_h$) do seem to converge to patterns very different from the generating (i.e. ground-truth) patterns. At a closer inspection, also the variance patterns are optimal in the likelihood sense, however. Again, the maximum non-linearity explains the divergence from the original patterns: For non-bar pixels of component means (with zero values), the background will always be larger. The variance of any non-bar pixel is consequently never used to generate a bar (is irrelevant for the likelihood value). The only variance values that are important, are consequently the variance values for those pixels that correspond to high mean values and also the background values. For those pixels, the learned variance values estimate the ground-truth variances well. In fact, observe that the position of the bars in $\vec{\Sigma}_h$ values for $h = 1, \ldots, H$ coincides with the position of the bars in basis functions $\vec{M}_h$ which is due to the maximum function in (3.38) indicating the dominant cause. Nevertheless, the learned $\vec{\Sigma}_h$ values are observed to be slightly noisy that can be seen as the effect of finite sample size. Here, we further found that the basis functions are mostly learned in the first few iterations which ensures the applicability of the update equations.

Importantly, the model can distinguish between the two causes with the same mean values and different variances. In this context, Figure 5.3-D illustrates the inference results for two unseen data points where each of them has only one of these two causes as the active cause. The model can then correctly infer the true active causes for these two data points (even though it still assigns a low probability for the activation of the other cause with the same mean value). Finally, we observed that the Beta-MCA algorithm estimates the generating π_h values well.

In similar experiments, however, we observed local optima to be a serious issue for this task. That is, when executing several runs with different initializations, we also frequently observed convergence to lower log-likelihood values (convergence to local optima). When measuring the number of bars for the mean that can be correctly inferred after learning, we find an average of 8 bars out of 10 across different runs. However, as we can compute the log-likelihood exactly, we are able to simply use the run with the highest log-likelihood out of several runs. Alternatively, it would be possible to introduce additional annealing strategies (see, e.g., Lücke and Sahani, 2008 for a concrete discussion).

Anyhow, to assess the robustness of the proposed model, we performed a similar experiment 100 times and each time with a new initialization (we used one specific set of data points for all these runs). We then examined the behaviour of log-likelihood and free energy functions closely. In these runs, we considered a simpler task than the experiment above (we did not use causes with the same means and different variances) in order to have a better comparison. Although we have seen the local optima effect in many of these runs, a monotonic increase of the free energy function and also the log-likelihood of the data is observed in all these different runs. Behaviour of the log-likelihood function is then depicted in Figure 5.4-A (for the case of experiment above, the amount of runs converged to the ground-truth value observed to be much lower than the current experiment). Besides, we again emphasize that no annealing procedure is applied here to avoid the local optima problem. Later, we will propose a method to avoid local optima that can be seen useful for some of the ef-MCA models. Overall, the results approve the strength and reliability of the proposed Beta-MCA model and also the presented update equations in recovering the ground-truth parameters.

The Gamma-MCA model

Next, we considered the Gamma-MCA model and likewise generated $N = 1000$ i.i.d. data points according to this generative model (the data is therefore Gamma distributed). For this experiment, we assumed basis functions $\vec{M}_h$ with the values of 10 for pixels representing a

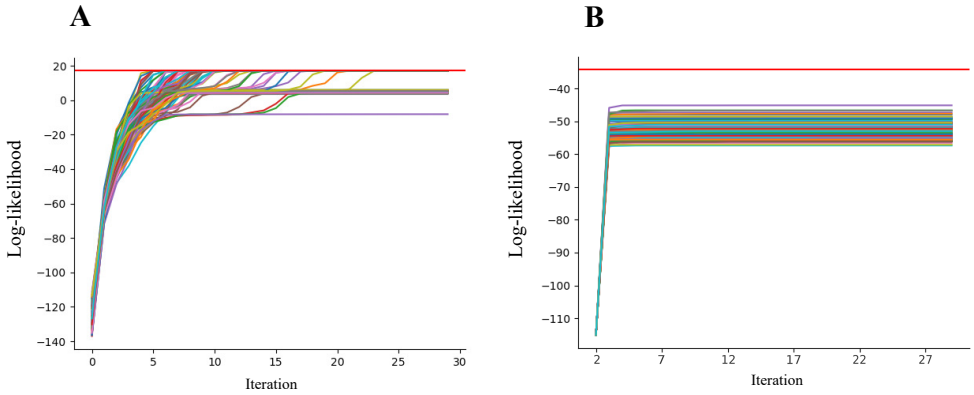


FIGURE 5.4: Behaviour of the log-likelihood function corresponding to (**A**) Beta-MCA and (**B**) Gamma-MCA for 100 runs using artificial bars tests. Here only the first 30 iterations are depicted. Also, different colors denote different runs of the algorithms. For the Gamma-MCA model, we depicted the log-likelihood values after the second iteration for the sake of a better visualization.

bar and zero for non-bar pixels. Together with these basis functions, we further assumed a background $\vec{M}_B$ with pixel values of 1.0. Likewise, the background ensures that the lowest value of the observation mean is 1.0. Furthermore, we considered $\vec{\Sigma}_h$ functions with pixel values of 1.0, and also $\vec{\Sigma}_B$ (the background) with pixel values of 0.5. The ground-truth value of π_h (the prior parameter) for each h is also set to be 0.2.

We then employed the Gamma-MCA model on this dataset and initialized the model parameters similar to the settings of the Beta-MCA model above. That is, we randomly chose $H+1$ data points and rearranged them to obtain a $D \times (H+1)$ matrix. The obtained matrix used for the initialization of W (note that $M(\Theta) = W$) and also the log of this matrix is used for the initialization of V. Moreover, we initialized the prior parameters at 0.3 and applied 50 EM iterations (similarly exact EM is used here). We repeated this experiment 100 times each time with a new set of initializations and carefully examined the behaviour of log-likelihood and free energy functions. Similar to the previous experiments, the proposed update equations (4.27)-(4.28) and (4.33) (which obtained directly as the results of Theorems 2 and 3) presented a robust and reliable algorithm for the Gamma-MCA model, and resulted in a monotonic increase of the log-likelihood function (to at least a local optima). Figure 5.4-B then shows the evolution of the log-likelihood for these runs. The effect of local optima, however, is observed to be the most severe for the Gamma-MCA as we (for all the runs) observed convergence to log-likelihood values lower than the ground-truth. Such an effect can be attributed to the inherent properties of the Gamma distribution so that the Gamma-MCA model is not as robust as the other ef-MCA models used here in avoiding local optima. In the next section, we thus design an experiment to assess the local optima issue and further propose a specific approach in order to alleviate the effect of local optima in our experiments.

5.1.3 Avoiding local optima

In our experiments above, we executed each of the algorithms multiple times while holding the hyperparameters fixed and used different realizations of the initial model parameters in each run. In general, we observed that the best solutions recovered all bars with high accuracy. In some cases, however, a slightly overfitting effect occurred which could be diminished by increasing the number of data points. We also frequently observed that the algorithms could not recover the ground-truth generative parameters. This can be seen as the effect of local optima which is different for each of the ef-MCA models. Local optima effects showed, for

instance, to be stronger for the Beta- and Gamma-MCA models compared to Exponential- or Bernoulli-MCA. While variational annealing schemes have frequently been applied as a measure against local optima effects, we here investigated an alternative approach in order to prevent the algorithm from converging to suboptimal solutions. For these investigations, we then used the Gamma-MCA as we observed the local optima effect to be the most severe for this model.

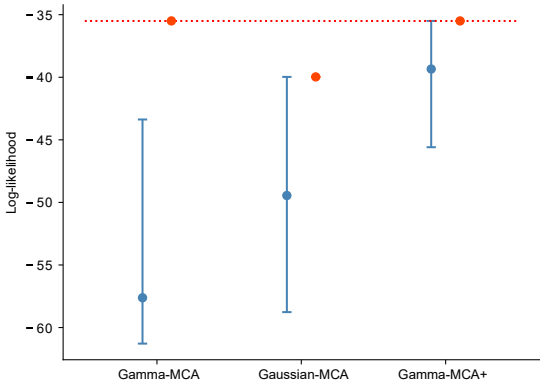


FIGURE 5.5: Comparison of the log-likelihood values for Gamma-MCA, Gaussian-MCA and Gamma-MCA+ algorithms trained on a dataset generated according to the Gamma-MCA model (see text for details). Here, the red color denotes initializing the model parameters Θ at ground-truth values and the blue corresponds to the random initialization. Moreover, the dots denote the average value of the log-likelihood values over 10 different runs; and further, the upper and lower bounds denote the maximum and minimum log-likelihoods achieved in these 10 runs. Finally, the red line represents the ground-truth log-likelihood value. As it can be seen, Gamma-MCA and Gamma-MCA+ have higher log-likelihoods at the optimum solution (red dots) rather than the Gaussian-MCA model which is the case for Gamma distributed data. Moreover, blue lines illustrate the effect of local optima for the Gamma-MCA model and also the beneficiary of the Gamma-MCA+ model. That is, the Gamma-MCA model never achieved the optimal log-likelihood value during these 10 runs and its results illustrate a clear local optima effect.

Concretely, we considered bars test data generated according to the Gamma-MCA model and applied three different algorithms on this dataset: In the first and second case, we applied Gaussian- and Gamma-MCA separately. In the third case, we applied a Gamma-MCA model which (this time) had initialized using the parameters learned by the Gaussian-MCA model. We refer to this latter approach as Gamma-MCA+. We then considered two different scenarios and compared the log-likelihood values of these three algorithms (Gaussian-MCA, Gamma-MCA and Gamma-MCA+). In the first scenario, we initialized the model parameters of the Gaussian- and Gamma-MCA models at the ground-truth values. For the second, we used randomly initialized parameters instead. We then performed 10 runs of each algorithm using 100 EM iterations in each run. The results are depicted in Figure 5.5. As the figure shows, when initializing the parameters at ground-truth, the learned parameters tend to stay at initial values which can be observed consistently for all models. Further, when initializing the parameters randomly, the Gamma-MCA+ shows to achieve higher log-likelihood values in comparison to the other two models. Moreover, the results of the Gamma-MCA model show that the model never achieves the optimal log-likelihood value. Hence, we can conclude that the Gamma-MCA+ approach helps alleviate the effect of local optima. In other words, we

can first apply the Gaussian-MCA model which has been observed to be less sensitive than the Gamma-MCA and then use the learned parameters as the initialization of the Gamma-MCA model. This, in general, can be used for any distribution of the exponential family for which we have difficulties regarding the local optima problem. It should be also mentioned that there have been different studies investigating the local optima, e.g. using deterministic or variational annealing schemes, for the models that are mainly tailored to the Gaussian noise (see, e.g., Ueda and Nakano, 1998; Sahani, 1999; Lücke and Sahani, 2008; Lücke and Eggert, 2010). Such approaches can be further used here in conjunction with the Gaussian-MCA model (but *not* necessarily with other ef-MCA models) in order to improve the results, and to consequently obtain better initializations for the other ef-MCA models. We will later exploit the Gamma-MCA+ in our experiments in Section 5.2.3 instead of the Gamma-MCA model.

5.1.4 Reliability of the proposed non-linear SC approaches

In addition to the local optima issue, the inherent complexity of the model can also affect its performance. That is, for instance, increasing the sparsity ($\vec{\pi}H$) will decrease the *reliability* of the model. The probability of recovering all bars has been termed reliability of the algorithm (Spratling, 2006). This is specifically a measurement that reveals how robust an algorithm is w.r.t. the average number of active bars, and how often it reaches the global vs. any local optimum. In the following, we therefore attempt to examine such a measurement for the family of data models presented in this study. More specifically, we investigate the reliability of the Bernoulli-MCA model (as an example of the family of ef-MCA models) for different levels of sparsity. For our purposes, we notably did not optimize learning to improve reliability (e.g., by introducing annealing procedures; Lücke and Eggert, 2010), but used the canonical form of Bernoulli-MCA with exact E-steps (i.e., full posteriors) and updated W and $\vec{\pi}$ using (4.27) and (4.33).

To this task, we generated artificial datasets according to the Bernoulli-MCA model similar to the procedure presented above with $W_{dh} \in \{0.99, 0.01\}$ and varied the value of π (we assumed $\pi_h = \pi$ for $h = 1, \ldots, H$) that determines the average number of active bars per data point. For each value of π, we generated 200 different bars datasets each with $N = 1000$ i.i.d. data points and then fitted the Bernoulli-MCA model. We then computed 50 full EM iterations; initialization of W_{dh} and π were chosen as in the bars test experiments above. Reliability further measured in terms of the percentage of all trained Bernoulli-MCA models that achieved higher log-likelihood values than the ground-truth (note that this is a good measure as the converged log-likelihoods are slightly higher than the ground-truth value due to the overfitting effect). Results are then presented in Figure 5.6.

As it can be observed, the best runs always reach higher log-likelihood values than the generating parameters (due to slight overfitting). These runs do recover all bars' patterns and the prior parameters. Even for values $\pi = 0.5$, i.e., for five out of ten bars per input on average, two to three out of 200 runs do extract all bars. The best runs in terms of log-likelihood values can automatically be determined without knowledge of the ground-truth such that the resulting method would yield a highly reliable approach to extract all bars for binary data (compare, e.g., Spratling, 2006; Frolov, Húsek, and Polyakov, 2015). We will later design another experiment to show how such a criteria (comparing the log-likelihood or free energy values) can in practice be used to, e.g., estimate the noise type of the data.

5.2 Practical Applications of the Proposed Generative Models

Now, we point out some practical applications of the family of generative models presented in (3.25)-(3.27). We first consider the task of medical data analysis and later exploit our models for feature extraction, noise type estimation and denoising tasks. Note that these are only

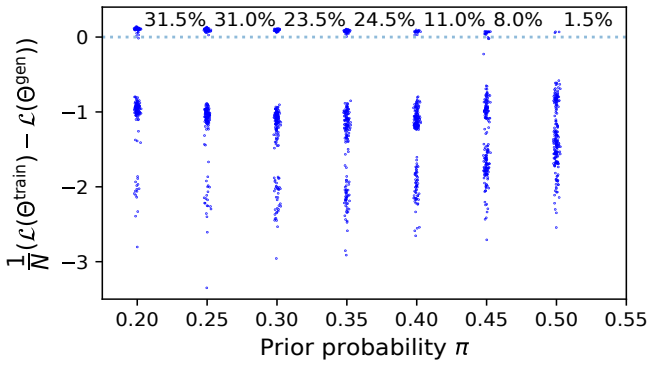


FIGURE 5.6: Differences of the log-likelihoods between trained Bernoulli-MCA models and ground-truth parameters for bars test datasets. Average active number of bars (πH) is set by different prior probabilities $0.2 \leq \pi \leq 0.5$ in steps of 0.05. Individual runs are shown with small offsets along the π-axis for a better visualization. Percentages refer to the fraction of trained Bernoulli-MCA models with higher log-likelihoods than the ground-truth values.

a few examples that we have chosen here and, in general, the application of the proposed models is beyond these experiments. With that in mind, we stress that real-world applications often require large-scale models such that computing full posteriors becomes infeasible. For the first experiment, which is application to medical data, we are able to use full EM as the size of dataset used here is somehow small, but for the other cases, we exploit variational EM. Therefore, these tasks are also designed such that they demonstrate the scalability of the proposed approaches. Moreover, it should be mentioned that these are in general highly challenging tasks for many of the similar approaches that are, e.g., based on fully Bayesian methods. However, we apply truncated variational approximations (see Section 4.4 for details) in the form of truncated posteriors given by (4.47) to scale the algorithms. Truncated posteriors have been used for many MCA models with Gaussian noise (Lücke and Eggert, 2010; Shelton et al., 2017) and have shown promising results in approximating the exact posteriors with a high precision. The approach we use here, as mentioned before, has the additional benefit of being a 'black-box', i.e., given a new distribution of the exponential family, we do not require new derivations to apply the approximation. More specifically, for all the mentioned experiments, we employed EEM (Guiraud, Drefs, and Lücke, 2018) to approximate the E-steps using 60 variational states per subset $\mathcal{K}^{(n)}$ and used the same sliding window averaging technique as studied by (Guiraud, Drefs, and Lücke, 2018).

5.2.1 Application to Medical Data

For the purpose of showing the application of Beta-MCA model to medical data, a dataset containing Common Audiological Functional Parameters (CAFPAs) is used. CAFPAs have been introduced by Buhl et al. (Buhl et al., 2019; Buhl et al., 2020) as an abstract representation of the human auditory system which are independent of audiological tests performed for the respective patients. The data used here comprises CAFPA values determined by an expert survey, where leading clinical audiologists and physicians labelled the database from Hörzentrum Oldenburg, Germany, by indicating audiological findings, treatment recommendations, and CAFPAs for 287 single patients. The CAFPAs are defined on a continuous scale (the interval $[0, 1]$) representing 0 as normal and 1 as pathological and describe different functional aspects of the auditory system: Four CAFPAs are related to hearing thresholds

in different frequency ranges, two CAFPAs are related to suprathreshold deficits below and above 1.5 kHz, and finally one CAFPA for each of the binaural hearing, neural processing, cognitive abilities and socio-economic components. We here excluded the socio-economic CAFPA as it describes the social environment of a patient rather than a physiological cause of hearing impairment. The data are then deliberately de-identified and here, only CAFPAs for the audiological findings of *high-frequency hearing loss* and *broadband hearing loss* are used. This results in amounts of 124 data points with $D = 9$ CAFPA values in which a number of 52 patients with high-frequency hearing loss, 26 with broadband hearing loss, 9 patients with both causes and 37 with normal hearing are included. Moreover, for the sake of computations, a small value of 1.0×10^{-10} is added (subtracted) to the CAFPAs with 0 (1) values.

We then used a model with $H = 2$ (three causes including background) and applied Beta-MCA to learn the causes/symptoms relations between the two diseases and CAFPAs. In the upcoming experiments, we divided the considered data into two sets of training and test using 10-fold cross-validation and further assessed the performance of the proposed model in predicting the true active causes on held-out data. That is, after learning parameters Θ of the model, we computed the following posterior probabilities for an unseen data point y:

$$\mathcal{S} = \begin{Bmatrix} s^{(1)} = (0,0,1) \\ s^{(2)} = (1,0,1) \\ s^{(3)} = (0,1,1) \\ s^{(4)} = (1,1,1) \end{Bmatrix} \Longrightarrow \begin{cases} p(s^{(1)}; y, \Theta) \longrightarrow \text{probability of } y \text{ being healthy} \\ p(s_1 = 1; y, \Theta) \longrightarrow \text{probability of } s_1 \text{ being active for } y \\ p(s_2 = 1; y, \Theta) \longrightarrow \text{probability of } s_2 \text{ being active for } y \\ p(s^{(4)}; y, \Theta) \longrightarrow \text{probability of both } s_1 \text{ and } s_2 \text{ being active} \end{cases}$$

where $\mathcal{S}$ denotes the set of all possible hidden states such that s_1 corresponds to the high-frequency hearing loss and s_2 to the broadband hearing loss. Also note that we here considered the Beta-MCA model with a background (the last element of the hidden states considered above), which corresponds to the symptom statistics of healthy patients.

We further investigated three different datasets: *Real CAFPAs*, *simulated CAFPAs* and *augmented CAFPAs*. The first corresponds to training and testing on amounts of 124 real CAFPA values collected and labelled by the experts; the second denotes the synthetic data that we have generated for our purposes in this study (we will discuss its details further below); and the third corresponds to training on simulated CAFPAs and testing on real CAFPAs. Each of the cases will be discussed in the following.

In addition and for the sake of comparison, we also applied a probabilistic noisy-OR model (see Sections 2.4 and A.5 for details) on the binarized datasets. To this, we compared each observable with an arbitrary and constant threshold α and set the binary output to 0 if the observed value is below the threshold and 1 otherwise. We repeated the experiments for several possible values of the binarization threshold, and the value to yield the best results found to be $\alpha = 0.5$. The corresponding Receiver Operating Characteristic (ROC) curves are then depicted in Figure 5.7 (for all the three settings discussed above) where we used the scikit package *metrics* for computing the area under the curve (AUC) values (we will discuss the details of ROC curves further below).

Considering the middle column of Figure 5.7 (training and testing on real CAFPAs), it can be seen that the Beta-MCA model can outperform the noisy-OR model on predicting the high-frequency hearing loss disease, but it performs worse than noisy-OR for broadband hearing loss. This can be seen as the effect of having very few data points for training the model: Note that the Beta-MCA model has to estimate approximately twice as many parameters as the noisy-OR. Consequently, the effect of finite sample size is more severe for this model. Therefore, in order to obtain a more informed comparison, a set of synthetic data, called simulated CAFPAs, is generated to be used for data augmentation and further analysis of the two diseases.

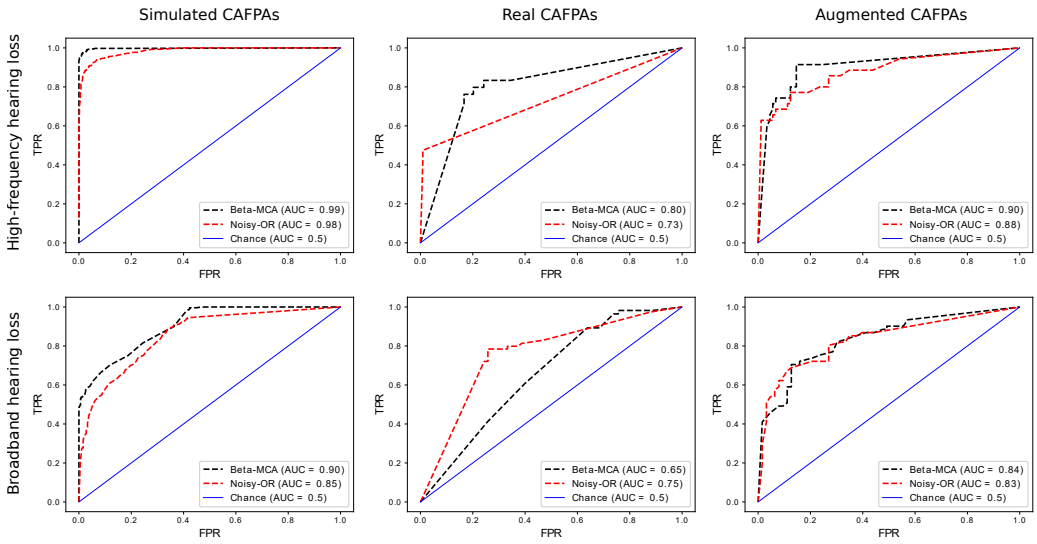


FIGURE 5.7: Illustration of the ROC curves presenting the results of Beta-MCA and noisy-OR models trained on simulated, real and augmented CAFPAs (see text for details). The two models of Beta-MCA and noisy-OR are distinguished with black and red colors, respectively. Moreover, the blue line denotes the prediction results corresponding to tossing a coin (each disease has a 50% chance to be active). The computed AUC values reveal the beneficiary of the Beta-MCA model in 5 out of 6 cases. Moreover, TPR and FPR denote the true and false positive rates, respectively (see text for details and analysis of the presented ROC curves).

Simulated CAFPAs – data augmentation

We first learned the symptom statistics of the two diseases (high-frequency and broadband hearing losses) by fitting a Beta distribution (and also a Gaussian for the sake of comparison) to the data for which only one of the causes is active; i.e., to the data points with only high-frequency hearing loss or broadband hearing loss. Likewise, we fitted a Beta (and a Gaussian) distribution to the amount of 37 data points where none of the causes are active to learn the symptom statistics of a healthy patient. Doing so, we obtained the continuous disease profiles corresponding to the two diseases of high-frequency and broadband hearing losses and also of the normal case. The results of these disease profiles are further presented in Section B.3. In short, we observed that the Beta distribution is a better fit to the CAFPAs rather than Gaussian as higher log-likelihood values are obtained for the case of Beta (see Figures B.4, B.5 and B.6). This is further in line with the results described in (Buhl et al., 2020).

We further used the learned mean and variance parameters (obtained from fitting a Beta distribution) to generate $N = 1000$ data points according to the Beta-MCA model (we refer to it as the simulated CAFPAs which, in a sense, can be used as data augmentation for the aforementioned generative models). It should be mentioned that the π_h values were also computed by fitting a mixture of two Beta distributions to the data in which both diseases of high-frequency and broadband hearing losses are active. The outcome is simulated CAFPAs consisting of 510 data points with high-frequency hearing loss and 436 data points with broadband hearing loss where for 235 cases both diseases are active at the same time.

We then considered the two models of Beta-MCA and noisy-OR and repeated the experiments above but this time using the ($N = 1000$) simulated CAFPAs. The results are illustrated in Figure 5.7 (the first column from the left). Here, we observed that Beta-MCA yields reliable results in inferring the causes/symptoms relations, and in predicting the true active causes for an unseen data point as it performs better in comparison to the noisy-OR model. Finally, for

the third case, we trained the two models on simulated CAFPAs and tested the outcome on real CAFPAs (we refer to this case as the augmented CAFPAs). Considering Figure 5.7 (the first column from the right), we observed that additional data provided by simulated CAFPAs improves the performance of both models. Simulated CAFPAs are consequently a valuable form of data augmentation such that performance (in practice) can be improved. Additionally, we find that Beta-MCA is (as for simulated CAFPAs) now the preferable model as it achieves higher AUC values than noisy-OR with an optimized threshold.

Analysis of the ROC curves

The Receiver Operating Characteristic (ROC) curves have become a benchmark in assessing the reliability of a model in binary classification. The approach has been specifically used in medical data analysis to quantify the accuracy of different models (usually medical diagnostic tests or systems) in discriminating between, e.g., the so-called *diseased* and *non-diseased* patients (Hajian-Tilaki, 2013). Although ROCs are very popular in the field of medical data analysis, they can be applied to any given dataset (for analysis of the binary classification task). In detail, ROC curves demonstrate the trade-off between the True Positive Rate (TPR, a.k.a. *sensitivity*) and False Positive Rate (FPR) for the given model using different probability thresholds (i.e. the "criterion" for positivity on the decision axis).

TPR is the probability of accurately predicting a positive outcome given that the observation is truly positive. Besides, another value that we require here is the True Negative Rate (TNR, a.k.a. *specificity*) which is the probability of predicting a negative outcome given that it is truly negative. To compute these values, we first define the subsets N_h (for each h) as the set of all held-out data points $\vec{y}^{(n)}$ which does contain the cause h (i.e., the cause h is labelled as being active for these data points). In more detail, considering $\vec{l}^{(n)}$ to denote the label of data point $\vec{y}^{(n)}$, we let:

$$N_h := \{\vec{y}^{(n)} \,|\, l_h^{(n)} = 1\}, \quad \forall h. \tag{5.2}$$

Then, the TPR and TNR of the cause h can be defined as follows:

$$TPR_h := \frac{1}{|N_h|} \sum_{n \in N_h} p(s_h = 1 \,|\, \vec{y}^{(n)}, \Theta) \tag{5.3}$$

$$TNR_h := \frac{1}{N - |N_h|} \sum_{n \notin N_h} p(s_h = 0 \,|\, \vec{y}^{(n)}, \Theta) \tag{5.4}$$

where $|\cdot|$ denotes the number of elements contained in each set. Moreover, observe that the above posteriors can be easily computed given the generative models presented in Chapter 3. Further, the above formulas are specific to probabilistic models and other binary classifiers (which could exploit a deterministic approach) may use another method to compute the TP and TN rates. The outcome of the above formulas is any value in the interval $[0, 1]$ which refers to the TPR or TNR; equivalently, we can also use percentages (as it is common in many studies) to refer to TPR and TNR (e.g., 0.2 or 20%). Subsequently, the FPR, which is the probability of predicting a positive outcome given that the observation is in fact negative, can be computed by:

$$FPR_h = 1 - TNR_h. \tag{5.5}$$

In addition, it can be easily seen that:

$$
\begin{aligned}
FPR_h &= 1 - \frac{1}{N - |N_h|} \sum_{n \notin N_h} p(s_h = 0 \,|\, \vec{y}^{(n)}, \Theta) \\
&= \frac{1}{N - |N_h|} \big((N - |N_h|) - \sum_{n \notin N_h} p(s_h = 0 \,|\, \vec{y}^{(n)}, \Theta) \big) \\
&= \frac{1}{N - |N_h|} \big(\sum_{n \notin N_h} 1 - \sum_{n \notin N_h} p(s_h = 0 \,|\, \vec{y}^{(n)}, \Theta) \big) \\
&= \frac{1}{N - |N_h|} \sum_{n \notin N_h} \big(1 - p(s_h = 0 \,|\, \vec{y}^{(n)}, \Theta) \big) \\
&= \frac{1}{N - |N_h|} \sum_{n \notin N_h} p(s_h = 1 \,|\, \vec{y}^{(n)}, \Theta).
\end{aligned} \tag{5.6}
$$

In order to produce the ROC curves, we consider number of $K \in \mathbb{N}$ thresholds $\lambda_k \in [0,1]$ such that $0 = \lambda_0 < \lambda_1 < \ldots < \lambda_K = 1$ (we used $K = 100$ for the ROCs in Figure 5.7). Later, we take the assumption that the posterior values in (5.3)-(5.4) are greater than zero (> 0) for each n and h, and then for each k, we let:

$$
TPR_h^k = \frac{1}{|N_h|} \sum_{n \in N_h} \delta\big(p(s_h = 1 \,|\, \vec{y}^{(n)}, \Theta) > \lambda_k \big) \tag{5.7}
$$

$$
TNR_h^k = \frac{1}{N - |N_h|} \sum_{n \notin N_h} \delta\big(p(s_h = 0 \,|\, \vec{y}^{(n)}, \Theta) > 1 - \lambda_k \big) \tag{5.8}
$$

where δ denotes the delta kronecker. Furthermore, we have:

$$
FPR_h^k = 1 - TNR_h^k. \tag{5.9}
$$

Observe that the above equations compute the rates of TP and FP at each given threshold λ_k. That is, for instance, we divide the number of diagnosed cases (the cases where the model assigns a higher probability of being positive than the threshold) by the total number of positives, i.e., $\frac{TP}{TP+FN}$. For $\lambda_0 = 0$, we classify all the patients as positively diseased and thus get $TPR_h^0 = 1$ and $TNR_h^0 = 0$ (consequently $FPR_h^0 = 1$); and for $\lambda_K = 1$, we classify all the patients as non-diseased and thus obtain $TPR_h^K = 0$ and $TNR_h^K = 1$ (and $FPR_h^K = 0$). For other thresholds, we obtain any value in between and a better model would be the one which produces higher TPR values and lower FPRs (the ROC curve is closer to the top left corner of the figure). As a consequence, we can consider the area under an ROC curve (AUC) as a measure of efficacy for the given model. In this respect, the best value to achieve is 1 and, as it is shown in Figure 5.7, the value of 0.5 corresponds to tossing a coin (50% chance for each patient to be diseased or non-diseased). So, the higher the AUC value is (closer to 1), the better the performance of the model will be.

Now, in order to closely examine the prediction capabilities of the Beta-MCA model, we again consider the simulated CAFPAs and present the ROC curves corresponding to the two diseases of high-frequency and broadband hearing losses in Figure 5.8. Here, besides the two models of Beta-MCA and noisy-OR, we also apply a Gaussian-MCA with a global variance as presented in Examples 1 and 2 and compare the inferences obtained by these three models. As it can be seen from Figure 5.8, Beta-MCA outperforms the other two models as its ROC curve is closer to the top left corner of the figure (consequently, the model achieves higher AUCs); but also noisy-OR and Gaussian-MCA illustrate a good performance. Let us now analyze these ROC curves as, especially for medical data, seemingly small differences of ROC

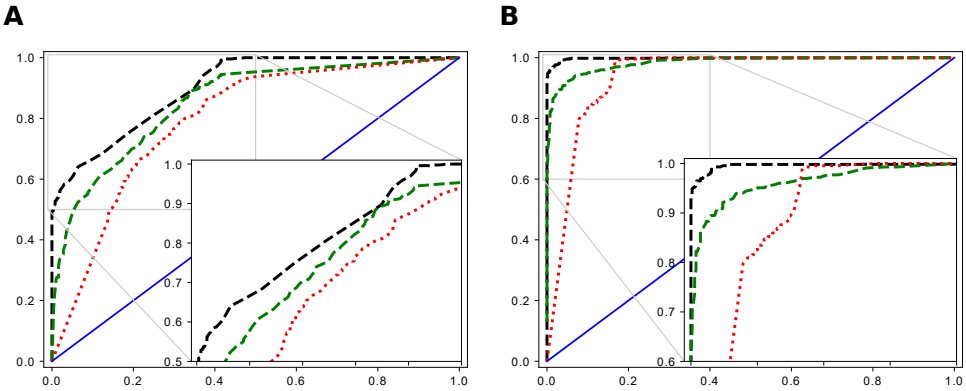


FIGURE 5.8: Illustration of the ROC curves presenting the results of Beta-MCA (black dashed lines), noisy-OR (green dashed lines) and Gaussian-MCA (red dotted lines) trained on simulated CAFPAs. Moreover, the blue line denotes the prediction results corresponding to tossing a coin (each disease has a 50% chance to be active). Here, (**A**) corresponds to broadband hearing loss and (**B**) to high-frequency hearing loss.

curves can be very important in practice.

To this, we consider two different instances as following: First is the diagnostics of the patients with broadband hearing loss where a sensitivity of 60% (i.e., TPR = 0.6) is assumed; and the second is analyzing the results for the high-frequency hearing loss with the sensitivity of 99% (i.e., TPR = 0.99). For the simulated CAFPAs, we recall that the number of patients with broadband hearing loss is 436 (the total number of positive cases) and the number of patients with high-frequency hearing loss is 510. Having these values for the two diseases, we can further compute Positive Predictive Value (PPV) and Negative Predictive Value (NPV) of the aforementioned models as follows:

$$PPV = \frac{TP}{TP + FP} \quad \text{and} \quad NPV = \frac{TN}{FN + TN}. \tag{5.10}$$

Here, FN denotes the false negative value which is the probability of predicting a negative outcome given that the observation is in fact positive.
The corresponding FPR obtained by the Beta-MCA model for broadband hearing loss with sensitivity of 60% is 0.05 which results in the specificity of 95%. Consequently, the PPV and NPV of the Beta-MCA model are computed as 90% and 75%, respectively. These values can be better described given a *contingency table* (Pearson, 1904) (a.k.a. *confusion matrix*) as it is illustrated in Figure 5.9. In addition, Table 5.1 represents a comparison of the obtained PPVs and NPVs for these three models given the first case. As observed, Beta-MCA achieves better PPV and NPV in comparison to the other two models which have also been reflected in the corresponding ROC curve.

Next, for the second case, we consider high-frequency hearing loss with the sensitivity of 99%. Then, the corresponding contingency table of the Beta-MCA model is presented in Figure 5.10. In addition, Table 5.2 provides a comparison for the obtained PPVs and NPVs of the three models. As illustrated, Beta-MCA achieves a better PPV, but all NPVs of the three models are the same. Moreover, Gaussian-MCA is observed to perform better (in this case) rather than noisy-OR as it achieves a higher PPV. Note that this result is interesting in a sense because noisy-OR has (in general) obtained better inference results compared to Gaussian-MCA (the corresponding AUC value of the noisy-OR model is higher for the high-frequency hearing loss). But at this specific sensitivity, the ROC curve of the Gaussian-MCA

		Patients with broadband hearing loss		
		Condition positive	**Condition negative**	
The Bayes net prediction results	**B-MCA outcome positive**	**True Positive** TP = 436 × 60% = 262	**False Posiitive** FP = 564 × 5% = 28	**Positive Predictive Value (PPV)** PPV = TP / (TP + FP) = 262 / (262 + 28) = 90%
	B-MCA outcome negative	**False Negative** FN = 436 × 40% = 174	**True Negative** TN = 564 × 95% = 536	**Negative Predictive Value (NPV)** NPV = TN / (FN + TN) = 536 / 710 = 75%
		Sensitivity TP / (TP + FN) ≈ 60%	**Specificity** TN / (TN + FP) ≈ 95%	

FIGURE 5.9: Contingency table of the Beta-MCA model for broadband hearing loss with sensitivity of 60%.

TABLE 5.1: Comparison of the obtained PPVs and NPVs. The results correspond to the inferences of the three mentioned models for the broadband hearing loss with sensitivity of 60% (the models have been applied on simulated CAFPAs; see text for details). As it can be seen, Beta-MCA outperforms the other two models and also, noisy-OR is the second best model.

model	PPV	NPV
Beta-MCA	90%	75%
noisy-OR	79%	74%
Gaussian-MCA	72%	73%

TABLE 5.2: Comparison of the obtained PPVs and NPVs. The results correspond to the inferences of the three mentioned models for the high-frequency hearing loss with sensitivity of 99% (the models have been applied on simulated CAFPAs; see text for details). As it can be seen, Beta-MCA is better than the other two models in terms of PPV and also Gaussian-MCA is performing better than noisy-OR. Nonetheless, all the three models obtained the same NPVs.

model	PPV	NPV
Beta-MCA	97%	99%
noisy-OR	80%	99%
Gaussian-MCA	86%	99%

is higher than the noisy-OR which explains the difference between the computed PPVs. This ascertains the importance of such analysis and further reveals the capacity of the ROCs in providing an accurate comparison for the inferences obtained by different models. Especially, for medical diagnosis, a small fraction of improvement in the ROCs is of great interest. For detections of COVID-19 infections, for e.g., it is very important not to miss an infection. Therefore, a classifier would have to operate at a high sensitivity (TPR) ideally at 0.99 or even higher. So, if the disease of Figure 5.8-B was COVID-19 instead of high-frequency hearing loss, then we could operate a Beta-MCA-based classifier above 99% TPR and it would produce (approximately) 3% false positives (wrongly positive COVID-19 diagnoses). The noisy-OR-based and Gaussian-MCA-based classifiers, on the other hand, would produce approximately 30% and 20% false positives respectively, i.e., many more patients would wrongly be diagnosed COVID-19 positive. To give you a better intuition, let us elaborate a little more in the following example.

		Patients with high-frequency hearing loss		
		Condition positive	**Condition negative**	
The Bayes net prediction results	**B-MCA outcome positive**	**True Positive** TP = 510 × 99% = 505	**False Posiitive** FP = 490 × 3% = 15	**Positive Predictive Value (PPV)** PPV = TP / (TP + FP) = 510 / (510 + 15) = 97%
	B-MCA outcome negative	**False Negative** FN = 510 × 1% = 5	**True Negative** TN = 490 × 97% = 475	**Negative Predictive Value (NPV)** NPV = TN / (FN + TN) = 475 / (5 + 475) = 99%
		Sensitivity TP / (TP + FN) ≈ 99%	**Specificity** TN / (TN + FP) ≈ 97%	

FIGURE 5.10: Contingency table of the Beta-MCA model for high-frequency hearing loss with sensitivity of 99%.

Example 3. *We again refer to the example of Coronavirus (SARS-CoV-2) stated in Chapter 1, and consider, for e.g., a population of* $10,000$ *where* 100 *of them are contracted to COVID-19. The task is to diagnose patients with the COVID-19 disease and isolate them for further investigations (or keep them in quarantine). We therefore aim at finding people with the highest probability of being diseased and at the same time reducing the number of people sent for further investigations as much as possible. In other words, having the highest TP and lowest FP values. Given the results of the ROC curves in Figure 5.8-B and the sensitivity of* 99%*, for instance, the Beta-MCA model would yield* $TP = 99$ *and* $FP = 297$*. This sums up the amount of* 396 *patients with the highest probability of being diseased (diagnosed with COVID-19). In addition, FPR values of the Gaussian-MCA and noisy-OR models in their ROC curves are obtained as* 0.17 *and* 0.26*, respectively. Therefore, Gaussian-MCA yields* $TP = 99$ *and* $FP = 1683$ *and noisy-OR yields* $TP = 99$ *and* $FP = 2574$*. This results in total numbers of* 1782 *and* 2673 *patients diagnosed by the Gaussian-MCA and noisy-OR models, respectively, which compares with the amount of* 396 *patients for the Beta-MCA model.*

5.2.2 Feature extraction – natural image patches

We trained the Gaussian- and Beta-MCA models on a set of $N = 100,000$ natural image patches chosen randomly from the van Hateren database (van Hateren and Schaaf, 1998). To apply the Gaussian-MCA model, we used ZCA-whitening as a preprocessing approach and further exploited the maximum magnitude superposition model presented in Section 3.4.2. For Beta-MCA, however, we trained the model on raw datasets and only rescaled the data into the interval $[0.1, 0.9]$. We also trained the Beta-MCA model with a background here. Except the two mentioned pre-processing steps, no extra pre- or post-processing approach is applied to improve the results.

The maximum superposition of the causes has been shown to be a suitable choice for training such natural images as stems and blades of grasses seen to occlude each other (Lücke and Sahani, 2008). Examples of generative models with a maximum superposition applied to a similar task are (Lücke and Sahani, 2008) which exploits a Poisson noise and (Bornschein, Henniges, and Lücke, 2013; Lücke and Eggert, 2010; Puertas, Bornschein, and Lücke, 2010) that are tailored to a Gaussian noise. For the Beta distribution, however, the following experiment can be seen, to our best knowledge, as the first to explore the performance of a LVM with a Beta noise. In addition, all the previous LVMs applied to natural images have learned a global variance σ^2 for their corresponding models. The two models used here, on the contrary, use two matrices: One to learn component means and another for component

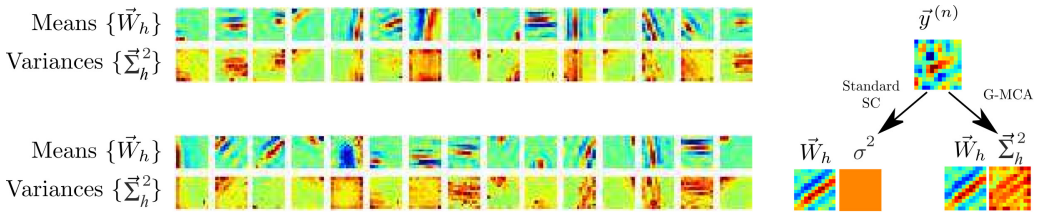


FIGURE 5.11: Dictionaries W_{dh} and Σ^2_{dh} (we here dropped the dependencies on Θ for the sake of brevity) learned by applying the Gaussian-MCA to natural image patches (30 generative fields each). We show Σ^2_{dh} instead of V_{dh} for better interpretability. Also, the right sketch illustrates the difference between using a standard SC model (scalar σ^2) and Gaussian-MCA. See Section B.1 for full dictionaries.

variances. The following experiments (with both Gaussian- and Beta-MCA models) can thus be seen, to our best knowledge, as the first LVMs that are applied with double-dictionaries to natural image patches (see Section 3.3 for details).

The Gaussian-MCA model

As the first application to natural data, let us use the model of Example 2, i.e., Gaussian-MCA. As discussed previously, the model has two matrices that we can optimize using Theorem 2: One for the mean W and a combination of V and W for the variance of Gaussian. To facilitate the interpretations, one can reparameterize the matrices back to the normal Gaussian parameters by setting $\Sigma^2_{dh}(\Theta) = V_{dh} - W^2_{dh}$ (see Section A.1 for further details).

To the set of patches with size $D = 12 \times 12$ we applied a Gaussian-MCA model with $H = 1,000$ components and learned individual dictionaries for component means (W_{dh}) and variances ($\Sigma^2_{dh}(\Theta)$). In addition, as the considered data encompasses both positive and negative values, we used a maximum magnitude function defined in (3.41). In detail, we let $h(d,\vec{s},\Theta) = \text{argmax}_h |W_{dh}s_h|$ (note that $M(\Theta) = W$ for the case of Gaussian). We initialized the parameters Θ randomly and applied 2000 variational EM iterations. After learning, we observed a large variety of generative fields (GFs) for component means (including the familiar Gabor-like and globular fields) as well as a large variety of GFs for component variances. Figure 5.11 then illustrates 30 examples of such GFs where the full dictionaries are provided in Section B.1. The observed variety of variance GFs stands in contrast to a uniform variance with equal value for all latents (a global variance σ^2) as assumed by standard SC or previous MCA versions. Compared to such previous approaches, a generalized encoding of component means alongside component variances may allow us to describe more accurately how first- and second-order statistics are combined in real world datasets and may be more closely aligned with neural responses in V1.

Such a double-dictionary approach raises very interesting questions that can be investigated in detail and opens the door to many possible future studies (see Section 3.3 for some discussion). If we identify image patches that maximally activate a given latent, we can systematically change these patches while measuring changes of the latent's responses. By adding specific types of changes including addition of specific noise, the response properties of the latent changes depending on how it encodes component variances. For instance, noise added in proximity of a Gabor component or distant from it has a different effect for latents with individual variances compared to latents with the same variance for all pixels and all components (as used for standard SCs). Such differences in responses between the models could be used to design stimuli that can detect potential variance encoding in V1 and other sensory areas.

The Beta-MCA model

Next, we exploited the Beta-MCA model and assessed its performance on raw natural image patches. We used $D = 8 \times 8$ image patches with pixel intensities linearly rescaled to fill the interval $[0.1, 0.9]$. After rescaling, we added a small amount of Beta noise to the image patches (which increased stability). On these data points, we trained the Beta-MCA model with $H = 100$ latent components for 500 variational EM iterations.

We were particularly interested in how learned data representations varied when transitioning from a model with a global scalar variance parameter σ^2 to a model with individual variances per latent component Σ^2_{dh} (note that we here dropped the dependency on parameters Θ for the sake of readability). Consequently, we trained two different models: One with a dictionary M for component means and a global variance σ^2 (see Section A.3 for details) and another with individual dictionaries for component means and component variances. The details of how to train these two Beta-MCA models with a matrix or a scalar variance and the corresponding updates are further presented in Section A.3.

To initialize parameters Θ of the two models, we applied the following procedure: Component means $\vec{M}_h^{\text{init}}$ were initialized with the mean of the data points plus a small amount of Gaussian noise, the prior components π_h^{init} and also the values of σ^{init} and Σ^{init} were uniformly randomly sampled from the interval $(0, 1)$. We further kept the variance parameters σ^2 and Σ^2 constant at their initial values during the first 30% of the iterations and only updated parameters M (using W and V) and $\vec{\pi}$: We found this to lead to a more stable convergence behaviour of the algorithm.

The results are then presented in Figure 5.12. Looking at the generative fields (GFs) corresponding to the component means $\vec{M}_h$ learned by the Beta-MCA model with scalar variance parameter, we observed many globular fields, some elongated fields and some gratings (Figure 5.12-A). For the variance, we inferred the single globular value of $\sigma = 0.086$. For the model with individual component variances, the diversity of GFs increased with many elongated fields (with different locations and orientations), gratings, and with large globular fields (Figure 5.12-B, top). A crucial qualitative difference of the model with individual variances is, of course, the additional GFs for the variances (Figure 5.12-B, bottom). The variances allow for increased flexibility in terms of intensity variations of elongated fields (for many elongated fields variances are high where the corresponding means are high). GFs for variances also allow for more intricate modelling of image structures, however. As it can be observed, for some fields (e.g., bottom right) variance encoding allows for shape variations rather than intensity variations: Variances are high only at the transition of high to low mean values, which means that constructing variations of the precise shape by a component is allowed and modelled.

In terms of free energy, the Beta-MCA model with individual component variances achieved a better fit to the data compared to the model with global variance parameter (Figure 5.12-E): The corresponding final free energy value was **82.03** for the model with individual variances compared to **70.31** achieved by the model with global variance parameter. While a higher free energy can be expected because of more parameters of the model with individual variances, the higher free energy can be taken as a confirmation that encoding a more intricate image structure in Figure 5.12-B corresponds to a more refined model of image patches structure.

To obtain more intuition about the learning results, we also employed the trained models to generate new data points which can then visually be compared to the image patches used for training. Consistent with the higher free energy value, we also observed that the data points generated using the model with individual component variances appear to be more similar to the image patches used for training (Figure 5.12-D, F and G).

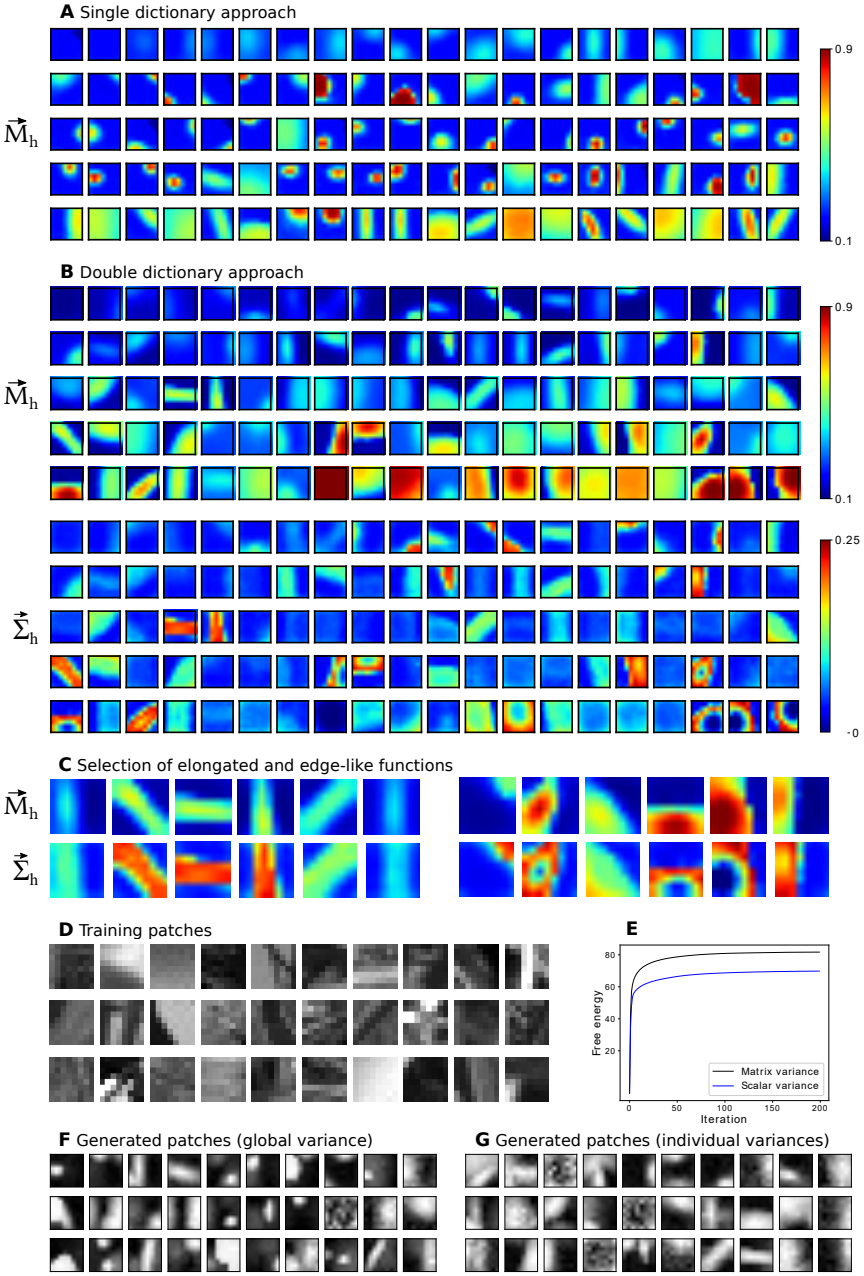


FIGURE 5.12: Feature extraction. **A**-**B** Parameters $\vec{M}_h$ and $\vec{\Sigma}_h$ of the Beta-MCA generative model trained with (A) a single-dictionary (a scalar variance σ^2) and (B) individual dictionaries for component means and component variances. Displayed are $H = 100$ GFs sorted from left to right and from top to down based on their prior values, π_h. The top left corresponds to the background which has a prior parameter close to 1. Each of the component variances in the lower part of panel B corresponds to the respective component means in the upper part of panel B. **C** Examples of (left) elongated and (right) edge-like functions learned with the double-dictionary approach. **D** 30 training patches. **E** Evolution of the free energy functions for both single- and double-dictionary approaches. Here, only the first 200 (out of 500) iterations are depicted. **F**-**G** 30 examples of the patches generated according to the Beta-MCA trained with (F) scalar variance and (G) matrix variance. We here deliberately reduced the amount of noise in the generating process for the sake of visualization. Also, in all the above cases, we linearly scaled the learned values to fill the corresponding color range in the gray (color) space.

Later, we considered two specific components from the learned GFs together with the background in order to closely assess the capacity of the two approaches used here in generating artificial data points. This time, we were especially interested in investigating how extra information obtained from the double-dictionary approach can be leveraged in this process. Therefore, we deliberately chose two components with approximately the same mean values but different variances (these components are chosen from GFs learned by the double-dictionary approach). We then superimposed these two components (plus the background) according to the proposed Beta-MCA model once with double-dictionaries (matrix variance) and another time with a single-dictionary (scalar variance). For the latter case, we used $\sigma = 0.086$ and considered the same value of variance for all different pixels. The results are then illustrated in Figure 5.13. As it can be seen, the double-dictionary approach distinguishes between the two components with different variance values and reflects such a difference in the generating process. In contrast, the generated data point obtained from the single-dictionary approach seems to be much smoother which explains the effect of a global variance value. For medical data, as described in Chapter 1, we demand the dominant disease (cause) to specify the statistics of an observable d (we argued that both mean and variance values should be determined by the dominant cause). Hence, the current experiment illustrates that a double-dictionary approach is a more suitable choice for such analysis as it can deliver the specific patterns of the strongest cause (both mean and variance values) in the generating process. The obtained extra information can consequently result in a better inference which is the main goal of our investigations.

Finally, to examine the implications of different settings, we repeated the experiment above several times using different choices of hyperparameters. For instance, we ran the two algorithms again with $H = 200$ components and in another run with $H = 100$ components and $D = 16 \times 16$ (the results corresponding to training with $H = 200$ components and $D = 8 \times 8$ is further presented in Section B.1). In all different cases, we observed similar results both qualitatively and quantitatively. In particular, the model with individual component variances achieved consistently the higher final free energies.

Consistent with higher free energy values, the current experiments illustrate that the variance components learned by the proposed Beta-MCA (with double-dictionaries) provide (in contrast to the Beta-MCA with scalar variance and other conventional LVMs) more accurate information on how first- and second-order statistics of the data are combined. Such generative models with non-linear combinations of latents are particularly well suited to model, for instance, occlusion effects in natural image patches (as has been argued previously, e.g., Lücke and Sahani, 2008; Bornschein, Henniges, and Lücke, 2013; Lücke and Eggert, 2010; Puertas, Bornschein, and Lücke, 2010). All of these previous works used generative models with either Gaussian or Poisson noise assumptions, however. Considering Figure 5.12, this study is, as mentioned before, the first to report results for a non-linear generative model with a Beta noise assumption and individual component variances. As one example of the general LVMs proposed here, the experiments specifically ascertain the effectiveness of the family of generative models and also the corresponding parameter optimization approaches.

5.2.3 Noise type estimation

As another application, we considered the problem of determining the unknown noise type of a given dataset. This is an important problem for data analysis in a number of applications and has been actively researched (see, e.g., Teymurazyan et al., 2013; Valera and Ghahramani, 2017; Vergari et al., 2019). In (Teymurazyan et al., 2013), for instance, the problem of distinguishing the type of noise in Positron Emission Tomography (PET) data is studied. In PET, radioactive fluids are injected into humans or animals in order to obtain images for diagnostics or scientific studies. As a result, knowing the type of noise is crucial for different

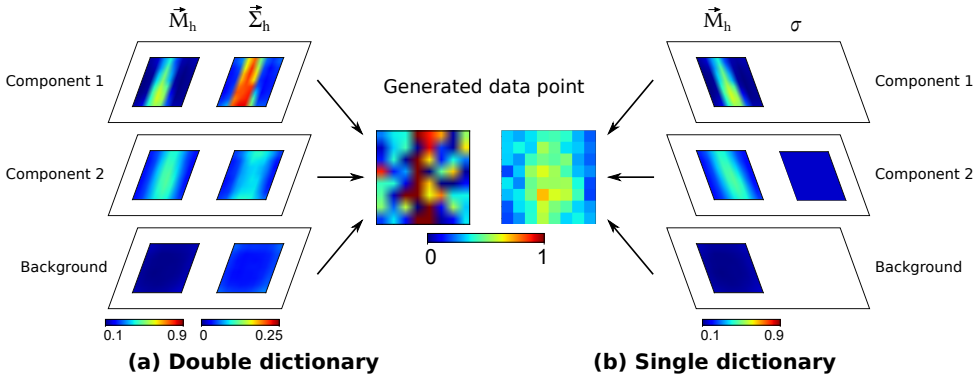


FIGURE 5.13: (**a**) Corresponds to the double-dictionary approach where component variances are available together with component means; and (**b**) corresponds to the single-dictionary approach as only a scalar variance is used. The two components are chosen such that their variances differ from each other (this is a common setting that exists in real datasets). The maximum superposition (3.38) is then used for the combination of the two causes together with the background. Consequently, we generated two data points according to the Beta-MCA model with either double-dictionaries or a single-dictionary which are shown in the middle of the figure. Visually speaking, the double-dictionary approach can reflect the high variance value in the generating process while the single-dictionary approach produces a somehow smooth data point. We here reduced the noise by 30% for the sake of a better visualization.

image processing routines (Teymurazyan et al., 2013; Mou, Huang, and O'Sullivan, 2018). Other related work, e.g. (Valera and Ghahramani, 2017; Vergari et al., 2019), focused on machine learning automation. In (Valera and Ghahramani, 2017), a Bayesian approach is presented to infer the statistical types of data as being categorical, ordinal, count, real-valued, positive real-valued or interval. The algorithm does not directly suggest a noise distribution for the data, however. For instance, it is still necessary to find out the best probability density function (e.g. Gamma, Inverse Gamma, Inverse Gaussian or Exponential distribution) that fits positive real-valued data. Vergari et al. in (Vergari et al., 2019) have further attempted to improve these results and aimed at inferring the parametric likelihood model of the data. They have then proposed an approach which is shown to be more robust in working with missing, corrupted or anomalous data and can successfully estimate the noise type of many different datasets.

In this study, the general parameter optimization method presented here allows for developing a direct approach to determine the type of data distribution. As a proof of concept in this direction, we show how, e.g., Gaussian- and Gamma-MCA models can be used to determine which of the two noise distributions is better suited for a given dataset. To this, we consider two different datasets: Visual and acoustic data and present the results in the following.

Before doing so, we recall that detailed information regarding the variance update of Gaussian-MCA is presented in Examples 1 and 2 and we further discuss it in Section A.1. Also the update equation for the variance of Gamma-MCA is presented in Section A.2 (note that similar to the Gaussian-MCA model, we use a combination of W and V to estimate the variance of the Gamma distribution). Moreover, we used the Gamma-MCA+ approach in order to obtain reliable results. We also applied the two generative models with $H = 512$ components for an amount of 1000 variational EM iterations.

TABLE 5.3: Results of the noise type estimation experiment with noisy 'house' images (see text for details). Listed are the highest free energies per data point for five different runs.

model	Gaussian-MCA	Gamma-MCA
Gaussian-noise	-741.18	-755.12
Gamma-noise	-742.90	-737.00

Visual data

For the first experiment, we considered the standard 'house' image (Figure 5.14), to which we either added Gaussian or Gamma noise. We then applied the Gaussian- and Gamma-MCA models and compared their variational lower bounds (as the approximations of their log-likelihoods) on different datasets. We used a common scenario for both noise models and added Gamma or Gaussian noises with random variance values. We then segmented the corrupted image into the patches with the size $D = 12 \times 12$ and applied each of the Gaussian- and Gamma-MCA models on these two corrupted images (one with Gamma noise and another with Gaussian noise). The obtained free energies are then presented in Table 5.3.

From Table 5.3, we observed that the lower bound of the Gaussian-MCA is higher for the data with Gaussian noise than the lower bound of the Gamma-MCA and vice versa. Although this may be seen as a proof-of-concept example, the experiment reveals that the free energy comparison can be exploited in a straightforward manner to distinguish between the two noise types given a dataset. Here, since we applied models with the same number of model parameters, we directly used the lower bounds (as approximate to log-likelihoods) for model selection, i.e., we did not have to consider penalty terms for the number of parameters (compare, e.g., AIC or BIC criteria MacKay, 2003).

Acoustic data

Next, we used data with a natural noise source without artificially added noise. That is we considered audio examples from the CHiME dataset (Foster et al., 2015) and fitted Gaussian- and Gamma-MCA models to amplitude spectrograms. In detail, we used the following three sound files from the CHiME dataset: *CR-lounge-200110-1711.s0-chunk46.48kHz.wav*, *CR-lounge-200110-1601.s0-chunk25.48kHz.wav* and *CR-lounge-200110-1601.s0-chunk7.48kHz.wav*. We have carefully listened to different examples from the dataset and have chosen the ones that seemed to be noisier than the others. Also amplitude spectrograms were computed as follows: Time domain signals were resampled to 22.050 Hz and cut into 2 seconds long mono segments. We then computed the STFT using a 2048-point FFT, 512 samples frame shift and Hann windowing. This resulted in spectrograms with 1025 frequency channels and 87 time steps. The amplitude spectrograms were then cut into the patches of size $D = 12 \times 12$ resulting in $N = 77,064$ data points on which Gaussian- and Gamma-MCA were trained. The obtained free energy values are then presented in Table 5.4.

For both Gaussian- and Gamma-MCA models, very small values of components Σ were learned (between 0.1 and 0.2) which resulted in very high, positive free energies. Besides, we observed that the free energy values of the Gaussian-MCA varied much stronger between different sound examples compared to the free energies of the Gamma-MCA. This might be attributed to different sound characteristics of the considered examples. Nonetheless, the consistently higher free energies obtained by Gamma-MCA suggest that the Gamma distribution is a better noise model for the considered data rather than the Gaussian.

The two examples here show that the updates (4.27)-(4.28) and (4.33) (together with the relations presented in Sections A.1 and A.2) can supply sufficiently flexible and precise algorithms for noise estimation also at large-scales. Model selection using further types of

TABLE 5.4: Results of the noise type estimation experiment with natural noise sources (see text for details). Listed are the highest free energies per data point for three different runs.

model	Gaussian-MCA	Gamma-MCA
example 1	215.20	298.53
example 2	95.61	250.77
example 3	47.36	206.14

noise distributions can proceed along the same line but requires a more elaborate treatment (also compare, e.g., Valera and Ghahramani, 2017) which exceeds the purposes of this study.

5.2.4 Denoising

Finally, we used the presented updates to denoise images corrupted by non-Gaussian noises. This task can be considered more difficult compared to the removal of additive white Gaussian noise which numerous established denoising algorithms are optimized for (compare, e.g., Dabov et al., 2007; Gu et al., 2014; Burger, Schuler, and Harmeling, 2012; Chaudhury and Roy, 2017; Chen and Pock, 2017; Zhang et al., 2017; Tai et al., 2017; Zhang, Zuo, and Zhang, 2018). Although such established algorithms can still be applied for denoising an image corrupted by other noises rather than Gaussian, we here showed that a proper model which is tailored to the specific noise of the data can yield better results for this task. We subsequently investigated three different noise types, namely Poisson, Exponential and Beta. As datasets, we used the standard benchmark images 'house', 'cameraman' and 'peppers' (Dabov et al., 2007; Salmon et al., 2014). From each image we then generated one with added Poisson noise, one with added Exponential noise and one with added Beta noise. Next, we applied Poisson-MCA (P-MCA), Exponential-MCA (E-MCA) and Beta-MCA (B-MCA) models to denoise the images corrupted with the corresponding noises. In particular, we considered the non-linear inverse problem (compare Tarantola, 2005) of estimating the non-noisy image only from observing a noisy version of the clean image: P-MCA, E-MCA and B-MCA are directly applied to the noisy image itself without leveraging external, clean training data (this is sometimes also referred to as *zero-shot* learning; compare, e.g., Shocher, Cohen, and Irani, 2018; Imamura, Itasaka, and Okuda, 2019). Mathematically speaking, given a corrupted image $\vec{y}_{\text{corrupted}}$ (note that it is common to represent such an image as a column-stacked vector), we aimed at computing an estimation $\vec{y}_{\text{estimate}}$ of the clean image $\vec{y}_{\text{original}}$. In general we let $\vec{y}_{\text{corrupted}} = F(\vec{y}_{\text{original}})$, where F denotes a noise operator (in this case Poisson, Exponential or Beta), and then computed the estimator $\vec{y}_{\text{estimate}}$ as follows (see, e.g., Salmon et al., 2014; Guiraud, Drefs, and Lücke, 2018 for some details):

$$\left(\vec{y}_{\text{estimate}}\right)_d = \langle \bar{M}_d(\vec{s},\Theta)\rangle_q \quad \text{where} \quad \bar{M}_d(\vec{s},\Theta) = M_{dh(\vec{s},d,\Theta)}(\Theta). \tag{5.11}$$

In addition, note that for Poisson and Exponential distributions the sufficient statistics are $T(y) = y$, i.e., we have $M(\Theta) = W$ and therefore:

$$\left(\vec{y}_{\text{estimate}}\right)_d = \langle \bar{W}_d(\vec{s},\Theta)\rangle_q.$$

We used images with gray scales in the interval $[0, 255]$, and further, in order to rescale images to smaller intervals (e.g., for Poisson or Beta noise), we divided the data by its maximum and then multiplied the results by the desired peak value. For all the upcoming experiments (unless stated otherwise), the noisy image is segmented into patches of size $D = 12 \times 12$ on which we applied the corresponding ef-MCA model without any pre- or post-processing and performed 1000 variational EM iterations with $H = 512$ components.

We also applied the same variational approximation as the previous experiments in order to scale the EM algorithm. In addition, for all the three P-MCA, E-MCA and B-MCA algorithms, we assumed a background model that can avoid degeneracies in the training.

Note that the performance of any denoising algorithm will depend on many different aspects that are not the focus of this study. This includes the used approximate inference approach for the E-step, the averaging and inpainting algorithms which are used by the trained data model, and methods to avoid local optima in training a dictionary. We did not use annealing to avoid local optima, and also did not optimize approximate inference or the averaging algorithm to improve the denoising performances. In general, such optimization techniques could be further performed, but this exceeds the purposes of the current study.

Nonetheless, we can in general conclude that the upcoming results confirm both scalability and effectiveness of the three models used here. Moreover, the family of ef-MCA models proposed here allows for treating a much larger variety of noise distributions, of course, but focusing on P-MCA, E-MCA and B-MCA may illustrate the potential also for other noise types which may be encountered in a dataset.

Poisson noise

After Gaussian denoising, Poisson noise removal benchmarks do presumably provide the most extensive opportunities for quantitative comparisons of different denoising algorithms. As mentioned before, one possibility for denoising an image corrupted with Poisson noise is to simply ignore the specific properties of the Poisson noise and employ a method that is tailored to the Gaussian. For instance, the sparse 3D transform-domain collaborative filtering (BM3D) (Dabov et al., 2007) can be applied directly. BM3D is a well-known denoising method which often outperforms sparse coding or k-SVD (Aharon, Elad, and Bruckstein, 2006) algorithms on image denoising benchmarks. The method is tailored to 2D-images corrupted by Gaussian noise but, as mentioned, can also be applied to Poisson noise removal (and also others) to certain extents.

A frequently followed alternative approach is to first convert the Poisson noise into Gaussian noise, and then apply an algorithm which assumes Gaussian observables. Such methods apply a non-linear Variance Stabilization Transform (VST) such as Anscombe (Anscombe, 1948) and Fisz (Fisz, 1955) that produces a signal in which the noise can approximately be treated as additive Gaussian with unit variance. Approaches such as VST+BM3D use the Anscombe root transformation followed by standard BM3D in this case (see, e.g., Makitalo and Foi, 2010; we also elaborate further in Section B.2). Such transformation-based approaches are specific to the Poisson noise as similar transformations are not in general available for exponential family distributions. But also in the case of Poisson noise, transformation to Gaussian noise can be problematic. In fact, it is known that transformation-based methods can yield poor results for low-intensity signals (see Salmon et al., 2014; Makitalo and Foi, 2010). It should be noted that the Poisson noise is not additive (the same is for Exponential and Beta) and the image intensity (or the peak value of the original image) determines the strength of the noise. For Poisson, the lower the peak is, the stronger the noise would be. For low peak values (e.g. lower than 3), VST transformations are shown to be less effective. As a result, many approaches have been suggested to improve such transformations or exploit alternative approaches that yield better results in the case of low-intensity signals (Salmon et al., 2014; Makitalo and Foi, 2010; Rond, Giryes, and Elad, 2016; Azzari and Foi, 2016; Giryes and Elad, 2014; Niknejad and Figueiredo, 2018). One example is the P^{4}IP approach (Rond, Giryes, and Elad, 2016) that applies a general plug-and-play-prior approach on Poisson inverse-problems. The method still relies on Gaussian-based denoising algorithms, however. Other studies (e.g. Makitalo and Foi, 2010; Azzari and Foi, 2016) attempt to improve the transformation techniques and therefore enhance the performance

TABLE 5.5: Comparison of the PSNR values (in terms of dB) for the considered Poisson denoising benchmarks. Values of BM3D, VST+BM3D, I+VST+BM3D, P^{4}IP, DenoiseNet, NLSPCA and SPDA are taken from (Azzari and Foi, 2016; Makitalo and Foi, 2010; Remez et al., 2017). The considered algorithms are divided into two groups: BM3D, VST+BM3D, I+VST+BM3D and P^{4}IP are highly engineered algorithms that assume a Gaussian noise model. Also DenoiseNet is a deep learning-based algorithm that uses a large number of training data. NLSPCA, SPDA and P-MCA, on the other hand, are based on the assumption of a Poisson noise model (see text for more information). The bold number in each subgroup denotes the best PSNR value in comparison to the other models of the subgroup; the bold and underlined number denotes the best PSNR value amongst all different models. As illustrated, P-MCA represents a competitive performance compared to BM3D, VST+BM3D, I+VST+BM3D and P^{4}IP, and outperforms NLSPCA and SPDA in 5 out of 6 cases that are considered here. *This is the value reported by the P^{4}IP authors in their arXiv paper (Rond, Giryes, and Elad, 2016), and this value is also used by other later contributions for comparisons (e.g. Azzari and Foi, 2016; Remez et al., 2017); but we remark that another value (22.33, i.e., the second digit is different) is reported in the journal version.

Method	Peak	House	Cameraman	Peppers
BM3D	2	24.18	22.13	21.97
VST+BM3D		23.79	21.97	22.02
I+VST+BM3D		24.62	22.25	21.93
P^{4}IP		24.65	21.87	21.33*
DenoiseNet		**24.77**	**23.25**	**23.19**
NLSPCA		23.16	20.64	20.48
SPDA		**24.37**	21.35	21.18
P-MCA		23.51	**21.79**	**22.03**
BM3D	4	26.04	23.94	24.07
VST+BM3D		25.49	23.82	24.01
I+VST+BM3D		26.07	24.10	24.04
P^{4}IP		26.33	23.29	23.88
DenoiseNet		**26.59**	**24.87**	**24.83**
NLSPCA		24.26	20.97	21.07
SPDA		25.30	21.72	22.20
P-MCA		**26.14**	**23.42**	**23.87**

of the methods. For instance, an iterative VST framework (known as I+VST+BM3D) is established by Azzari and Foi (Azzari and Foi, 2016) that can cope with extreme low signal-to-noise ratio (SNR) cases.

More relevant for comparison to our approach are methods that explicitly assume a Poisson noise. One important study is the work by Salmon et al. (Salmon et al., 2014) that uses Poisson PCA in which the link between latents and observables (as discussed in Chapter 3) is considered to be a weighted linear sum that sets the Poisson's natural parameter. Their approach is named Non-Local Sparse Principal Component Analysis (NLSPCA). In another study, Giryes and Elad (Giryes and Elad, 2014) introduce a Poisson SC model (Sparse Poisson Denoising Algorithm (SPDA)) capable of learning dictionaries from a Poisson distributed dataset. Their method likewise considers a linear superposition of the latents to set the Poisson's natural parameter. Despite these two approaches, the used P-MCA model exploits a non-linear (maximum) superposition of the latents that sets the mean of observables. Moreover, this model is only one example of the general results represented by Theorems 2, 3 and 4.

In addition to the approaches presented above, deep learning-based techniques have also been applied to Poisson denoising in recent years (see, e.g., Remez et al., 2017; Kumwilaisak et al., 2020; DeGuchy et al., 2019; Piriyatharawet, Kumwilaisak, and Lasang, 2018; Jin et al., 2018). Importantly, Remez et al. (Remez et al., 2017, DenoiseNet) explore deep convolutional neural networks (CNNs) to denoise low-light images, and the proposed DenoiseNet has shown to establish state-of-the-art results for Poisson denoising. Nonetheless, there are substantial differences between, for instance, DenoiseNet and P-MCA: P-MCA is trained directly on the noisy image and the approach does not require a-priori information about the peak value of the corrupted data; this contrasts with DenoiseNet which requires external training data and which is optimized for a specific peak value. In another recently published work (Kumwilaisak et al., 2020), authors use multi-directional long-short term memory (LSTM) networks along with the CNNs that can also capture and learn the statistics of residual noise components. The method can further achieve better results in comparison to the DenoiseNet. Describing all the details of the aforementioned studies is, however, far from our goals here and we refer the readers to the corresponding papers for further details.

Table 5.5 presents a quantitative performance comparison of the different approaches using the standard measure of peak-signal-to-noise-ratio (PSNRs). The methods we chose for comparison in Table 5.5 represent, to our best knowledge, state-of-the-art results for Poisson denoising given different peak values. As it can be observed, P-MCA produces competitive results but can be outperformed by BM3D, VST+BM3D, I+VST+BM3D, P^4IP or DenoiseNet. At a closer inspection, these five approaches are all using considerable fine-tuning specific to images. Also the first four algorithms are based on a Gaussian assumption (potentially after extensive pre-processing steps). P-MCA is like SPDA and NLSPCA not tailored to images. All these three approaches also have in common that they are directly based on the assumption of a Poisson distribution for the observables. While both NLSPCA and SPDA assume a linear superposition link to the natural parameters, P-MCA assumes a maximum superposition for the Poisson mean. Considering Table 5.5, P-MCA shows in comparison to SPDA and NLSPCA improved performance in terms of PSNR values in five out of the six investigated settings. Such improvements for one example of the exponential family distributions (P-MCA) may argue in favor of the general approach presented here as the proposed family of generative models and their parameter optimization based on Theorems 2 and 3 (and also Theorem 4 for the general case). Furthermore, even compared to approaches such as BM3D, VST+BM3D, I+VST+BM3D, P^4IP or DenoiseNet, which are image specific and highly optimized for denoising tasks, P-MCA shows comparable results. Figure 5.14 further illustrates the reconstructed house, cameraman and peppers images using the P-MCA model. The estimated house image can be further compared with Figure 9 of (Giryes and Elad, 2014) which presents the results of BM3D, SPDA and NLSPCA methods.

Besides, it should be mentioned that the denoising results presented here for the P-MCA model correspond to one run for each of the peak values while the results of other approaches are (mainly) the averaged values over five different noise realizations. We further present a more elaborate and somehow complete comparison in Section B.2.

Exponential noise

For the Exponential noise (E-MCA), a similar comparison as for the Poisson noise is not possible because, to the best of our knowledge, no other latent variable model is tailored to the removal of Exponential noise. Consequently, E-MCA illustrates an example where the general results of Theorems 2 and 3 give rise to an algorithm directly applicable to less commonly encountered noise types (such as Exponential). Figure 5.15 then illustrates the results for denoising the house image when this time is corrupted by Exponential noise (we used the original house image in the interval $[0, 255]$). We here compared the performance of E-MCA

(a) Original image (b) Corrupted image (c) Denoised with P-MCA

FIGURE 5.14: Denoising the house, cameraman and peppers images when corrupted by Poisson noise (the peak values of the house and peppers images are 2 and of the cameraman is 4). The figure illustrates (a) original and (b) noisy images together with (c) reconstructed images using P-MCA. The corresponding PSNR values are further presented in Table 5.5. The denoised house image can be compared with Figure 9 of (Giryes and Elad, 2014) which depicts the reconstructed image using BM3D, SPDA and NLSPCA methods.

with other models: P-MCA and G-MCA (Gaussian-MCA; with a global variance parameter σ^2, i.e. a single-dictionary approach) as other ef-MCA models presented here, and BM3D, VST+BM3D and spike-and-slab sparse coding (SSSC) (Sheikh, Shelton, and Lücke, 2014; Goodfellow, Courville, and Bengio, 2012b). The SSSC is a sparse coding approach with a flexible prior distribution that assumes a Gaussian noise model (we used the same settings for training the SSSC model as the E-MCA and other ef-MCA models). As it can be observed from Figure 5.15, E-MCA achieves the highest PSNR value compared to all other approaches. This result might be considered intuitive as E-MCA is based on the Exponential distribution and is thereby most closely aligned with the Exponential noise of the image. In addition, P- and G-MCA models perform reasonably well in comparison to SSSC; and VST+BM3D is shown to perform the worst, which confirms the fact that the VST transformation is specific to Poisson noise.

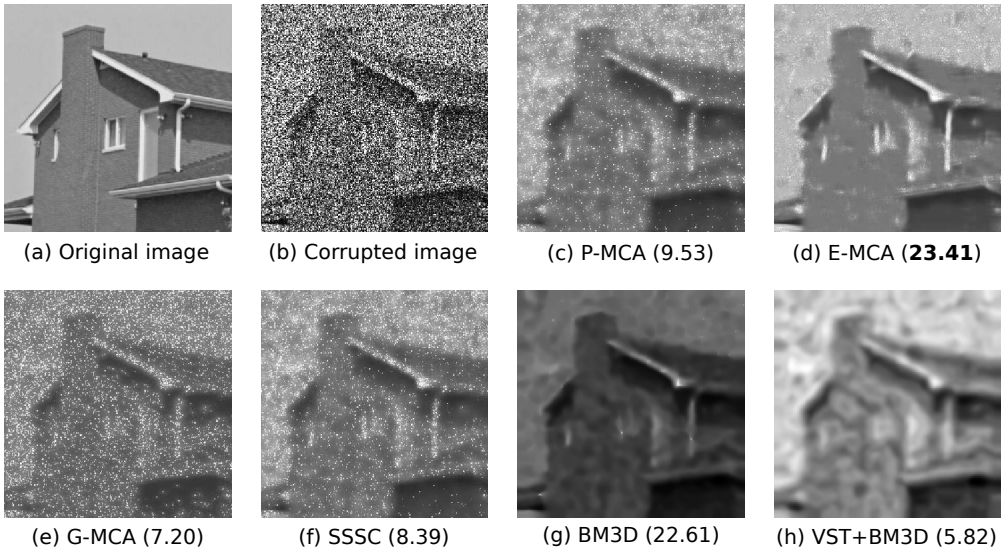


FIGURE 5.15: Denoising the house image when heavily corrupted by Exponential noise (peak = 255). Depicted are (a) original, (b) corrupted, and (c)-(h) reconstructed images obtained by using different models and the corresponding PSNR values in dB (see text for details). Here, E-MCA achieves the highest PSNR value in comparison to the others which is consistent with the Exponential noise. Also note that P- and E-MCA use one-parameter distributions of the exponential family and hence, only use the information of one dictionary (related to the mean of the distribution); whilst the other models (G-MCA, SSSC) use extra parameters to additionally model the second-order statistics. Also, BM3D requires the variance value for its algorithm.

Visual inspection may even argue in favor of using the right noise model more than what PSNR values would suggest. Figure 5.15, for instance, represents larger differences between E-MCA and BM3D reconstructed images compared to what may be expected by considering the PSNR values of the two approaches. Consistent with higher PSNRs, we observed that, e.g., P-MCA better recovers the structure of the original image when the noise is Poisson and vice versa for E-MCA. This, however expected, further highlights the importance of tasks such as noise type estimation and model selection. Further, the results emphasize the utility and practical applicability of the presented parameter optimization approach for the non-linear generative models.

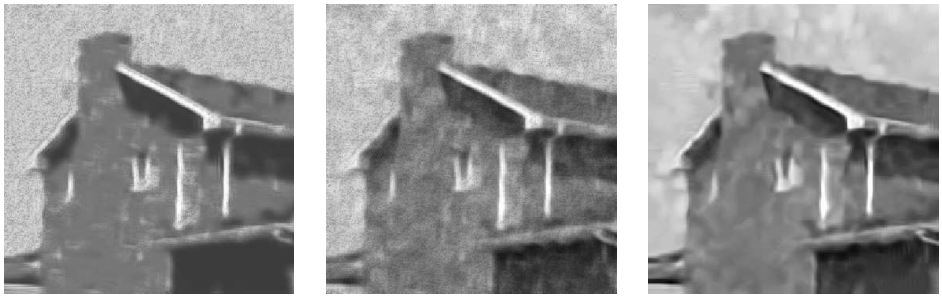
Beta noise

Finally, we study a concrete example of denoising an image corrupted by Beta noise. The Beta distribution can be considered the most difficult case and, to our best knowledge, there has been no similar experiment with Beta noise that we could consider for our comparisons. In previous sections, we examined the capabilities of the Beta-MCA (B-MCA) model in training medical data (as presented in Section 5.2.1) as well as extracting statistical features from natural image patches (as presented in Section 5.2.2). In both cases, the B-MCA (together with other ef-MCA models considered here) produced high-quality results and reliable inferences. Now, we investigate the denoising experiment which likewise requires training at large-scales.

As an example, we used the house image which we rescaled from $[0, 255]$ to $[0.2, 0.8]$. The image was then corrupted by Beta noise with a standard deviation of 0.3. Importantly, for this task, we considered a B-MCA model with a global variance parameter (in contrast to the double-dictionary). This is common as a homoscedastic noise is mostly considered

for the denoising experiment. Details of the variance update equation are then presented in Section A.3.1.

(a) Original image (b) Corrupted (8.53/76.69) (c) B-MCA (**26.45/9.73**)



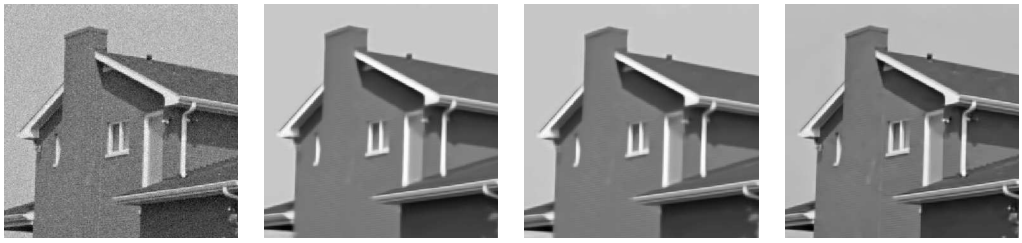
(d) MCA (25.61/10.73) (e) SSSC (24.96/11.55) (f) BM3D (26.21/11.55)

FIGURE 5.16: Denoising the house image corrupted by Beta noise with $\sigma = 0.3$. Depicted are (a) original, (b) corrupted with added Beta-noise, and (c)-(f) reconstructed images obtained by applying four different models and also their corresponding PSNR (in dB) and MSE values (see text for details). The first number in the parentheses corresponds to the PSNR value and the second to MSE. As it can be seen, the B-MCA model outperforms the other three models which is consistent with the Beta noise.

In addition to the B-MCA model, we further applied (Gaussian-)MCA (Bornschein, Henniges, and Lücke, 2013; Lücke and Eggert, 2010) and SSSC models each with $H = 512$ components and 1000 variational EM iterations, and also BM3D to the corrupted image. The details of the MCA approach are presented in Sections 2.3 and A.4. All these three approaches are tailored to the Gaussian observation noise and can be directly applied to interval data. The first two approaches are, however, generative models and exploit an EM approach for training.

Figure 5.16 then depicts the results of this experiment where the reconstruction performances are compared using the standard measures of PSNR and also Mean Squared Error (MSE). Amongst the considered models, the B-MCA is found to achieve the best reconstruction performance in terms of both PSNR and MSE values. However, visual inspection may even favor the results of B-MCA (in comparison to the other three approaches) more than what PSNR and MSE values suggest.

Finally, in order to assess the performance of B-MCA for interval data if the noise is not Beta distributed, we repeated the experiment but this time using Gaussian noise. To this, we again considered the house image and corrupted the image by additive Gaussian noise with $\sigma = 10$. We then rescaled the corrupted image to the interval $[0.01, 0.99]$ and applied the B-MCA model, (Gaussian-)MCA and SSSC to denoise the image (rescaling enables the application of all models). Concretely, we performed 1000 variational EM iterations for each of the three models and used the same settings as for the previous experiments. The results are then presented in Figure 5.17. As the considered data is Gaussian distributed, we observed that the two models of MCA and SSSC, which are tailored to Gaussian noise, perform best.



(a) Corrupted (28.18/9.98) (b) B-MCA (32.63/5.97) (c) MCA (32.78/5.87) (d) SSSC (**36.09**/**4.01**)

FIGURE 5.17: Denoising the house image corrupted by Gaussian noise. (a) Image with added Gaussian-noise. (b)-(d) Reconstructed images obtained by applying three different data models. Numbers in parentheses are the corresponding PSNR (in dB) and MSE values (first number is PSNR, second number is MSE). As illustrated, denoising with Beta-MCA (B-MCA) and (Gaussian-)MCA works similarly well but the SSSC model shows the best performance. See text for further details.

The performance of B-MCA is almost as high as the performance of MCA, though, which is evidence for the flexibility of the Beta distribution. The SSSC approach is clearly the most competitive, which may not be surprising as the model (A) assumes the correct type of noise in this case, and (B) has a relatively flexible prior which is advantageous for denoising. On the other hand, when the noise is Beta-distributed and the SSSC noise assumption is thus not correct, B-MCA is observed to produce significantly better denoising results (Figure 5.16).

Chapter 6

Conclusion and Final Remarks

In the previous chapter, we evaluated the performance of the proposed LVMs and the corresponding update equations obtained from Theorems 2 and 3. In short, we observed that the considered ef-MCA models perform reliably well in different statistical tasks with different experimental settings at both small- and large-scales. We now discuss final details of the proposed generative models and their optimization algorithms in order to conclude the thesis. We also point out some possible generalizations that can be further considered as future research directions. Moreover, the content of the discussion section is taken from (Mousavi et al., 2020) (currently under review) and (Mousavi et al., 2021) which has been written jointly by Jörg Lücke and me.

6.1 Discussion

Latent variable models (LVMs) have found a widespread use in Machine Learning, Statistics and Artificial Intelligence applications. They can, in principle, be used to learn important patterns, clusters, statistical correlations and causal structures from a set of complex and unlabelled data (Olshausen and Field, 1996a; Mohamed, Heller, and Ghahramani, 2010; Everitt, 1984). For many cases, however, such powerful tools are tailored to the Gaussian distributed data and most commonly, a linear combination of the latents is considered as the main assumption of these approaches (Olshausen and Field, 1996a; Tipping and Bishop, 1999; Roweis, 1998; Cattell, 2012). Nevertheless, it has been long argued that such assumptions may limit the applications of the considered LVMs to specific datasets. As a consequence, other general (and more applicable) LVMs have been suggested in the literature (e.g., Roweis, 2003; Lücke and Sahani, 2008; Bornschein, Henniges, and Lücke, 2013; Sheikh et al., 2019; Lee et al., 2009; Collins, Dasgupta, and Schapire, 2002; Mohamed, Ghahramani, and Heller, 2008). Each of these studies attempts to modify the two mentioned assumptions by either incorporating a certain non-linearity into the model or generalizing the assumption of Gaussian distributed data to, e.g., other distributions of the exponential family. Amongst the studies that have been conducted in this direction, however, there have been few contributions which focus on both generalizations at the same time. One example is the work by Lücke and Sahani (Lücke and Sahani, 2008) which investigates Poisson distributed data along with a maximum superposition function. Investigating other intricate distributions of the exponential family along with such a maximum superposition require addressing many challenges that have been the subject of our work here. The current study can therefore be seen as a general approach for introducing a family of non-linear LVMs which are capable of training different data types.

A central element of all LVMs is the derivation of parameter update equations, specifically the weights that couple the latent variables to the observables (the dictionary). In this concept, the specific form of dictionary update suggested for non-negative matrix factorization (NMF) is prominently a hallmark of seminal papers for dictionary learning (Lee and Seung, 1999). Broadly speaking, deriving a set of update equations that is, in general, applicable to any distribution of the exponential family is an arduous task even for models with a linear

superposition. For instance, we are not aware of any contribution reporting a set of update equations for the parameters of a Beta distribution in the framework of LVMs. Studies such as (Collins, Dasgupta, and Schapire, 2002; Lee et al., 2009; Mohamed, Ghahramani, and Heller, 2008) attempt to tackle this challenge, but their definitions do not contain intricate cases of the exponential family such as Beta or Gamma (at least they have not discussed the procedure of applying these complicated distributions as the noise model and/or the details of the parameter optimization procedure for such distributions). It should be mentioned that all these three studies exploit a linear combination of the latents. For any model with a non-linear combination of generative fields, derivation of the dictionary updates is more challenging because closed-form solutions are typically hard to obtain (Saul, Jaakkola, and Jordan, 1996; Šingliar and Hauskrecht, 2006; Lücke and Sahani, 2008; Frolov, Husek, and Polyakov, 2014; Gan et al., 2015). Nonetheless, it has been discussed that a non-linear superposition can be more closely aligned with the true data generating process of different data types (Roweis, 2003; Lücke and Sahani, 2008; Bornschein, Henniges, and Lücke, 2013; Frolov, Húsek, and Polyakov, 2015; Sheikh et al., 2019). Therefore, in order to obtain an algorithm capable of extracting the true structural primitives (in the form of learned dictionaries) for different data types, derivation of the update rules from the corresponding non-linearities is required. This importantly necessitates the existence of a general and applicable optimization procedure for such models.

A non-linearity which may be considered as a natural alternative to summation of the latents is maximization. The maximum function has been investigated for occlusion-like non-linearities in image data (Lücke and Sahani, 2008; Lücke and Eggert, 2010; Bornschein, Henniges, and Lücke, 2013) as well as masking based non-linearities in sounds (Roweis, 2003; Lücke and Sahani, 2008; Sheikh et al., 2019). While it would be challenging to address the maximum non-linearity with any gradient descent (ascent) approach, it has convenient analytical properties which can be exploited here (see Lemma 1). These properties allowed us to derive our main result: A general set of parameter update equations applicable to any regular distribution of the exponential family. In this manner, we presented a general parameter optimization approach for the proposed family of generative models and specifically, proved that the same mathematical formulation can be derived for the updates of the models' parameters.

In addition, we exploited a novel generalization for the link function (the used maximum function) which enabled us to relate the latent variables and observables using two matrices (if the considered noise distribution has two parameters). Intuitively, we here argued that LVMs do require a combination of means as well as variances in order to better describe the statistical dependencies of the complex datasets such as medical data. We consequently addressed the implicit modelling challenges by exploiting exclusiveness (Lücke and Sahani, 2008; Dai, Exarchakis, and Lücke, 2013; Henniges et al., 2014). Thus, the family of generative models that we proposed here provide individual variance parameters (or parameters of the second moment) per latent variable. For mixture models such as Gaussian mixtures, individual variances per latent often carry important information (Zoran and Weiss, 2009). For multiple-causes models, however, individual variances have not been considered so far, and we are not aware of any linear or non-linear LVM for which similar ways of encoding the model's parameters have been proposed. We believe that such a shortage can limit the modelling capabilities as our example for medical data illustrated.

Exponential family SC (EF-SC) (Lee et al., 2009), which is a closely related approach to our work, used MAP-based optimization and distributions of the exponential family with a linear superposition that is designed to set the natural parameters of the considered observation noise. Such a link from latents to observables has advantages in terms of MAP-based training (mono-modal posteriors are maintained). A disadvantage is that different observable distributions essentially define different non-linearities into the model (because the natural

parameters are in general non-linear functions of the mean). Also how to exploit two (or more) parameter distributions (specifically how to define the coupling) seems to be challenging for this approach alongside many of the similar approaches. At least, neither EF-SC nor PCA-like approaches which use the same coupling (Collins, Dasgupta, and Schapire, 2002; Mohamed, Ghahramani, and Heller, 2008) report numerical results for more than one-parameter distributions of the exponential family (only Poisson and Bernoulli distributions have been treated in these contributions which, for both cases, we showed in Section 3.1.2 that a closed-form transformation from natural parameter to the mean is available). Additionally, EF-SC would not be able to learn prior parameters because of the used MAP approach (additional cross-validation would be required). In contrast, the generative models presented here are generally defined with parameter update equations given by Theorems 2, 3 and 4 that are directly applicable to any regular member of the exponential family. Theorems 2 and 4 provide updates for the observables' parameters and, furthermore, Theorem 3 allows for learning prior parameters. Hence, we constantly argue in favor of the proposed LVMs and the proposed parameter optimization algorithm here which represents one of the most flexible approaches that have been suggested so far for exponential family observables. As such, the proposed models also enable using many more distributions of the exponential family that have not been investigated before (in the framework of LVMs) including Categorical, Dirichlet, Normal-Gamma etc.

Another related approach to address non-binary or non-Gaussian types of observation statistics in other contexts is the work by Vergari et al. (Vergari et al., 2019). Their approach focuses on discovering the statistical data types automatically, and further providing a number of automatic data analysis tasks (also see Valera and Ghahramani, 2017; Valera et al., 2020). While being applicable to interval data (and more generally to many different data types), the approach does not provide a model for the combination of the means and variances which we believe is of importance, e.g., for analysis of medical data. Furthermore, large-scale applicability (e.g., for images) of the method has not been reported.

Other lines of research include deep learning approaches (such as Rajkomar et al., 2018; Shickel et al., 2017; Remez et al., 2017; Kumwilaisak et al., 2020; DeGuchy et al., 2019) that have been widely used for data analysis. The usage of such approaches, however, can not be motivated for our study here as the size of used data, in some cases, is limited. For instance, the medical data that we used here comprises CAFPA values for 287 single patients that have been collected and labelled manually by the experts (see Buhl et al., 2020). In such a regime, most deep-based algorithms fail to provide a reliable result as they usually require a large number of (clean) data points for training. This is also the case for other datasets such as images and for tasks like denoising where DNNs substantially differ from the used probabilistic generative models. One example is the DenoiseNet algorithm (Remez et al., 2017) that is designed for denoising images corrupted with Poisson noise and requires a large number of images (for different peak values) for training. On the contrary, the approach presented here uses only the considered noisy data (the noisy image) for training. This has been, in a broader sense, termed as the "zero-shot" learning (see Shocher, Cohen, and Irani, 2018; Imamura, Itasaka, and Okuda, 2019; Drefs, Guiraud, and Lücke, 2020). Besides, another important argument that exists against using deep-based approaches is the fact that usually the final results and obtained inferences are not interpretable (see, e.g., Pearl, 2014; Fei and Li, 2017; Ravuri et al., 2018 for a discussion). This is specifically vital when the considered algorithm is employed for the task of medical data reasoning where the experts tend to avoid black-box approaches. Although models such as VAEs (Kingma and Welling, 2013; Rezende, Mohamed, and Wierstra, 2014) do provide statistical information of the data generating process, generalization of these models to meet the desired properties raised here (e.g., to consider exponential family noise distributions or the coupling of component means and component variances) seems to be formidable. Moreover, conventional VAEs exploit continuous latent variables and their extension to binary latents (and in general to discrete

latents) is very challenging since the used optimization algorithm, the backpropagation, cannot be used in conjunction with discrete latents (Rolfe, 2016; Bengio, Léonard, and Courville, 2013). To address the latter issue, the usage of specific treatments have been further suggested in the literature in order to maintain the general optimization framework of VAEs for the models with discrete latent variables (Rolfe, 2016; Lorberbom et al., 2018; Oord, Vinyals, and Kavukcuoglu, 2017; Roy et al., 2018). In this direction, the study by Guiraud et al. (Guiraud, Drefs, and Lücke, 2020) can be seen as an exception which introduces an evolutionary optimization algorithm for the VAEs with binary latents.

Our numerical experiments based on one- and two-parameter distributions of the exponential family suggest that feasible and competitive learning algorithms are obtained if the results of Theorems 2, 3 and 4 are applied. Especially when symptom variances carry important information, our results suggest that such information can help inference (see the medical data analysis and feature extraction experiments in Sections 5.2.1 and 5.2.2, respectively). Furthermore, learning variance information unsupervised could also help in characterizing diseases in medical data applications. In short, we showed that the proposed family of generative models enables us to reliably express the underlying generating process of many different datasets. Such information allows researchers to build representations of the observables which will be used for reasoning, predicting, decision making and ultimately optimal inference. In addition, we have demonstrated scalability of the proposed approaches, e.g., by using approximate posteriors for efficient variational optimization (see Section 4.4). Thanks to the newly established EEM approach (Guiraud, Drefs, and Lücke, 2018; Drefs, Guiraud, and Lücke, 2020), which can be used as a black-box method, we were able to efficiently extend our optimization procedure to large-scale models. For such an approach, importantly, no additional derivation was required for any new distribution of the exponential family.

By using the Beta-MCA model as one example of the family of generative models studied here, we investigated the task of inferring binary causes from interval data. The usage of interval data has been commonly exploited in medical data recordings (Buhl et al., 2020; van Esch et al., 2013; Lehnhardt, 2009), but also emerges in other domains of machine learning. For instance, we considered intervals of $[0, 255]$ for grey level images. To infer latent causes, any interval data could in principle be binarized in order to apply standard models such as noisy-OR-like nets. Alternatively, one can endeavor to model interval data directly, and we here addressed such a task using a maximum non-linear superposition.

Other applications included structure finding in image patches, automatic estimation of the noise distributions (also compare (Vergari et al., 2019) and (Valera and Ghahramani, 2017)), and denoising images subject to non-Gaussian noises. In the case of Poisson noise, where a number of alternative approaches are available, we observed a competitive performance of our model. In particular, denoising results improved on models which used Poisson noise with a linear superposition (NLSPCA (Salmon et al., 2014) and SPDA (Giryes and Elad, 2014)). Comparisons with NLSPCA and SPDA approaches may be particularly interesting in the context of this work because (A) both NLSPCA and SPDA use the standard coupling to observables via the natural parameter of the Poisson, whereas Poisson-MCA sets the mean; and also (B) NLSPCA and SDPA use a linear superposition, while Poisson-MCA uses a maximum. Comparison in Table 5.5 (and also Table B.1) showed, in general, a better denoising performance of the Poisson-MCA compared to both NLSPCA and SDPA. Nonetheless, Poisson denoising is just one example for which the results of Theorems 2 and 3 apply. We further showed the denoising results of images corrupted by Exponential and Beta noises. Such noise distributions have not been investigated in the context of denoising or other similar tasks before and the experiments studied here revealed the effectiveness and importance of LVMs with such noise distributions. That, for instance, Poisson-MCA (and also other ef-MCA models) is competitive for the denoising task without further mechanisms (compare NLSPCA and SDPA) and without extensive fine-tuning may be taken as evidence in

favor of the general approaches proposed in this study.

6.2 Summary

In this thesis, we presented a family of non-linear generative models which encompasses a large variety of noise distributions from the exponential family. In this context, we studied Gaussian, Poisson, Exponential, Bernoulli, Gamma and Beta distributed data (we only chose these examples to show the potential of the family of generative models presented here). We further defined a novel non-linear combination of the latents and observables using a point-wise maximum function. Such a combination was specifically designed to retain the main property of previously established link functions in superimposing the active causes for setting the mean of observables. We considered a double-dictionary approach (given that the noise distribution is a two-parameter distribution) and introduced two matrices (along the prior) as the parameters of the model: One to model the component means and another for the component variances (or the second moment). Given that, we were consequently able to learn component variances per cause which rendered valuable information for an optimal inference. We then exploited an EM algorithm to train our family of generative models and took advantage of the proposed link function to derive a set of succinct and concise update equations. Importantly, we proved that the same functional form can be obtained for the parameter updates of any regular distribution of the exponential family. These updates were then evaluated using both synthetic and real datasets.

We first used the bars test to assess the effectiveness of the derived update equations. Given the test and different settings of the hyperparameters, the considered ef-MCA models were able to (in most cases) recover the true ground-truth parameters with a high precision. Furthermore, as one central application, we used the Beta-MCA model (which is an example of the generative models presented here) for training the medical data of hearing impairments (we used CAFPA values described in Buhl et al., 2020), and later used the trained model to predict the responsible causes (the audiological findings) for a set of unseen data points (patients in this case). The results consequently illustrated the efficacy of the considered model in extracting reliable causes/symptoms relations from the data and performing favorably compared to a probabilistic noisy-OR model. We also examined the performance of the proposed ef-MCA models (we specifically used Poisson-, Exponential-, Bernoulli-, Gaussian-, Gamma- and Beta-MCA models) in other applicable tasks such as denoising, feature extraction and noise-type estimation. For these specific tasks, we further exploited variational approximations in the form of truncated posteriors to effectively scale the used algorithms (we used up to hundreds of hidden states). In all experiments, we observed a consistent and robust performance of the generative models introduced here, which in some cases, produced high-quality results that can compete with the state-of-the-art approaches.

6.3 Outlook

We developed a family of generative models that are able to learn hidden features from a set of unlabelled data. We believe that such powerful interpretable models are of interest in many different areas which can be further investigated. First of all, from a theoretical point of view, one can investigate refinements of the proposed ef-MCA models as future research directions. This includes, for instance, exploring other properties of the exponential family distributions that can be generalized and employed in the context of LVMs introduced in (3.25)-(3.27). In addition, implementations of different ef-MCA models are also of interest. In general, one can establish a platform such that many of the proposed ef-MCA generative models (at least the frequently used models) are implemented and accessible without much effort. Then, a feasible

approach would search amongst these models to find the most suitable noise distribution for a given dataset (e.g., by estimating the noise-type of the data similar to what we discussed in Section 5.2.3), and later employ the proper ef-MCA model automatically. The outcome of such a "black-box" approach could be, e.g., inferences for unseen data points or relevant information that can be used by experts. This is intuitively in conjunction with the EEM approach employed here for variational approximations which can be used independently of the chosen noise distribution (it only requires the joint distribution for a given data model). Nonetheless, enhancements of the variational EM algorithm that is used can also be pursued.

Secondly, applications of the proposed generative models can be further investigated by considering different datasets and/or different tasks (this can be seen from a practical point of view). In analogy to the use of standard SC approaches, further application examples would include inpainting, compression, feature learning, or compressed sensing for data with in principle any exponential family noise distribution. The experiments that we used here illustrated the huge potential of the proposed models which may positively affect the current status of the machine learning algorithms. As the proposed models leverage a variety of probability distributions to treat different data types, the range of their applications are subsequently large which creates many opportunities for future research studies. In particular, the usage of Categorical and Multinomial distributions (the application of Categorical- and Multinomial-MCA models) are very desired.

Other future work may also include the use of more richly structured prior distributions. As the derived update equations apply in general for any binary latents, the independent Bernoulli prior could be replaced by a deep model. Binary latents for deep generative graphical models are very common such as deep SBNs (Saul, Jaakkola, and Jordan, 1996; Gan et al., 2015) and deep restricted Boltzmann Machines (Larochelle and Bengio, 2008) which may also be used.

Appendix A

Additional Details on Parameter Update Equations

We consider the three models of Gaussian-, Gamma- and Beta-MCA and provide the details of their parameter update rules. These are two-parameter distributions of the exponential family and therefore, for each of these cases, we have $\Theta = (\vec{\pi}, W, V)$ and also $M(\Theta)$ and $\Sigma^2(\Theta)$ as the mean and variance components, respectively. We assessed the performance of these models using different experiments in Chapter 5 and showed the efficacy of their learning algorithms (either using full EM or variational EM). Now, we present extra explanations of their algorithms in the following. Besides the three ef-MCA models, we also provide the details of M-step update equations for the MCA and noisy-OR models described in Chapter 2.

The content of upcoming sections are taken from (Mousavi et al., 2020) (currently under review) and (Mousavi et al., 2021). Descriptions of the Gaussian-MCA model (Section A.1) were written by Jörg Lücke, and later revised and completed by me. Next, I wrote Sections A.2 and A.3 and provided the details for Gamma- and Beta-MCA models. The details of Section A.4 are further taken from (Bornschein, Henniges, and Lücke, 2013), and finally the formulas presented in Section A.5 are provided by Enrico Guiraud.

A.1 Parametrization of the Gaussian-MCA

For Gaussian-MCA in Example 2, the generative model was shown to be given by:

$$p(\vec{s}\,|\Theta) = \prod_{h=1}^{H} \pi_h^{s_h}(1-\pi_h)^{1-s_h} \tag{A.1}$$

$$p(\vec{y}\,|\,\vec{s},\Theta) = \prod_{d=1}^{D} \mathcal{N}(y_d; \bar{W}_d(\vec{s},\Theta), \bar{V}_d(\vec{s},\Theta) - \bar{W}_d^2(\vec{s},\Theta)) \tag{A.2}$$

where we can use the update equations of Theorem 2 for W and V. To obtain some intuition, we can also define the variance function to be given by:

$$\bar{\Sigma}_d^2(\vec{s},\Theta) := \bar{V}_d(\vec{s},\Theta) - \bar{W}_d^2(\vec{s},\Theta), \quad \forall d. \tag{A.3}$$

Consequently, the Gaussian-MCA generative model becomes:

$$p(\vec{s}\,|\Theta) = \prod_{h=1}^{H} \pi_h^{s_h}(1-\pi_h)^{1-s_h} \tag{A.4}$$

$$p(\vec{y}\,|\,\vec{s},\Theta) = \prod_{d=1}^{D} \mathcal{N}(y_d; \bar{W}_d(\vec{s},\Theta), \bar{\Sigma}_d^2(\vec{s},\Theta)) \tag{A.5}$$

The latents thus change the mean via the matrix W and the variance via the matrices V and W. Now, given the definitions of $\bar{W}_d(\vec{s},\Theta)$ and $\bar{V}_d(\vec{s},\Theta)$ in (3.38), we can introduce a matrix $\Sigma(\Theta)$ with $D \times H$ entries corresponding to the component standard deviations (and its squared to the component variances) and let:

$$\Sigma^2_{dh}(\Theta) = V_{dh} - W^2_{dh}. \tag{A.6}$$

Then, considering the two cases of $h = h(d,\vec{s},\Theta)$ and $h \neq h(d,\vec{s},\Theta)$ separately, one can easily show that definition (A.3) is equivalently given by:

$$\bar{\Sigma}^2_d(\vec{s},\Theta) := \Sigma^2_{dh(d,\vec{s},\Theta)}(\Theta) \;\text{ where }\; h(d,\vec{s},\Theta) = \text{argmax}_h\{W_{dh}s_h\}.$$

Note that in the case of Gaussian $W = M(\Theta)$. We can use the update rules for V_{dh} and W_{dh} and compute the matrix $\Sigma^2_{dh}(\Theta)$ in each M-step using (A.6). Alternatively, we can also combine the update rules for V_{dh} and W_{dh} to directly obtain an update rule for $\Sigma^2_{dh}(\Theta)$:

$$\begin{aligned}
(\Sigma^2{}_{dh}(\Theta))^{\text{new}} &= V^{\text{new}}_{dh} - (W^{\text{new}}_{dh})^2 = V^{\text{new}}_{dh} - 2\,W^{\text{new}}_{dh}\,W^{\text{new}}_{dh} + (W^{\text{new}}_{dh})^2 \\
&= \frac{\sum_{n=1}^N \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}} (y^{(n)}_d)^2}{\sum_{n=1}^N \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}} - 2\frac{\sum_{n=1}^N \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}} y^{(n)}_d}{\sum_{n=1}^N \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}} W^{\text{new}}_{dh} + (W^{\text{new}}_{dh})^2 \\
&= \frac{\sum_{n=1}^N \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}} \left((y^{(n)}_d)^2 - 2y^{(n)}_d W^{\text{new}}_{dh} + (W^{\text{new}}_{dh})^2 \right)}{\sum_{n=1}^N \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}} \\
&= \frac{\sum_{n=1}^N \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}} \left(y^{(n)}_d - W^{\text{new}}_{dh} \right)^2}{\sum_{n=1}^N \langle \mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}}.
\end{aligned} \tag{A.7}$$

This form of update is more familiar as it evaluates the square deviation from the mean given by W^{new}_{dh}. Also note that we first have to compute W^{new}_{dh} before updating $\Sigma^2_{dh}(\Theta)$, which is in analogy, e.g., to standard Gaussian mixtures for variance (or covariance) updates. For the Gaussian-MCA application of Section 5.2.2, we showed the matrices W_{dh} and $\Sigma^2_{dh}(\Theta)$ for whitened natural image patches.

A.2 Parametrization of the Gamma-MCA

Let the Gamma distribution to be the noise of observables in the data model (3.25)-(3.27); i.e., consider the following generative model:

$$p(\vec{s}\,|\Theta) = \prod_{h=1}^H \pi_h^{s_h}(1-\pi_h)^{1-s_h} \tag{A.8}$$

$$p(\vec{y}\,|\,\vec{s},\Theta) = \prod_{d=1}^D \text{Gamma}(y_d;\vec{\eta}_d(\vec{s},\Theta)) \quad \text{for} \quad y_d \in (0,\infty). \tag{A.9}$$

According to Section 3.1.2, for the case of Gamma distribution we have:

$$\vec{\Phi}(\vec{w}) \approx \frac{1}{2(\log(w_1) - w_2)} \begin{pmatrix} -1/w_1 \\ 1 - 2(\log(w_1) - w_2) \end{pmatrix}. \tag{A.10}$$

Therefore, based on the definition of $\vec{\tilde{\eta}}_d(\vec{s},\Theta)$ in (3.28), we can write:

$$\vec{\tilde{\eta}}_d(\vec{s},\Theta) = \vec{\Phi}\big(\bar{W}_d(\vec{s},\Theta),\bar{V}_d(\vec{s},\Theta)\big) \approx \begin{pmatrix} \frac{-1}{2\bar{W}_d(\vec{s},\Theta)\left(\log(\bar{W}_d(\vec{s},\Theta))-\bar{V}_d(\vec{s},\Theta)\right)} \\ \frac{1}{2\left(\log(\bar{W}_d(\vec{s},\Theta))-\bar{V}_d(\vec{s},\Theta)\right)} - 1 \end{pmatrix}.$$

On the other hand, for any Gamma distribution we know that the variance parameter is given by:

$$\sigma^2 = \frac{\alpha}{\beta^2} = \frac{\eta_2+1}{\eta_1^2} \tag{A.11}$$

where α and β are the shape and rate parameters and η_1 and η_2 are the natural parameters (from (3.9), we have $\vec{\eta} = \vec{\Phi}(\vec{w})$). Then, assuming $\bar{\Sigma}^2_d(\vec{s},\Theta)$ to denote the variance of the Gamma-MCA model for each d, we have:

$$\begin{aligned} \bar{\Sigma}^2_d(\vec{s},\Theta) &= \frac{\Phi_2\big(\bar{W}_d(\vec{s},\Theta),\bar{V}_d(\vec{s},\Theta)\big)+1}{\Phi_1^2\big(\bar{W}_d(\vec{s},\Theta),\bar{V}_d(\vec{s},\Theta)\big)} \\ &\approx 2\bar{W}^2_d(\vec{s},\Theta)\Big(\log(\bar{W}_d(\vec{s},\Theta)) - \bar{V}_d(\vec{s},\Theta)\Big). \end{aligned} \tag{A.12}$$

Now, in analogy to the Gaussian-MCA model, we introduce a matrix $\Sigma(\Theta)$ with $D \times H$ entries and subsequently define:

$$\bar{\Sigma}_d(\vec{s},\Theta) := \Sigma_{dh(d,\vec{s},\Theta)}(\Theta) \quad \text{where} \quad h(d,\vec{s},\Theta) = \text{argmax}_h\{W_{dh}s_h\} \tag{A.13}$$

which using (A.12) results in:

$$\Sigma^2{}_{dh}(\Theta) \approx 2W^2_{dh}\big(\log(W_{dh}) - V_{dh}\big). \tag{A.14}$$

This relation enables us to (approximately) compute the variance components of the Gamma-MCA model in each M-step using matrices W and V. Consequently, the Gamma-MCA model can be restated by:

$$p(\vec{s}\,|\Theta) = \prod_{h=1}^{H} \pi_h^{s_h}(1-\pi_h)^{1-s_h} \tag{A.15}$$

$$p(\vec{y}\,|\,\vec{s},\Theta) = \prod_{d=1}^{D} \text{Gamma}(y_d; \bar{W}_d(\vec{s},\Theta),\bar{\Sigma}^2_d(\vec{s},\Theta)) \quad \text{for} \quad y_d \in (0,\infty) \tag{A.16}$$

where the Gamma distribution is parameterized w.r.t. its mean and variance parameters. Similar to the Gaussian-MCA, we can use Equations (4.27)-(4.28) and (4.33) to update parameters Θ of the model and further compute $\Sigma^2(\Theta)$ using (A.14) or instead, directly update $\Sigma^2(\Theta)$ as follows:

$$
\begin{aligned}
(\Sigma^2{}_{dh}(\Theta))^{\text{new}} &\approx 2(W_{dh}^{\text{new}})^2\Big(\log(W_{dh}^{\text{new}}) - V_{dh}^{\text{new}}\Big) \\
&= 2(W_{dh}^{\text{new}})^2\Big(\log(W_{dh}^{\text{new}})\frac{\sum_{n=1}^{N}\langle\mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}}{\sum_{n=1}^{N}\langle\mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}} - \frac{\sum_{n=1}^{N}\langle\mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}\log(y_d^{(n)})}{\sum_{n=1}^{N}\langle\mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}}\Big) \\
&= 2(W_{dh}^{\text{new}})^2\Big(\frac{\sum_{n=1}^{N}\langle\mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}\log(W_{dh}^{\text{new}})}{\sum_{n=1}^{N}\langle\mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}} - \frac{\sum_{n=1}^{N}\langle\mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}\log(y_d^{(n)})}{\sum_{n=1}^{N}\langle\mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}}\Big) \\
&= 2\frac{\sum_{n=1}^{N}\langle\mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}(W_{dh}^{\text{new}})^2\big(\log(W_{dh}^{\text{new}}) - \log(y_d^{(n)})\big)}{\sum_{n=1}^{N}\langle\mathcal{A}_{dh}(\vec{s},\Theta)\rangle_{q^{(n)}}}. \qquad \text{(A.17)}
\end{aligned}
$$

Likewise, we here first update the matrix W (which is equal to $M(\Theta)$) and then use the updated components to compute the variance $\Sigma^2(\Theta)$. In addition, observe that the above equation represents an approximation of the variance components since the Digamma function is approximated with its first two terms. Despite the approximation used here, we observed in our experiments that the equation provides a good estimation of the variance parameter. In practice, nevertheless, one can exploit an enhanced estimation of the variance by considering a better approximation of the Digamma function and consequently improve the performance of the model. For our experiments, we have also tried the first three terms of the Digamma function as an approximation where we were able to consistently observe an improvement in the variances' estimations. But, this (of course) results in a more intricate update equation than (A.17).

A.3 Parametrization of the Beta-MCA

For the Beta-MCA model, the proposed generative model can be written as follows:

$$
p(\vec{s}\,|\Theta) = \prod_{h=1}^{H}\pi_h^{s_h}(1-\pi_h)^{1-s_h} \qquad \text{(A.18)}
$$

$$
p(\vec{y}\,|\,\vec{s},\Theta) = \prod_{d=1}^{D}\text{Beta}(y_d;\vec{\eta}_d(\vec{s},\Theta)) \quad \text{for} \quad y_d \in [0,1]. \qquad \text{(A.19)}
$$

Based on our definitions in Section 3.1.2, we know that the mean value parameters of the Beta distribution are given by:

$$
\bar{W}(\vec{s},\Theta) := \langle\log(\vec{y})\rangle_p \quad \text{and} \quad \bar{V}(\vec{s},\Theta) := \langle\log(1-\vec{y})\rangle_p. \qquad \text{(A.20)}
$$

where $\bar{W}(\vec{s},\Theta)$ and $\bar{V}(\vec{s},\Theta)$ are introduced in (3.38).

Such a model results in learning two dictionaries: W and V. Having these two dictionaries and further assuming the existence of the function $\vec{\Phi}$ in (3.9) (we assume $\eta_1 \neq \eta_2$), we can compute the natural parameters as well as the mean and variance parameters of the Beta distribution. For the case of Beta-MCA, the use of mapping $\vec{\Phi}$ from mean value parameters to natural parameters is not only required for the formal definition of the generative model, but also the mapping will be required for the concrete parameter updates, Equations (4.27)-(4.28) and (4.33). These updates are formulated in terms of the mean value parameters, and at the same time, the updates also require expectation values w.r.t. the posteriors which are computable in closed-form for the natural parameters. Additionally, note (see below) that the mean $\bar{M}(\vec{s},\Theta)$ is given by a closed-form expression of the natural parameters (because $T_1(y) = y$ does not apply for the Beta distribution and therefore we require to compute the

weight matrix $M(\Theta)$ using the natural parameters or equivalently using the mapping $\vec{\Phi}$). To see this, let $\Sigma(\Theta)$ be a $D \times H$ matrix containing, for each cause h, a set of D different standard deviations (one for each observable). Then, for $d = 1, \ldots, D$ and $h = 1, \ldots, H$, we define (based on the relations between the mean and variance of the Beta distribution and its natural parameters, see (3.19)):

$$M_{dh}(\Theta) := \frac{\Phi_1(W_{dh}, V_{dh})}{\Phi_1(W_{dh}, V_{dh}) + \Phi_2(W_{dh}, V_{dh})} \tag{A.21}$$

$$\Sigma^2_{dh}(\Theta) := \frac{\Phi_1(W_{dh}, V_{dh})\Phi_2(W_{dh}, V_{dh})}{\big(\Phi_1(W_{dh}, V_{dh}) + \Phi_2(W_{dh}, V_{dh})\big)^2\big(\Phi_1(W_{dh}, V_{dh}) + \Phi_2(W_{dh}, V_{dh}) + 1\big)}. \tag{A.22}$$

Equations (A.21) and (A.22) taken together with Equation (3.38) model how component means and component variances can be obtained using the function $\vec{\Phi}$ (note that $\vec{\Phi}(\vec{w}) = \vec{\eta}$). We further take on the same approach as for the Gaussian- and Gamma-MCA models and let:

$$\begin{aligned} &\bar{M}_d(\vec{s}, \Theta) = M_{dh(d,\vec{s},\Theta)}(\Theta) \;\text{ and }\; \bar{\Sigma}_d(\vec{s}, \Theta) = \Sigma_{dh(d,\vec{s},\Theta)}(\Theta) \\ &\text{where }\; h(d, \vec{s}, \Theta) = \text{argmax}_h\{M_{dh}(\Theta)s_h\}. \end{aligned} \tag{A.23}$$

The proposed generative model (A.18)-(A.19) can be considered as a generalization of binary-binary models such as noisy-OR or Sigmoid Belief Networks (SBNs) towards continuously distributed interval data. The model takes advantage of the previous MCA approaches but decisively amends the original model in the central aspect of our goals that are: (a) modelling interval data by using Beta distributions, and (b) exploiting properties of the maximum that allows to model a combination of symptom variances alongside a combination of symptom means. In detail, the dominant cause is obtained using the maximum function that corresponds to the effective weight on observable y_d (which sets the mean), and further determines the mean value parameters and consequently the variance of the Beta distribution. Similar to the previous MCA models, for any one generation, a different cause can be the dominant cause for each symptom which here not only determines the symptom mean but also the symptom variance. Our running example for medical data may make the necessity of modelling variance combinations relatively obvious.

For the EM algorithm of the Beta-MCA, we initialize parameters Θ and update each of the parameters using Equations (4.27)-(4.28) and (4.33) in the M-step. The alternative E- and M-steps will continue until parameters Θ have sufficiently converged. The optimal value of Θ^* then obtains at the convergence point. Nevertheless, we still require the function $\vec{\Phi}(W_{dh}, V_{dh})$ in order to compute matrices $M(\Theta)$ and $\Sigma^2(\Theta)$. To this, we recall the definitions presented in Section 3.1.2 for the Beta distribution and apply a straightforward method and solve the following system of non-linear equations w.r.t. $D \times H$ matrices η_1 and η_2 in each M-step:

$$\begin{cases} W - \psi(\eta_1) + \psi(\eta_1 + \eta_2) = 0 \\ V - \psi(\eta_2) + \psi(\eta_1 + \eta_2) = 0 \end{cases} \tag{A.24}$$

We finally let $\Phi_1(W, V) = \eta_1$ and $\Phi_2(W, V) = \eta_2$ in (A.21) and (A.22) to compute matrices $M(\Theta)$ and $\Sigma^2(\Theta)$ corresponding to the mean and variance components. The obtained weight matrix $M(\Theta)$ can then be used to distinguish the dominant cause in (3.38). This set of updates provides an applicable EM method that can be used to train interval data using the proposed Beta-MCA model. Algorithm 2 presents a brief summary of the exact EM algorithm for this specific model.

Algorithm 2: Exact EM for parameter updates of the Beta-MCA model

initialize model parameters $\Theta = (\vec{\pi}, W, V)$;
repeat
  solve the system of two non-linear equations (A.24) and obtain η_1 and η_2;
  let $\Phi_1(W,V) = \eta_1$ and $\Phi_2(W,V) = \eta_2$;
  compute the weight matrix $M(\Theta)$ and the variance matrix $\Sigma^2(\Theta)$ using (A.21)-(A.22);
  for *each vector* $\vec{s}$ *of the latent space* **do**
    for $d = 1 : D$ **do**
      $h(d,\vec{s},\Theta) = \text{argmax}_h\{s_h M_{dh}(\Theta)\}$;
      $\bar{W}_d = W_{dh(d,\vec{s},\Theta)}$ and $\bar{V}_d = V_{dh(d,\vec{s},\Theta)}$;
    end
    for $n = 1 : N$ **do**
      $q^{(n)}(\vec{s}) = p(\vec{s} \,|\, \vec{y}^{(n)}, \Theta)$;
      for $h = 1 : H$ *and* $d = 1 : D$ **do**
        compute $q^{(n)}(\vec{s}) s_h$;
        compute $q^{(n)}(\vec{s}) \mathcal{A}_{dh}(\vec{s},\Theta)$ where $\mathcal{A}_{dh}(\vec{s},\Theta)$ is defined in (4.14);
      end
    end
  end
  update parameters Θ using (4.27)-(4.28) and (4.33);
until *parameters* Θ *have sufficiently converged*;

It should be mentioned that, for e.g., computing the term $\psi(\eta_1)$ in (A.24) involves computations of $\log(\eta_1)$, $1/\eta_1$ and also other terms which in the case that $\eta_1 \longrightarrow 0$ may cause numerical errors (the same is for the terms $\psi(\eta_2)$ and $\psi(\eta_1+\eta_2)$). Such cases should be then treated carefully to avoid degeneracies. In our implementations, we always checked the values of η_1 and η_2 obtained from system of equations (A.24) to be positive (in the case that a non-positive value encountered in our iterations, we simply set it manually to a small positive value, like 10^{-4}). Likewise, we checked for the values of $M(\Theta)$ and $\Sigma^2(\Theta)$ to be in the range that are justified for the Beta distribution. We emphasize that such sanity checks should be considered in our optimization algorithms in order to avoid inaccuracies or numerical errors; and we did the same also for the other ef-MCA models. For instance, for the Gamma- and Exponential-MCA models, we always checked for the component means (W_{dh} in this case) to be positive.

A.3.1 M-step updates – a global variance parametrization

In order to train the Beta-MCA model with a global variance parameter, one can simply use the updates (4.27)-(4.28) together with (A.22) to compute the matrix $\Sigma^2(\Theta)$ and then average over all elements of the matrix. However, we here consider another approach and directly update a scalar variance σ^2 from the model. To this, we assume the Beta distribution to be parameterized w.r.t. its mean $\bar{M}_d(\vec{s},\Theta)$ for $d = 1,\ldots,D$ and a global variance σ^2. That is we consider a relatively well interpretable but preliminary generative model as follows:

$$p(\vec{s}\,|\Theta) = \prod_{h=1}^{H} \pi_h^{s_h}(1-\pi_h)^{1-s_h} \tag{A.25}$$

$$p(\vec{y}\,|\vec{s},\Theta) = \prod_{d=1}^{D} \text{Beta}(y_d; \bar{M}_d(\vec{s},\Theta), \sigma^2), \quad y_d \in [0,1] \tag{A.26}$$

$$\text{where} \quad \bar{M}_d(\vec{s},\Theta) = \max_h\{s_h M_{dh}\}, \quad d = 1,\ldots,D \tag{A.27}$$

where the Beta distribution is parameterized w.r.t. the distribution mean and the distribution variance. Consequently, the parameters of the model are $\Theta = (\vec{\pi}, M, \sigma^2)$.

We here update parameter M using (4.27)-(4.28) and (A.21), and $\vec{\pi}$ using (4.33). Further, to obtain a σ^2 update equation in the M-step, we set the derivative of the free energy w.r.t. the parameter σ^2 to zero, i.e. $\frac{\partial \mathcal{F}}{\partial \sigma^2} = 0$. For the data model (A.25)-(A.27), the free energy function is given by:

$$\mathcal{F}(q,\Theta) = \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s})\{\sum_d \log(p(y_d^{(n)}|\vec{s},\Theta)) + \sum_h \log(p(s_h|\Theta))\} + \mathcal{H}(q). \tag{A.28}$$

Now, abbreviating $\bar{M}_d(\vec{s},\Theta)$ by $\bar{M}_d$, we get (recall the equations given by (3.20)):

$$\frac{\partial}{\partial\sigma^2}\mathcal{F}(q,\Theta) = \sum_n \sum_{\vec{s}} q^{(n)}(\vec{s}) \frac{\partial}{\partial\sigma^2}\{\sum_d \log(\text{Beta}(y_d^{(n)}|\vec{s},\Theta))\} \tag{A.29}$$

$$\begin{aligned}
&= \sum_{n,\vec{s},d} q^{(n)}(\vec{s}) \frac{\partial}{\partial\sigma^2}\Big\{ \big(\frac{(1-\bar{M}_d)\bar{M}_d^2}{\sigma^2} - \bar{M}_d - 1\big)\log(y_d^{(n)}) - \log\big(\Gamma(\frac{(1-\bar{M}_d)\bar{M}_d^2}{\sigma^2} - \bar{M}_d)\big) \\
&\quad + \big(\frac{(1-\bar{M}_d)^2\bar{M}_d}{\sigma^2} - 2 + \bar{M}_d\big)\log(1-y_d^{(n)}) - \log\big(\Gamma(\frac{(1-\bar{M}_d)^2\bar{M}_d}{\sigma^2} - 1 + \bar{M}_d)\big) \\
&\quad + \log\big(\Gamma(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2} - 1)\big)\Big\} \\
&= \sum_{n,\vec{s},d} q^{(n)}(\vec{s}) \frac{-1}{\sigma^4}\Big\{ (1-\bar{M}_d)\bar{M}_d^2 \log(y_d^{(n)}) - (1-\bar{M}_d)\bar{M}_d^2\psi(\frac{(1-\bar{M}_d)\bar{M}_d^2}{\sigma^2} - \bar{M}_d) \\
&\quad + (1-\bar{M}_d)^2\bar{M}_d \log(1-y_d^{(n)}) - (1-\bar{M}_d)^2\bar{M}_d\psi(\frac{(1-\bar{M}_d)^2\bar{M}_d}{\sigma^2} - 1 + \bar{M}_d) \\
&\quad + (1-\bar{M}_d)\bar{M}_d\psi(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2} - 1)\Big\} \\
&\stackrel{!}{=} 0.
\end{aligned}$$

Note that $\frac{d}{dx}\log(\Gamma(f(x))) = \big(\frac{d}{dx}f(x)\big)\psi(f(x))$. Now, multiplying the above equation by $-\sigma^4$ (we have $\sigma^2 \neq 0$) and using the approximation $\psi(x) \approx \log(x) - \frac{1}{2x}(1 + \frac{1}{6x})$, we

obtain:

$$\sum_{n,\vec{s},d} q^{(n)}(\vec{s})(1-\bar{M}_d)\bar{M}_d\Big\{\bar{M}_d\log(y_d^{(n)})+(1-\bar{M}_d)\log(1-y_d^{(n)}) \tag{A.30}$$
$$-\bar{M}_d\log(\frac{(1-\bar{M}_d)\bar{M}_d^2}{\sigma^2}-\bar{M}_d)+\frac{\bar{M}_d\,\sigma^2}{2\bar{M}_d\big((1-\bar{M}_d)\bar{M}_d-\sigma^2\big)}\Big(1+\frac{\sigma^2}{6\bar{M}_d\big((1-\bar{M}_d)\bar{M}_d-\sigma^2\big)}\Big)$$
$$-(1-\bar{M}_d)\log(\frac{(1-\bar{M}_d)^2\bar{M}_d}{\sigma^2}-1+\bar{M}_d)+\log(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2}-1)$$
$$+\frac{(1-\bar{M}_d)\sigma^2}{2(1-\bar{M}_d)\big((1-\bar{M}_d)\bar{M}_d-\sigma^2\big)}\Big(1+\frac{\sigma^2}{6(1-\bar{M}_d)\big((1-\bar{M}_d)\bar{M}_d-\sigma^2\big)}\Big)$$
$$-\frac{\sigma^2}{2\big((1-\bar{M}_d)\bar{M}_d-\sigma^2\big)}\Big(1+\frac{\sigma^2}{6\big((1-\bar{M}_d)\bar{M}_d-\sigma^2\big)}\Big)\Big\}\overset{!}{\approx}0.$$

For the above equation, we can further apply the following simplifications:

$$\begin{aligned}\bar{M}_d\log(\frac{(1-\bar{M}_d)\bar{M}_d^2}{\sigma^2}-\bar{M}_d)&=\bar{M}_d\log\big(\bar{M}_d(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2}-1)\big)\\&=\bar{M}_d\Big(\log(\bar{M}_d)+\log(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2}-1)\Big)\end{aligned}$$

and

$$\begin{aligned}(1-\bar{M}_d)\log(\frac{(1-\bar{M}_d)^2\bar{M}_d}{\sigma^2}-1+\bar{M}_d)&=(1-\bar{M}_d)\log\big((1-\bar{M}_d)(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2}-1)\big)\\&=(1-\bar{M}_d)\Big(\log(1-\bar{M}_d)+\log(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2}-1)\Big).\end{aligned}$$

Now, let

$$\begin{aligned}A:=&-\bar{M}_d\log(\frac{(1-\bar{M}_d)\bar{M}_d^2}{\sigma^2}-\bar{M}_d)+\log(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2}-1)\\&-(1-\bar{M}_d)\log(\frac{(1-\bar{M}_d)^2\bar{M}_d}{\sigma^2}-1+\bar{M}_d)\end{aligned}$$

therefore

$$\begin{aligned}A&=-\bar{M}_d\Big(\log(\bar{M}_d)+\log(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2}-1)\Big)+\log(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2}-1)\\&\quad-(1-\bar{M}_d)\Big(\log(1-\bar{M}_d)+\log(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2}-1)\Big)\\&=-\bar{M}_d\log(\bar{M}_d)-\bar{M}_d\log(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2}-1)+\log(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2}-1)\\&\quad-(1-\bar{M}_d)\log(1-\bar{M}_d)-(1-\bar{M}_d)\log(\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2}-1)\\&=-\bar{M}_d\log(\bar{M}_d)-(1-\bar{M}_d)\log(1-\bar{M}_d).\end{aligned}$$

Further, defining $f(\bar{M}_d) := (1-\bar{M}_d)\bar{M}_d$ and substituting A into the Equation (A.30), we obtain:

$$\sum_{n,\vec{s},d} q^{(n)}(\vec{s}) f(\bar{M}_d) \Big\{ \bar{M}_d \big(\log(y_d^{(n)}) - \log(\bar{M}_d) \big) + (1-\bar{M}_d) \big(\log(1-y_d^{(n)}) - \log(1-\bar{M}_d) \big) + \frac{\sigma^2}{2(f(\bar{M}_d)-\sigma^2)} \big(1 + \frac{\sigma^2}{6\bar{M}_d(f(\bar{M}_d)-\sigma^2)}\big) + \frac{\sigma^2}{2(f(\bar{M}_d)-\sigma^2)} \big(1 + \frac{\sigma^2}{6(1-\bar{M}_d)(f(\bar{M}_d)-\sigma^2)}\big) - \frac{\sigma^2}{2(f(\bar{M}_d)-\sigma^2)} \big(1 + \frac{\sigma^2}{6(f(\bar{M}_d)-\sigma^2)}\big) \Big\} \overset{!}{\approx} 0. \tag{A.31}$$

We can still simplify the above equation and obtain a more concise relation. To this, we assume:

$$B := \frac{\sigma^2}{2(f(\bar{M}_d)-\sigma^2)} \big(1 + \frac{\sigma^2}{6\bar{M}_d(f(\bar{M}_d)-\sigma^2)}\big) + \frac{\sigma^2}{2(f(\bar{M}_d)-\sigma^2)} \big(1 + \frac{\sigma^2}{6(1-\bar{M}_d)(f(\bar{M}_d)-\sigma^2)}\big) - \frac{\sigma^2}{2(f(\bar{M}_d)-\sigma^2)} \big(1 + \frac{\sigma^2}{6(f(\bar{M}_d)-\sigma^2)}\big)$$

and therefore

$$\begin{aligned} B &= \frac{\sigma^2}{2(f(\bar{M}_d)-\sigma^2)} \Big(1 + \frac{\sigma^2}{6\bar{M}_d(f(\bar{M}_d)-\sigma^2)} + \frac{\sigma^2}{6(1-\bar{M}_d)(f(\bar{M}_d)-\sigma^2)} - \frac{\sigma^2}{6(f(\bar{M}_d)-\sigma^2)}\Big) \\ &= \frac{\sigma^2}{2(f(\bar{M}_d)-\sigma^2)} \Big(1 + \frac{\sigma^2(1-\bar{M}_d) + \sigma^2\bar{M}_d - \sigma^2 f(\bar{M}_d)}{6f(\bar{M}_d)(f(\bar{M}_d)-\sigma^2)}\Big) \\ &= \frac{\sigma^2}{2(f(\bar{M}_d)-\sigma^2)} \Big(1 + \frac{\sigma^2\big(1-f(\bar{M}_d)\big)}{6f(\bar{M}_d)(f(\bar{M}_d)-\sigma^2)}\Big). \end{aligned}$$

Next, we define:

$$\begin{aligned} k(y_d^{(n)}, \vec{s}, \Theta) &:= \bar{M}_d \big(\log(y_d^{(n)}) - \log(\bar{M}_d) \big) + (1-\bar{M}_d) \big(\log(1-y_d^{(n)}) - \log(1-\bar{M}_d) \big) \\ l(y_d^{(n)}, \vec{s}, \Theta) &:= \frac{-1}{\big(f(\bar{M}_d)-\sigma^2\big)} \Big(1 + \frac{\sigma^2(1-f(\bar{M}_d))}{6f(\bar{M}_d)(f(\bar{M}_d)-\sigma^2)}\Big). \end{aligned}$$

Hence, by substituting the above definitions and also the value of B into the Equation (A.31), we can conclude:

$$\sum_{n,\vec{s},d} q^{(n)}(\vec{s}) f(\bar{M}_d) k(y_d^{(n)}, \vec{s}, \Theta) \approx \frac{\sigma^2}{2} \sum_{n,\vec{s},d} q^{(n)}(\vec{s}) f(\bar{M}_d) l(y_d^{(n)}, \vec{s}, \Theta) \tag{A.32}$$

and consequently the update equation of the variance parameter σ^2 can be presented by:

$$\big(\sigma^2\big)^{\text{new}} \approx \frac{2\sum_{n,d} \langle f(\bar{M}_d) k(y_d^{(n)}, \vec{s}, \Theta^{\text{old}}) \rangle_{q^{(n)}}}{\sum_{n,d} \langle f(\bar{M}_d) l(y_d^{(n)}, \vec{s}, \Theta^{\text{old}}) \rangle_{q^{(n)}}}. \tag{A.33}$$

Note that we here update the variance parameter in a fixed-point sense as the right hand side of the above equation also depends on σ^2. Moreover, we can either perform the above equation once in each M-step or several times (i.e. to continue the computation above in an

iterative procedure in each M-step till convergence at one point). We here used the former approach and (similar to the other updates) only performed the equation once in each EM algorithm. After a few EM iterations and the convergence of parameters Θ, we considered the equilibrium point of the above equation as the optimal scalar variance parameter. We observed that this approach provides a reliable method to estimate the variance σ^2.

In addition, observe that we here approximated the Digamma function with its first three terms. Such an estimation can also be improved (similar to the Gamma-MCA model) by considering a better approximation of the Digamma function. Finally, we emphasize that the parameters of the Beta distribution should be checked after each M-step to avoid computing unacceptable values and possibly numerical errors. Importantly, for the Beta distribution, we know that $\eta_1, \eta_2 > 0$. Therefore, given the equations in (3.20), we should always have:

$$\frac{(1-\bar{M}_d)\bar{M}_d^2}{\sigma^2} - \bar{M}_d > 0 \quad \Longrightarrow \quad \frac{(1-\bar{M}_d)\bar{M}_d^2}{\sigma^2} > \bar{M}_d$$

and

$$\frac{(1-\bar{M}_d)^2\bar{M}_d}{\sigma^2} - 1 + \bar{M}_d > 0 \quad \Longrightarrow \quad \frac{(1-\bar{M}_d)^2\bar{M}_d}{\sigma^2} > 1 - \bar{M}_d.$$

On the other hand, we always have $\bar{M}_d \in (0,1)$ which yields $\bar{M}_d > 0$ and $1 - \bar{M}_d > 0$. Hence, the above inequalities can be restated as the following:

$$\frac{(1-\bar{M}_d)\bar{M}_d}{\sigma^2} > 1 \quad \Longrightarrow \quad (1-\bar{M}_d)\bar{M}_d = f(\bar{M}_d) > \sigma^2.$$

The constraint $\sigma^2 < f(\bar{M}_d)$ ensures that we always obtain an acceptable variance parameter given the current matrix $M(\Theta)$. Thus, after each computation of (A.33) in the M-step, we enforce the above constraint and manually equate the value of σ^2 to $f(\bar{M}_d) - 10^{-4}$ in case that the constraint does not satisfy.

A.4 M-step Update Equations of the MCA model

Consider the MCA model presented in Section 2.3. Similar to the ef-MCA models presented in this study, we can also apply the EM algorithm and perform both E- and M-steps to train the model using a set of given data points. As mentioned before (see Section 4.1.1), the E-step is to equate the variational distributions $q^{(n)}(\vec{s})$ to the posterior values $p(\vec{s}\,|\vec{y}^{(n)}, \Theta)$. The M-step is then to update parameters Θ of the model. These update equations can be similarly obtained by equating the derivatives of the free energy w.r.t. the model parameters to zero and solving for the desired parameters. The resultant updates are then reported here for completeness (we refer the readers to, e.g., Puertas, Bornschein, and Lücke, 2010; Bornschein, Henniges, and Lücke, 2013 for more details):

$$\pi^{\text{new}} = \frac{1}{N}\sum_n \langle |\vec{s}| \rangle_{q^{(n)}} \tag{A.34}$$

$$M_{dh}^{\text{new}} = \frac{\sum_n \langle \mathcal{A}_{dh}(\vec{s}, \Theta^{\text{old}}) \rangle_{q^{(n)}} y_d^{(n)}}{\sum_n \langle \mathcal{A}_{dh}(\vec{s}, \Theta^{\text{old}}) \rangle_{q^{(n)}}} \tag{A.35}$$

$$(\sigma^2)^{\text{new}} = \frac{1}{ND}\sum_n \langle \|\vec{y}^{(n)} - \bar{M}(\vec{s}, \Theta^{\text{old}})\|^2 \rangle_{q^{(n)}} \tag{A.36}$$

where $|\vec{s}| = \sum_h s_h$. It should be also mentioned that we here defined the index matrix $\mathcal{A}_{dh}(\vec{s},\Theta)$ as is given by (4.14), but the original studies of (Puertas, Bornschein, and Lücke, 2010; Bornschein, Henniges, and Lücke, 2013) used a smooth approximation of the maximum function and consequently a different index matrix $\mathcal{A}_{dh}(\vec{s},\Theta)$. This approximation is defined because of the non-differentiability of the maximum function $\max_h\{M_{dh}s_h\}$ and, for a large and odd positive value of ρ, can be formally stated as follows:

$$\bar{M}_d^\rho(\vec{s},\Theta) = \left(\sum_{h=1}^{H}(M_{dh}s_h)^\rho\right)^{\frac{1}{\rho}} \tag{A.37}$$

where

$$\lim_{\substack{n\longrightarrow\infty\\ \rho=2n+1}} \bar{M}_d^\rho(\vec{s},\Theta) = \bar{M}_d(\vec{s},\Theta) = \max_h\{M_{dh}s_h\}. \tag{A.38}$$

As a consequence, they defined:

$$\mathcal{A}_{dh}^\rho(\vec{s},\Theta) := \frac{\partial}{\partial M_{dh}}\bar{M}_d^\rho(\vec{s},\Theta) \tag{A.39}$$

and replaced the term $\mathcal{A}_{dh}(\vec{s},\Theta)$ in (A.35) by the above definition. Moreover, as it can be observed from (A.35), the update equations for the weights M_{dh} do not allow a closed-form solution (similar to components W_{dh} and V_{dh} in Theorem 2). We thus employ a fixed-point equation whose fixed point is the exact solution of the maximization step. We exploit the fact that in practice one single evaluation of (A.35) is enough to (noisily, not optimally) move towards convergence.
We should also mention that the MCA model is the same as the proposed Gaussian-MCA model here when a global variance σ^2 is used with a prior parameter π for each cause h.

A.5 M-step Update Equations of the Noisy-OR model

Consider the Noisy-OR model presented in Section 2.4. Likewise, the M-step equations of this probabilistic model can be obtained similar to the MCA approach. The resultant equations are then as follows:

$$\pi_h^{\text{new}} = \frac{1}{N}\sum_n \langle s_h\rangle_{q^{(n)}} \tag{A.40}$$

$$W_{dh}^{\text{new}} = 1 + \frac{\sum_n (y_d^{(n)} - 1)\langle D_{dh}(\vec{s})\rangle_{q^{(n)}}}{\sum_n \langle C_{dh}(\vec{s})\rangle_{q^{(n)}}} \tag{A.41}$$

where

$$D_{dh}(\vec{s}) = \frac{\widetilde{W}_{dh}(\vec{s})s_h}{N_d(\vec{s})(1-N_d(\vec{s}))} \quad \text{and} \quad C_{dh}(\vec{s}) = \widetilde{W}_{dh}(\vec{s})D_{dh}(\vec{s}) \tag{A.42}$$

such that

$$\widetilde{W}_{dh}(\vec{s}) = \prod_{h'\neq h}(1 - W_{dh'}s_{h'}). \tag{A.43}$$

Similar to the MCA, we updated the weights W_{dh} in a fixed-point sense and computed the equilibrium point as the optimal solution (we performed the updates only once in each M-step).

Appendix B

Additional Details on Experimental Results

Finally, we present further details of our experimental results carried out in this study. This includes additional details or figures for the feature extraction, denoising and analysis of medical data. We refer the readers to Section 5.2 for a complete discussion for each of these experiments. Moreover, it should be stated that the content of this chapter is taken from (Mousavi et al., 2020) (currently under review) and (Mousavi et al., 2021) where Jakob Drefs performed the experiments with the Gaussian-MCA model and produced Figure B.1. Other figures were produced by me in consultation with Jörg Lücke.

B.1 Natural Image Patches

The complete set of learned dictionaries obtained by applying Gaussian-MCA on ZCA-whitened natural image patches is presented in Figure B.1. The Figure depicts the amount of $H = 1,000$ dictionaries (for both component means and variances) obtained by using a double-dictionary approach. As it can be seen, a variety of familiar Gabor-like and globular fields can be observed (component means) alongside the frequency of their appearances (component variances). The results consequently illustrate the applicability of the proposed approach in extracting rich information from a set of real datasets, and in addition, the scalability of the approach (to a high value of $H = 1,000$).

Next, Figure B.2 depicts the results of the Beta-MCA model applied on rescaled natural image patches. Similar to Figure 5.12, the current experiment illustrates a consistent behaviour of the Beta-MCA model (the settings here are different from the one in Section 5.2.2; in the previous experiment we used $D = 8 \times 8$ and $H = 100$ and the current experiment presents the results for $D = 8 \times 8$ and $H = 200$). As observed, the variety of learned generative fields and their frequencies is similar to the previous experiment. Likewise, the double-dictionary approach provides more elongated and edge-like functions rather than the single-dictionary approach and in addition, the frequency of learned globular fields is higher for the single-dictionary approach. Also, the double-dictionary approach achieves a higher free energy value rather than the other model which supports our claim that learning component variances per cause and per latent enables the model to extract richer information from the data.
The detailed settings of these two experiments are described in Section 5.2.2.

B.2 Poisson Denoising

We further investigated the Poisson denoising experiment and considered 8 different images presented in Figure B.3. We then assumed 6 different peak values and compared the performance of the proposed P-MCA model with the NLSPCA and SPDA approaches presented in (Salmon et al., 2014) and (Giryes and Elad, 2014), respectively. These two are the only

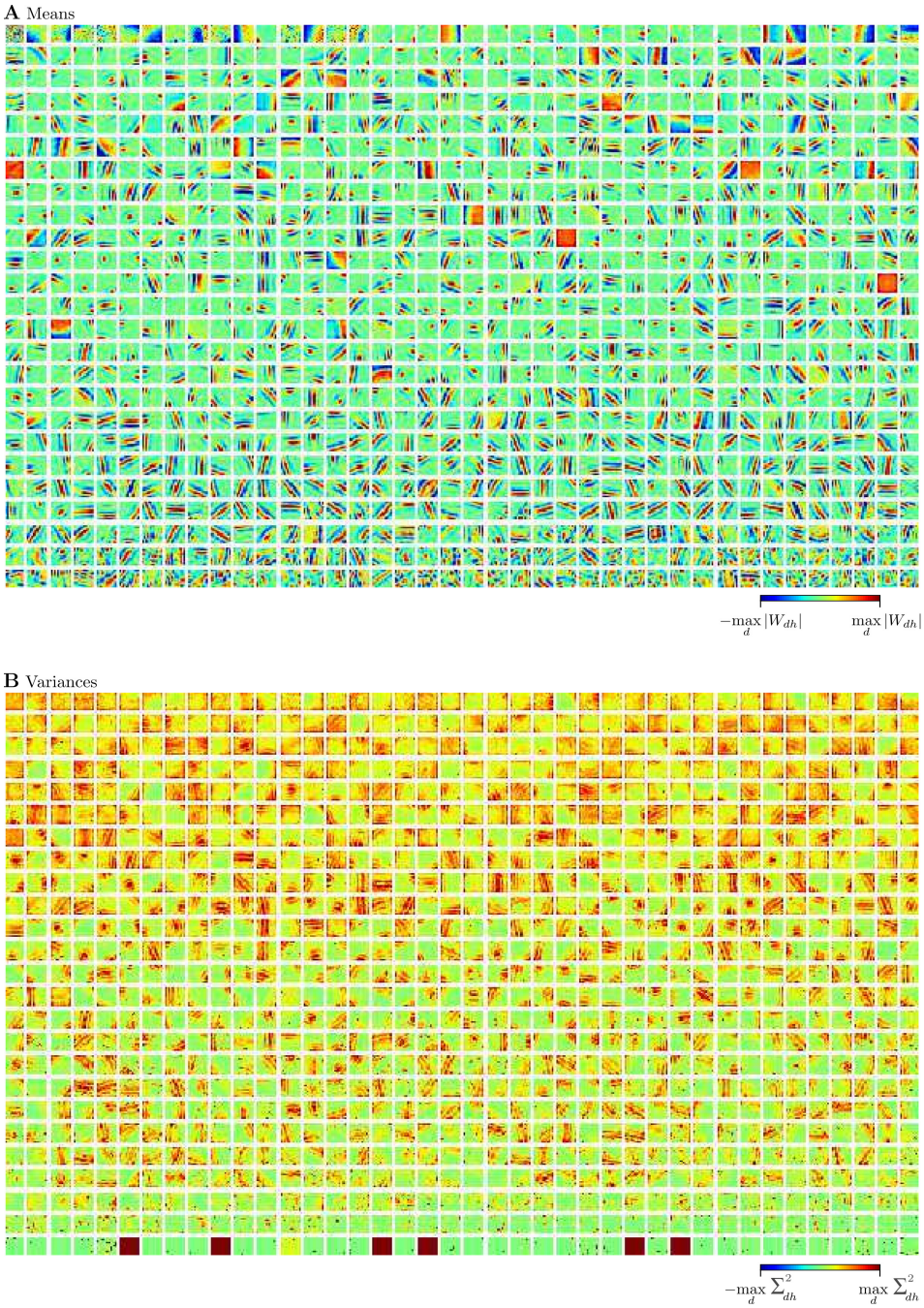


FIGURE B.1: Complete dictionaries with $H = 1,000$ component means and component variances learned from whitened natural image patches using a Gaussian-MCA model (compare Section 5.2.2). The generative fields are ordered according to their activations, starting with the fields corresponding to the most active hidden units.

models (in comparison to the other approaches used in Section 5.2.4) that are based on the assumption of Poisson distribution and therefore, provide a fair comparison. Moreover, for the experiments here, we trained the P-MCA model with $D = 20 \times 20$ (similar to the settings of SPDA in (Giryes and Elad, 2014)) and performed 100 EEM iterations. Regarding the H values (number of dictionaries), we used either 30 or 100 for different peak values. All the other settings of P-MCA were assumed to be equivalent to the experiments in Section 5.2.4.

Table B.1 then represents a comparison of the PSNR values for the P-MCA, NLSPCA

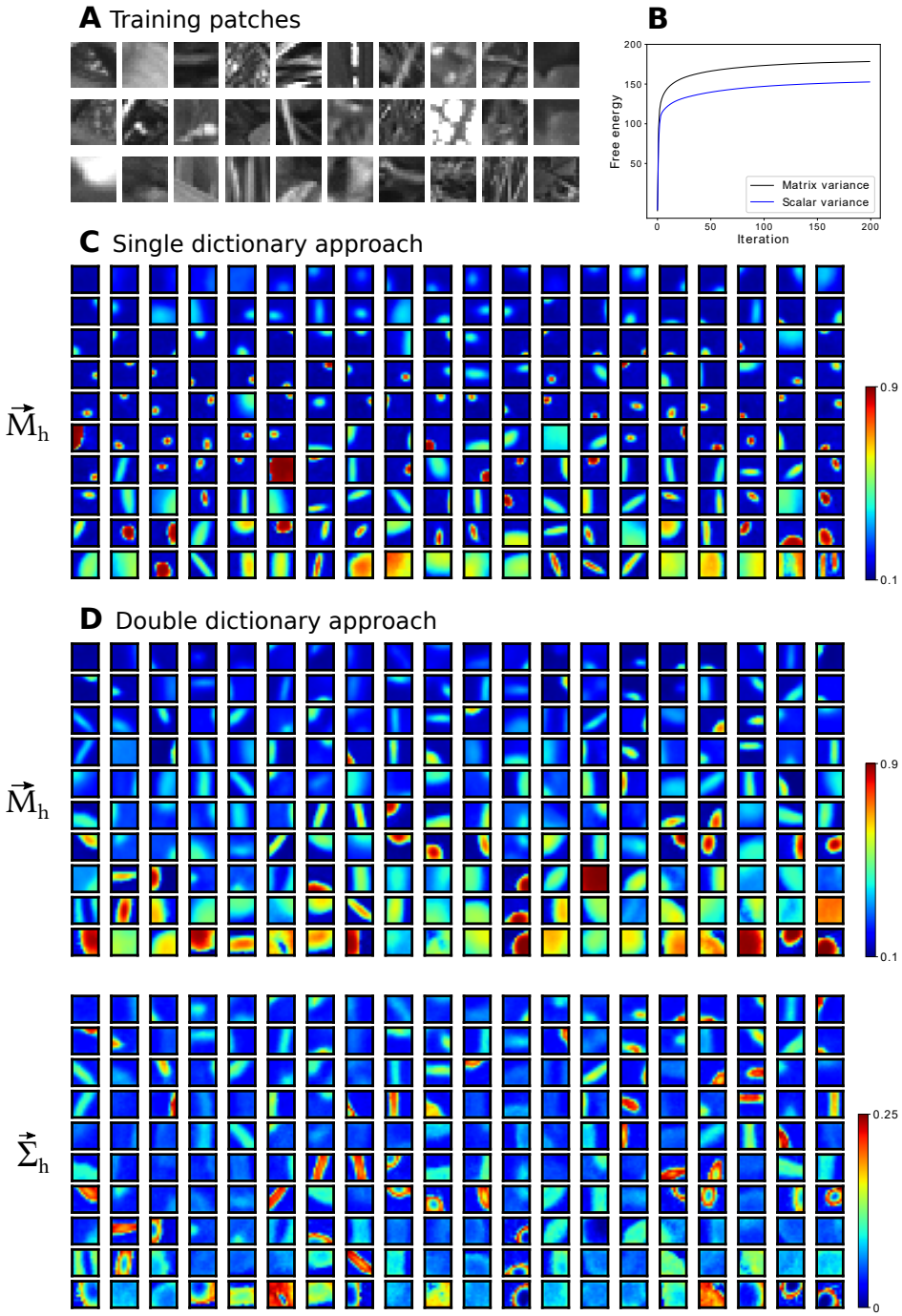


FIGURE B.2: Feature extraction using Beta-MCA with $D = 8 \times 8$ and $H = 200$. **A** 30 training patches. **B** Evolution of the free energy function for both single and double-dictionary approaches. Here, only the first 200 iterations are depicted. **C**-**D** Parameters $\vec{M}_h$ and $\vec{\Sigma}_h$ of the Beta-MCA generative model trained with (C) a single-dictionary (with scalar variance σ^2) and (D) double dictionaries. The $H = 200$ generative fields are sorted from left to right and from top to down based on their prior values, π_h. The top left corresponds to the background which has a prior parameter close to 1. In addition, each of the variance components in panel D-bottom corresponds to one of the component means in panel D-top. Also, in all cases, we linearly scaled the learned values to fill the corresponding color range in the gray (color) space.

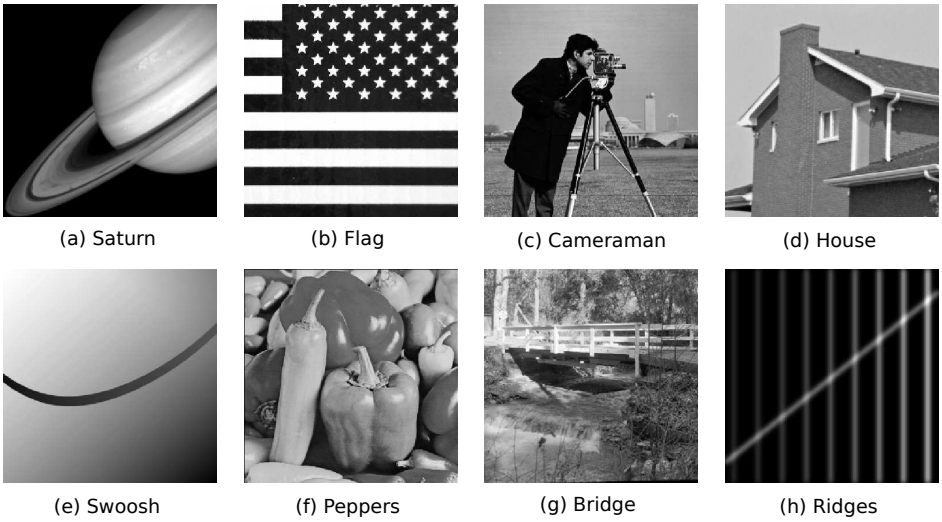


FIGURE B.3: Images used for the denoising experiments.

and SPDA models. For P-MCA, we used $H = 100$ for the peak values of 1, 2 and 4 and $H = 30$ for the other peak values and also the values of NLSPCA and SPDA are taken from (Giryes and Elad, 2014). As observed, the proposed model produces competitive results in comparison to the two models which are highly optimized for the denoising task. For instance, SPDA uses a sophisticated initialization approach where the model is first trained on images with higher peak values and the learned dictionaries are then used as initialization of the model. On the contrary, we used random initializations for the ef-MCA data models proposed here. In addition, no annealing method is applied here to avoid local optima. Intriguingly, the P-MCA model achieves (almost) the best results amongst the three algorithms for very low peak values. Therefore, it can be deduced that a model with a non-linear link function can better describe the statistical features of images corrupted by a very high amount of noise (at low SNR values). Specifically, the maximum superposition used here may seem to be superior as it is better suited for the occlusion-like non-linearities. Such an effect has been reported before for acoustic data (Roweis, 2003; Lücke and Sahani, 2008; Sheikh et al., 2019) and for natural images patches (Bornschein, Henniges, and Lücke, 2013), but the current experiments also confirm the results for denoising of images corrupted by Poisson noise.

In addition, given some random experiments, we observed that increasing the value of D (the number of image patches) will increase the performance of the P-MCA model which may be perceived as trivial. A similar effect, however, cannot be (in general) observed for the H values (number of dictionaries). In fact, it can be seen that higher number of dictionaries yield a better performance for the peak values of 1, 2 and 4, and the opposite is true for the low peak values of 0.1, 0.2 and 0.5 This effect may be attributed to the high amount of noise where a model with fewer number of dictionaries may have an easier task to restore the original image.

Finally, we provide the details for using the VST+BM3D model described in Section 5.2.4. For this model, one can use the Anscombe transformation given by:

$$f : z \longrightarrow 2\sqrt{z + \frac{3}{8}} \tag{B.1}$$

and then perform the BM3D algorithm with unit variance. The estimated, non-noisy image is then transformed back using the inverse transformation given by (see, e.g., Anscombe, 1948;

TABLE B.1: Comparison of the PSNR values (in terms of dB) for the considered Poisson denoising benchmarks. Values of NLSPCA and SPDA are taken from (Giryes and Elad, 2014). Here we report the results for six different peak values and the bold number in each subgroup denotes the best PSNR value in comparison to the other models. Also, we used $D = 20 \times 20$, and considered $H = 30$ for peak values $0.1, 0.2$ and 0.5 and also $H = 100$ for peak values $1, 2$ and 4. All values are average over five different noise realizations. In general, we observed the P-MCA to perform very well in comparison to the other two models given the fact that no extra pre- or post-processing method is used here to improve the results.

Method	Peak	Saturn	Flag	Camera	House	Swoosh	Peppers	Bridge	Rdiges	Average
NLSPCA	0.1	20.86	14.42	16.41	17.81	19.11	**16.24**	**16.59**	20.92	17.80
SPDA		17.40	13.35	14.36	14.84	15.12	14.28	14.60	19.86	15.48
P-MCA		**20.90**	**14.69**	**17.42**	**18.00**	**19.43**	**16.24**	16.57	**22.89**	**18.26**
NLSPCA	0.2	22.90	16.48	17.79	18.91	21.10	17.45	17.46	24.22	19.54
SPDA		21.52	16.58	16.93	17.83	18.91	16.75	16.80	23.25	18.57
P-MCA		**23.30**	**17.09**	**18.48**	**19.36**	**21.62**	**17.46**	**17.51**	24.79	**19.95**
NLSPCA	0.5	24.91	18.80	19.23	20.85	23.80	18.78	**18.50**	28.20	21.63
SPDA		**25.50**	**19.67**	18.90	20.51	24.21	18.66	18.46	27.76	21.71
P-MCA		25.48	19.06	**19.67**	**21.86**	**24.38**	**18.94**	18.47	28.51	**22.04**
NLSPCA	1	26.89	20.26	20.32	22.09	**27.42**	19.62	18.94	30.57	23.26
SPDA		27.02	**22.54**	20.23	22.73	26.28	19.99	19.20	30.93	23.61
P-MCA		**27.46**	21.28	**20.54**	**23.42**	26.85	**20.35**	**19.34**	30.55	**23.72**
NLSPCA	2	28.22	20.86	20.76	23.86	**29.62**	20.52	19.47	31.87	24.40
SPDA		**29.38**	**24.92**	**21.54**	**25.09**	29.27	21.23	**20.15**	33.40	**25.62**
P-MCA		28.64	21.41	21.16	24.70	28.72	**21.44**	20.11	32.25	24.80
NLSPCA	4	29.44	21.25	21.09	24.89	31.30	21.12	20.16	34.01	25.41
SPDA		**31.04**	**26.27**	21.90	**26.09**	**33.20**	22.09	20.55	36.05	**27.15**
P-MCA		30.14	22.89	**21.96**	25.83	30.82	**22.64**	**20.77**	33.56	26.07

Azzari and Foi, 2016 for further details):

$$f^{-1} : x \longrightarrow (\frac{x}{2})^2 - \frac{3}{8}. \tag{B.2}$$

B.3 Continuous Interval Disease Profiles for the CAFPAs

Finally, we present the symptom statistics (continuous disease profiles) of the two diseases analysed in Section 5.2.1 together with the normal hearing patients. To obtain the disease profiles, we fitted a Beta and a Gaussian distribution to the CAFPAs where only high-frequency hearing loss or broadband hearing loss or none of the two diseases are active. The results are then presented in Figures B.4, B.5 and B.6, respectively.

From the figures, it can be inferred that the Beta distribution is a better fit to such an interval data (this is further in line with the results presented in Buhl et al., 2020), which is also reflected on the log-likelihood values obtained by the two models (we consistently observed that the log-likelihood values obtained by fitting a Beta distribution to the data is higher than the log-likelihood values obtained by the Gaussian). This supports our claim here that a sophisticated generative model based on the Beta distribution enables us to obtain richer information from interval data.

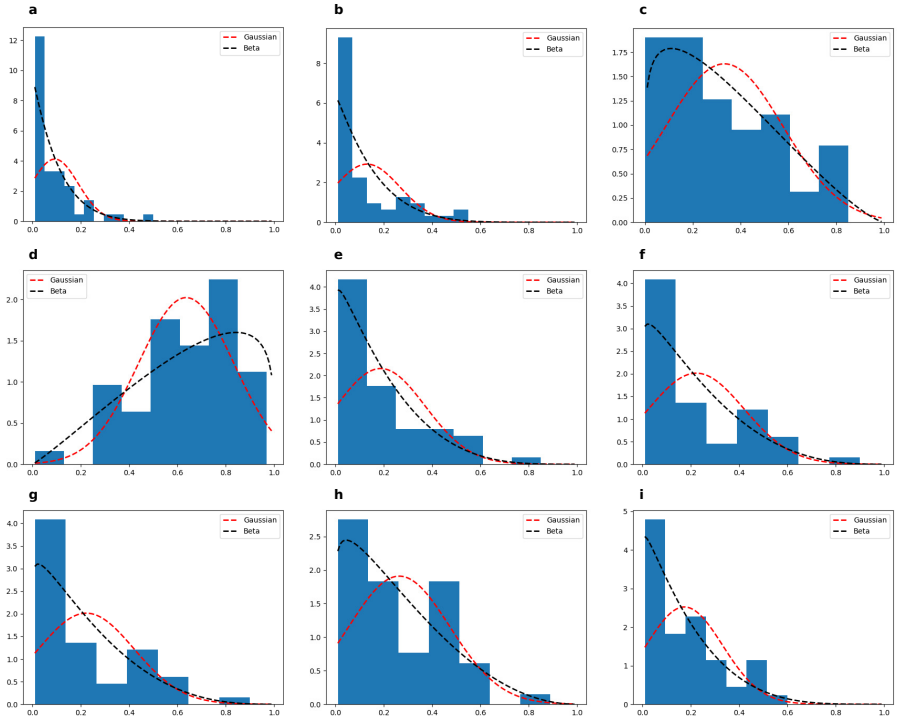


FIGURE B.4: Continuous disease profile corresponding to high-frequency hearing loss. (**a**-**i**) denote values for 9 different CAFPAs (see Buhl et al., 2020 for details) where only the high-frequency hearing loss is active. We here fitted Beta (black dotted line) and Gaussian (red dotted line) distributions to these CAFPAs and the obtained log-likelihood ($\mathcal{L}(\Theta)/N$) values are -19.058 and -67.660 for the Beta and Gaussian distributions, respectively.

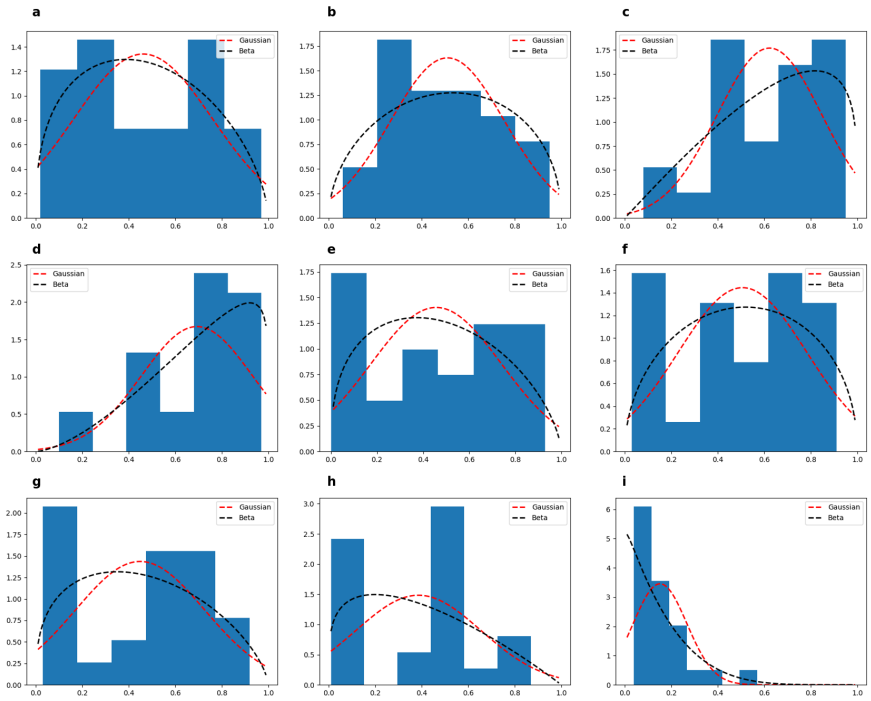


FIGURE B.5: Continuous disease profile corresponding to broadband hearing loss. (**a**-**i**) denote values for 9 different CAFPAs (see Buhl et al., 2020 for details) where only the broadband hearing loss is active. We here fitted Beta (black dotted line) and Gaussian (red dotted line) distributions to these CAFPAs and the obtained log-likelihood ($\mathcal{L}(\Theta)/N$) values are -35.917 and -73.178 for the Beta and Gaussian distributions, respectively.

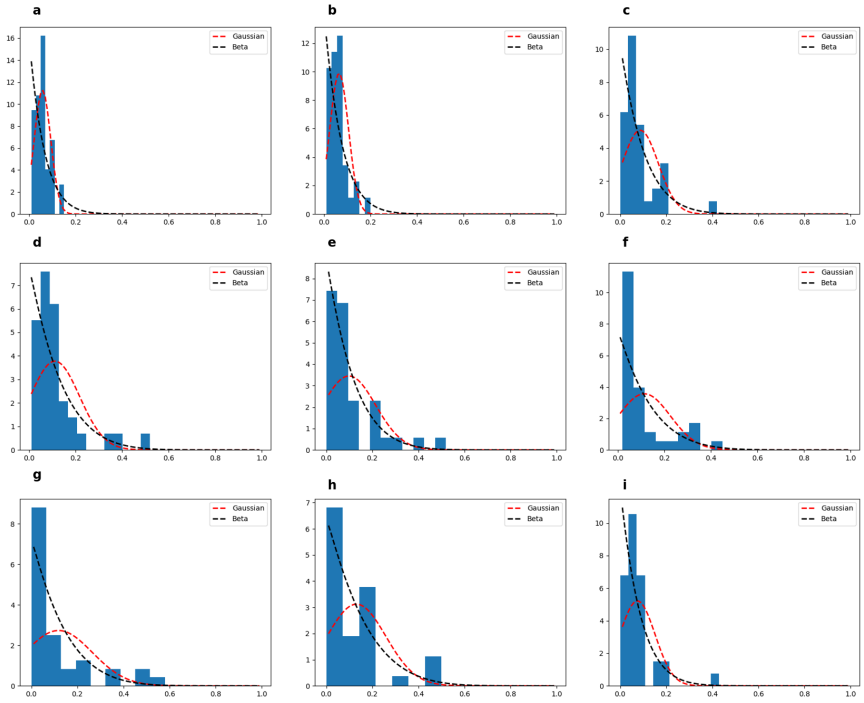


FIGURE B.6: Continuous disease profile corresponding to normal hearing patients. (**a**-**i**) denote values for 9 different CAFPAs (see Buhl et al., 2020 for details) for the patients with no serious hearing impairment. We here fitted Beta (black dotted line) and Gaussian (red dotted line) distributions to these CAFPAs and the obtained log-likelihood ($\mathcal{L}(\Theta)/N$) values are 103.104 and 17.889 for the Beta and Gaussian distributions, respectively.

Bibliography

Abramowitz, Milton, Irene A Stegun, et al. (1972). *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Vol. 55. Washington, DC: National bureau of standards.

Aharon, Michal, Michael Elad, and Alfred Bruckstein (2006). "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation". In: *IEEE Transactions on Signal Processing* 54.11, pp. 4311–4322.

Anscombe, Francis J. (1948). "The transformation of Poisson, Binomial and Negative-Binomial Data". In: *Biometrika* 35.3/4, pp. 246–254.

Arora, Sanjeev et al. (2017). "Provable learning of noisy-OR networks". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, pp. 1057–1066.

Azzari, Lucio and Alessandro Foi (2016). "Variance Stabilization for Noisy+Estimate Combination in Iterative Poisson Denoising". In: *IEEE Signal Processing Letters* 23.8, pp. 1086–1090.

Baraniuk, Richard G. (2007). "Compressive Sensing". In: *IEEE Signal Processing Magazine* 24.4.

Bengio, Yoshua, Nicholas Léonard, and Aaron Courville (2013). "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation". In: *arXiv preprint arXiv:1308.3432*.

Berkes, Pietro, Ben White, and József Fiser (2009). "No evidence for active sparsification in the visual cortex". In: *Advances in Neural Information Processing Systems* 22, pp. 108–116.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Bollen, Kenneth A. (2002). "Latent Variables in Psychology and the Social Sciences". In: *Annual Review of Psychology* 53.1, pp. 605–634.

Booth, James G. and James P. Hobert (1999). "Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.1, pp. 265–285.

Bornschein, Jörg, Marc Henniges, and Jörg Lücke (2013). "Are V1 Simple Cells Optimized for Visual Occlusions? A Comparative Study". In: *PLOS Computational Biology* 9.6, e1003062.

Bornschein, Jörg and Jörg Lücke (2009). "Applications of Non-linear Component Extraction to Spectrogram Representations of Auditory Data". In: *Proceedings of BCCN, Frontiers in Computational Neuroscience*.

Bowman, K. O. and L. R. Shenton (2014). "Estimation: Method of Moments". In: *Wiley StatsRef: Statistics Reference Online*.

Brémaud, Pierre (2012). *An Introduction to Probabilistic Modeling*. Springer Science & Business Media.

Buhl, Mareike et al. (2019). "Common Audiological Functional Parameters (CAFPAs): statistical and compact representation of rehabilitative audiological classification based on expert knowledge". In: *International Journal of Audiology* 58.4, pp. 231–245.

Buhl, Mareike et al. (2020). "Common Audiological Functional Parameters (CAFPAs) for single patient cases: deriving statistical models from an expert-labelled data set". In: *International Journal of Audiology*, pp. 1–14.

Bui, Thanh Minh et al. (2015). "Level-set segmentation of 2D and 3D ultrasound data using local gamma distribution fitting energy". In: *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. IEEE, pp. 1110–1113.

Burger, Harold C., Christian J. Schuler, and Stefan Harmeling (2012). "Image denoising: Can plain Neural Networks compete with BM3D?" In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2392–2399.

Cai, Qiong et al. (2019). "A Survey on Multimodal Data-Driven Smart Healthcare Systems: Approaches and Applications". In: *IEEE Access* 7, pp. 133583–133599.

Cattell, Raymond (2012). *The Scientific Use of Factor Analysis in Behavioral and Life Sciences*. Springer Science & Business Media.

Celeux, Gilles and Gérard Govaert (1992). "A classification EM algorithm for clustering and two stochastic versions". In: *Computational Statistics & Data Analysis* 14.3, pp. 315–332.

Chaudhury, Subhajit and Hiya Roy (2017). "Can fully convolutional networks perform well for general image restoration problems?" In: *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*. IEEE, pp. 254–257.

Chen, Yunjin and Thomas Pock (2017). "Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6, pp. 1256–1272.

Collins, John and Minh Huynh (2014). "Estimation of diagnostic test accuracy without full verification: a review of latent class methods". In: *Statistics in Medicine* 33.24, pp. 4141–4169.

Collins, Michael, Sanjoy Dasgupta, and Robert E. Schapire (2002). "A Generalization of Principal Components Analysis to the Exponential Family". In: *Advances in Neural Information Processing Systems*, pp. 617–624.

Cox, David Roxbee (2006). *Principles of Statistical Inference*. Cambridge University Press.

Dabov, Kostadin et al. (2007). "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering". In: *IEEE Transactions on Image Processing* 16.8, pp. 2080–2095.

Dai, Zhenwen, Georgios Exarchakis, and Jörg Lücke (2013). "What Are the Invariant Occlusive Components of Image Patches? A Probabilistic Generative Approach". In: *Advances in Neural Information Processing Systems*, pp. 243–251.

Dai, Zhenwen and Jörg Lücke (2012). "Unsupervised learning of translation invariant occlusive components". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2400–2407.

— (2014). "Autonomous Document Cleaning – A Generative Approach to Reconstruct Strongly Corrupted Scanned Texts". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.10, pp. 1950–1962.

Dawid, A. Philip (1980). "Conditional Independence for Statistical Operations". In: *The Annals of Statistics*, pp. 598–617.

Dayan, Peter and Laurence F. Abbott (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Computational Neuroscience Series.

DeGuchy, Omar et al. (2019). "Deep Neural Networks for Low-resolution Photon-limited Imaging". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 3247–3251.

Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39, pp. 1–38.

Donoho, David L. (2006). "Compressed Sensing". In: *IEEE Transactions on Information Theory* 52.4, pp. 1289–1306.

Drefs, Jakob, Enrico Guiraud, and Jörg Lücke (2020). "Evolutionary Variational Optimization of Generative Models". In: *arXiv preprint arXiv:2012.12294*.

Enøe, Claes, Marios P. Georgiadis, and Wesley O. Johnson (2000). "Estimation of sensitivity and specificity of diagnostic tests and disease prevalence when the true disease state is unknown". In: *Preventive Veterinary Medicine* 45.1-2, pp. 61–81.

Everitt, Brian (1984). *An Introduction to Latent Variable Models*. Chapman and Hall.

Exarchakis, Georgios and Jörg Lücke (2017). "Discrete Sparse Coding". In: *Neural Computation* 29, pp. 2979–3013.

Farouni, Rick (2017). "A Contemporary Overview of Probabilistic Latent Variable Models". In: *arXiv preprint arXiv:1706.08137*.

Fei, Yang and Wei-qin Li (2017). "Improve artificial neural network for medical analysis, diagnosis and prediction". In: *Journal of Critical Care* 40, p. 293.

Finlayson, Samuel G., Paea LePendu, and Nigam H. Shah (2014). "Building the graph of medicine from millions of clinical narratives". In: *Scientific Data* 1.1, pp. 1–9.

Fisz, Marek (1955). "The limiting distribution of a function of two independent random variables and its statistical application". In: *Colloquium Mathematicum*. Vol. 3, pp. 138–146.

Földiák, Peter (1990). "Forming sparse representations by local anti-Hebbian learning". In: *Biological Cybernetics* 64, pp. 165–170.

Forster, Dennis, Abdul-Saboor Sheikh, and Jörg Lücke (2018). "Neural Simpletrons: Learning in the Limit of Few Labels with Directed Generative Networks". In: *Neural Computation* 30.8, pp. 2113–2174.

Foster, Peter et al. (2015). "Chime-Home: A dataset for sound source recognition in a domestic environment". In: *Proceedings of the 11th Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–5.

Friston, Karl et al. (2007). "Variational free energy and the Laplace approximation". In: *Neuroimage* 34.1, pp. 220–234.

Frolov, Alexander A., Dusan Husek, and Pavel Y. Polyakov (2014). "Two Expectation-Maximization algorithms for Boolean Factor Analysis". In: *Neurocomputing* 130.0, pp. 83–97.

Frolov, Alexander A., Dušan Húsek, and Pavel Yu Polyakov (2015). "Comparison of Seven Methods for Boolean Factor Analysis and Their Evaluation by Information Gain". In: *IEEE Transactions on Neural Networks and Learning Systems* 27.3, pp. 538–550.

Gan, Zhe et al. (2015). "Learning Deep Sigmoid Belief Networks with Data Augmentation". In: *Artificial Intelligence and Statistics*. PMLR, pp. 268–276.

Giryes, Raja and Michael Elad (2014). "Sparsity-Based Poisson Denoising With Dictionary Learning". In: *IEEE Transactions on Image Processing* 23.12, pp. 5057–5069.

Goodfellow, Ian J., Aaron Courville, and Yoshua Bengio (2012a). "Large-Scale Feature Learning With Spike-and-Slab Sparse Coding". In: *arXiv preprint arXiv:1206.6407*.

— (2012b). "Spike-and-Slab Sparse Coding for Unsupervised Feature Discovery". In: *arXiv preprint arXiv:1201.3382*.

Goodfellow, Ian J. et al. (2014). "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., pp. 2672–2680.

Gu, Shuhang et al. (2014). "Weighted Nuclear Norm Minimization with Application to Image Denoising". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2862–2869.

Guiraud, Enrico, Jakob Drefs, and Jörg Lücke (2018). "Evolutionary Expectation Maximization". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, pp. 442–449.

— (2020). "Direct Evolutionary Optimization of Variational Autoencoders With Binary Latents". In: *arXiv preprint arXiv:2011.13704*.

Gulshan, Varun et al. (2016). "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs". In: *Jama* 316.22, pp. 2402–2410.

Haft, Michael, R. Hofman, and Volker Tresp (2004). "Generative binary codes". In: *Formal Pattern Analysis & Applications* 6.4, pp. 269–284.

Hajian-Tilaki, Karimollah (2013). "Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation". In: *Caspian Journal of Internal Medicine* 4.2, p. 627.

Hall, Alastair R. (2005). *Generalized Method of Moments*. Oxford University Press.

Henniges, Marc et al. (2010). "Binary Sparse Coding". In: *International Conference on Latent Variable Analysis and Signal Separation*. Springer, pp. 450–457.

Henniges, Marc et al. (2014). "Efficient Occlusive Components Analysis". In: *Journal of Machine Learning Research* 15, pp. 2689–2722.

Hinton, Geoffrey E. et al. (1995). "The 'Wake-Sleep' Algorithm for Unsupervised Neural Networks". In: *Science* 268, pp. 1158 –1161.

Hoffman, Matthew D. (2012). "Poisson-uniform nonnegative matrix factorization". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5361–5364.

Hoyer, Patrik O. (2004). "Non-negative Matrix Factorization with Sparseness Constraints". In: *Journal of Machine Learning Research* 5, pp. 1457–69.

Hubel, David H. and Torsten N. Wiesel (1968). "Receptive fields and functional architecture of monkey striate cortex". In: *The Journal of Physiology* 195.1, pp. 215–243.

Hyvärinen, Aapo and Hiroshi Morioka (2016). "Unsupervised Feature Extraction by Time-Contrastive Learning and Nonlinear ICA". In: *arXiv preprint arXiv:1605.06336*.

— (2017). "Nonlinear ICA of Temporally Dependent Stationary Sources". In: Proceedings of Machine Learning Research, pp. 460–469.

Hyvärinen, Aapo and Erkki Oja (1997). "A Fast Fixed-Point Algorithm for Independent Component Analysis". In: *Neural Computation* 9.7, pp. 1483–1492.

— (2000). "Independent component analysis: algorithms and applications". In: *Neural Networks* 13, pp. 411–430.

Hyvärinen, Aapo and Petteri Pajunen (1999). "Nonlinear independent component analysis: Existence and uniqueness results". In: *Neural Networks* 12.3, pp. 429–439.

Hyvärinen, Aapo, Hiroaki Sasaki, and Richard Turner (2019). "Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning". In: *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 859–868.

Imamura, Ryuji, Tatsuki Itasaka, and Masahiro Okuda (2019). "Zero-Shot Hyperspectral Image Denoising with Separable Image Prior". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0.

Izenman, Alan Julian (2008). "Modern Multivariate Statistical Techniques". In: *Regression, Classification and Manifold Learning* 10, pp. 978–0.

Jensen, Johan Ludwig William Valdemar et al. (1906). "Sur les fonctions convexes et les inégalités entre les valeurs moyennes". In: *Acta Mathematica* 30, pp. 175–193.

Jernite, Yacine, Yonatan Halpern, and David Sontag (2013). "Discovering Hidden Variables in Noisy-Or Networks using Quartet Tests". In: *Advances in Neural Information Processing Systems* 26, pp. 2355–2363.

Jiang, Fang et al. (2020). "Review of the Clinical Characteristics of Coronavirus Disease 2019 (COVID-19)". In: *Journal of General Internal Medicine*, pp. 1–5.

Jin, Qiyu et al. (2018). "Poisson image denoising by piecewise principal component analysis and its application in single-particle X-ray diffraction imaging". In: *IET Image Processing* 12.12, pp. 2264–2274.

Jones, Judson P. and Larry A. Palmer (1987). "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex". In: *Journal of Neurophysiology* 58.6, pp. 1233–1258.

Jordan, Michael I. et al. (1999). "An Introduction to Variational Methods for Graphical Models". In: *Machine Learning* 37, pp. 183–233.

Karklin, Yan and Michael S. Lewicki (2003). "Learning higher-order structures in natural images". In: *Network: Computation in Neural Systems*.

— (2009). "Emergence of complex cell properties by learning to generalize in natural scenes". In: *Nature* 457.7225, pp. 83–86.

Kass, Robert E. and Duane Steffey (1989). "Approximate Bayesian Inference in Conditionally Independent Hierarchical Models (Parametric Empirical Bayes Models)". In: *Journal of the American Statistical Association* 84.407, pp. 717–726.

Khan, Mohammad Emtiyaz et al. (2010). "Variational Bounds for Mixed-Data Factor Analysis". In: *Advances in Neural Information Processing Systems*, pp. 1108–1116.

Khemakhem, Ilyes et al. (2020). "Variational Autoencoders and Nonlinear ICA: A Unifying Framework". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 2207–2217.

Khoshaman, Amir H. and Mohammad H. Amin (2018). "GumBolt: Extending Gumbel trick to Boltzmann priors". In: *arXiv preprint arXiv:1805.07349*.

Kingma, Diederik P. and Max Welling (2013). "Auto-Encoding Variational Bayes". In: *arXiv preprint arXiv:1312.6114*.

Koller, Daphne and Nir Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Kumwilaisak, Wuttipong et al. (2020). "Image Denoising With Deep Convolutional Neural and Multi-Directional Long Short-Term Memory Networks Under Poisson Noise Environments". In: *IEEE Access* 8, pp. 86998–87010.

Lally, Adam et al. (2017). "WatsonPaths: Scenario-Based Question Answering and Inference over Unstructured Information". In: *AI Magazine* 38.2, pp. 59–76.

Larochelle, Hugo and Yoshua Bengio (2008). "Classification using discriminative restricted Boltzmann machines". In: *Proceedings of the 25th International Conference on Machine Learning*. ACM, pp. 536–543.

Lee, Daniel D. and H. Sebastian Seung (1999). "Learning the parts of objects by non-negative matrix factorization". In: *Nature* 401.6755, pp. 788–91.

Lee, Dong-Hyun (2013). "Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks". In: *Workshop on Challenges in Representation Learning, ICML*. Vol. 3, p. 2.

Lee, Honglak et al. (2009). "Exponential Family Sparse Coding with Application to Self-taught Learning". In: *IJCAI*. Vol. 9, pp. 1113–1119.

Lehnhardt, Ernst (2009). *Praxis der Audiometrie*. Georg Thieme Verlag.

Lindsten, Fredrik (2013). "An efficient stochastic approximation EM algorithm using conditional particle filters". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, pp. 6274–6278.

Ling, Yuan et al. (2017). "Diagnostic Inferencing via Improving Clinical Concept Extraction with Deep Reinforcement Learning: A Preliminary Study". In: *Machine Learning for Healthcare Conference*, pp. 271–285.

Lipton, Zachary C. et al. (2015). "Learning to Diagnose with LSTM Recurrent Neural Networks". In: *arXiv preprint arXiv:1511.03677*.

Lloyd, Stuart (1982). "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2, pp. 129–137.

Lorberbom, Guy et al. (2018). "Direct Optimization through arg max for Discrete Variational Auto-Encoder". In: *arXiv preprint arXiv:1806.02867*.

Lu, Meng, Jianhua Z. Huang, and Xiaoning Qian (2016). "Sparse exponential family Principal Component Analysis". In: *Pattern Recognition* 60, pp. 681–691.

Lücke, Jörg (2019). "Truncated Variational Expectation Maximization". In: *arXiv* 1610.

Lücke, Jörg, Zhenwen Dai, and Georgios Exarchakis (2017). "Truncated Variational Sampling for "Black Box" Optimization of Generative Models". In: arXiv:1712.08104.

Lücke, Jörg and Julian Eggert (2010). "Expectation Truncation and the Benefits of Preselection In Training Generative Models". In: *Journal of Machine Learning Research* 11, pp. 2855–900.

Lücke, Jörg and Dennis Forster (2019). "k-means as a variational EM approximation of Gaussian mixture models". In: *Pattern Recognition Letters* 125, pp. 349–356.

Lücke, Jörg and Maneesh Sahani (2008). "Maximal Causes for Non-linear Component Extraction". In: *Journal of Machine Learning Research* 9, pp. 1227–67.

Lücke, Jörg et al. (2009). "Occlusive Components Analysis". In: *Advances in Neural Information Processing Systems* 22, pp. 1069–77.

MacKay, David J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Mairal, Julien et al. (2010). "Online Learning for Matrix Factorization and Sparse Coding". In: *Journal of Machine Learning Research* 11.

Makitalo, Markku and Alessandro Foi (2010). "Optimal Inversion of the Anscombe Transformation in Low-Count Poisson Image Denoising". In: *IEEE Transactions on Image Processing* 20.1, pp. 99–109.

Mátyás, László (1999). *Generalized Method of Moments Estimation*. Vol. 5. Cambridge University Press.

Mcauliffe, Jon D. and David M. Blei (2008). "Supervised Topic Models". In: *Advances in Neural Information Processing Systems*, pp. 121–128.

Minka, Thomas P. (2002). "Estimating a Gamma Distribution". In: *Microsoft Research, Cambridge, UK, Technical Report*.

Miotto, Riccardo et al. (2016). "Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records". In: *Scientific Reports* 6.1, pp. 1–10.

Młynarski, Wiktor and Josh H. McDermott (2018). "Learning Midlevel Auditory Codes from Natural Sound Statistics". In: *Neural Computation* 30.3, pp. 631–669.

Mohamed, Shakir, Zoubin Ghahramani, and Katherine A. Heller (2008). "Bayesian Exponential Family PCA". In: *Advances in Neural Information Processing Systems* 21, pp. 1089–1096.

Mohamed, Shakir, Katherine A. Heller, and Zoubin Ghahramani (2010). "Sparse exponential family latent variable models". In: *NIPS Workshop*.

— (2011). "Bayesian and L1 Approaches to Sparse Unsupervised Learning". In: *arXiv preprint arXiv:1106.1157*.

Mou, Tian, Jian Huang, and Finbarr O'Sullivan (2018). "The Gamma Characteristic of Reconstructed PET Images: Implications for ROI Analysis". In: *IEEE Transactions on Medical Imaging* 37.5, pp. 1092–1102.

Mousavi, Hamid, Jakob Drefs, and Jörg Lücke (2020). "A Double-Dictionary Approach Learns Component Means and Variances for V1 Encoding". In: *International Conference on Machine Learning, Optimization, and Data Science*. Springer, pp. 240–244.

Mousavi, Hamid et al. (2020). "Maximal Causes for Exponential Family Observables". In: *arXiv preprint arXiv:2003.02214*.

Mousavi, Hamid et al. (2021). "Inference and Learning in a Latent Variable Model for Beta Distributed Interval Data". In: *Entropy* 23.5, p. 552.

Neal, Radford M. and Geoffrey E. Hinton (1998). "A View of the EM Algorithm that Justifies Incremental, Sparse, and other Variants". In: *Learning in Graphical Models*. Kluwer.

Niknejad, Milad and Mário A. T. Figueiredo (2018). "Poisson Image Denoising Using Best Linear Prediction: A Post-processing Framework". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, pp. 2230–2234.

Olshausen, Bruno A., Charles F. Cadieu, and David K. Warland (2009). "Learning real and complex overcomplete representations from the statistics of natural images". In: *Wavelets XIII*. Vol. 7446. International Society for Optics and Photonics, 74460S.

Olshausen, Bruno A. and David J. Field (1996a). "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". In: *Nature* 381, pp. 607–9.

— (1996b). "Natural image statistics and efficient coding". In: *Network: Computation in Neural Systems* 7.2, pp. 333–339.

— (1997). "Sparse coding with an overcomplete basis set: A strategy employed by V1?" In: *Vision Research* 37.23, pp. 3311–3325.

— (2004). "Sparse coding of sensory inputs". In: *Current Opinion in Neurobiology* 14.4, pp. 481–487.

Oord, Aaron van den, Oriol Vinyals, and Koray Kavukcuoglu (2017). "Neural Discrete Representation Learning". In: *arXiv preprint arXiv:1711.00937*.

Opper, Manfred and Cédric Archambeau (2009). "The Variational Gaussian Approximation Revisited". In: *Neural Computation* 21.3, pp. 786–792.

Opper, Manfred and Ole Winther (2005). "Expectation Consistent Approximate Inference". In: *Journal of Machine Learning Research* 6.12, pp. 2177–2204.

Patel, Ankit B., Tan Nguyen, and Richard G. Baraniuk (2016). "A probabilistic Theory of Deep Learning". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2558–2566.

Pearl, Judea (2014). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier.

Pearson, Karl (1904). *On the theory of contingency and its relation to association and normal correlation*. Dulau and Company.

Piriyatharawet, Teerawat, Wuttipong Kumwilaisak, and Pongsak Lasang (2018). "Image Denoising with Deep Convolutional and Multi-directional LSTM Networks under Poisson Noise Environments". In: *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE, pp. 90–95.

Puertas, Jose, Jörg Bornschein, and Jörg Lücke (2010). "The Maximal Causes of Natural Scenes are Edge Filters". In: *Advances in Neural Information Processing Systems*. Vol. 23, pp. 1939–1947.

Rajkomar, Alvin et al. (2018). "Scalable and accurate deep learning with electronic health records". In: *NPJ Digital Medicine* 1.1, p. 18.

Ramaswami, Prem (2015). "A remedy for your health-related questions: health info in the Knowledge Graph". In: *Google Official Blog*.

Ravuri, Murali et al. (2018). "Learning from the experts: From expert systems to machine-learned diagnosis models". In: *Machine Learning for Healthcare Conference*. PMLR, pp. 227–243.

Remez, Tal et al. (2017). "Deep Convolutional Denoising of Low-Light Images". In: *arXiv preprint arXiv:1701.01687*.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra (2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *International Conference on Machine Learning*. PMLR, pp. 1278–1286.

Rockafellar, R. Tyrrell and Roger J-B Wets (2009). *Variational Analysis*. Vol. 317. Springer Science & Business Media.

Rolfe, Jason Tyler (2016). "Discrete Variational Autoencoders". In: *arXiv preprint arXiv:1609.02200*.

Rond, Arie, Raja Giryes, and Michael Elad (2016). "Poisson Inverse Problems by the Plug-and-Play scheme". In: *arXiv:1511.02500, later in Journal of Visual Communication and Image Representation* 41, pp. 96–108.

Rothan, Hussin A. and Siddappa N. Byrareddy (2020). "The epidemiology and pathogenesis of coronavirus disease (COVID-19) outbreak". In: *Journal of Autoimmunity*, p. 102433.

Rotmensch, Maya et al. (2017). "Learning a Health Knowledge Graph from Electronic Medical Records". In: *Scientific Reports* 7.1, p. 5994.

Roweis, Sam (1998). "EM Algorithms for PCA and SPCA". In: *Advances in Neural Information Processing Systems*, pp. 626–632.

Roweis, Sam T. (2003). "Factorial Models and Refiltering for Speech Separation and Denoising". In: *Eighth European Conference on Speech Communication and Technology*. Vol. 7, pp. 1009 –1012.

Roy, Aurko et al. (2018). "Theory and Experiments on Vector Quantized Autoencoders". In: *arXiv preprint arXiv:1805.11063*.

Sadeghi, Hossein et al. (2019). "Pixelvae++: Improved PixelVAE with Discrete Prior". In: *arXiv preprint arXiv:1908.09948*.

Sahani, Maneesh (1999). "Latent Variable Models for Neural Data Analysis". PhD thesis. Caltech.

Salmon, Joseph et al. (2014). "Poisson Noise Reduction with Non-local PCA". In: *Journal of Mathematical Imaging and Vision* 48.2, pp. 279–294.

Saul, Lawrence K., Tommi Jaakkola, and Michael I. Jordan (1996). "Mean Field Theory for Sigmoid Belief Networks". In: *Journal of Artificial Intelligence Research* 4.1, pp. 61–76.

Schauerte, Boris and Rainer Stiefelhagen (2013). ""Wow!" Bayesian surprise for salient acoustic event detection". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, pp. 6402–6406.

Seeger, Matthias W. (2008). "Bayesian Inference and Optimal Design for the Sparse Linear Model". In: *Journal of Machine Learning Research* 9, pp. 759–813.

Sheikh, Abdul-Saboor, Jacquelyn A. Shelton, and Jörg Lücke (2014). "A Truncated EM Approach for Spike-and-Slab Sparse Coding". In: *Journal of Machine Learning Research* 15, pp. 2653–2687.

Sheikh, Abdul-Saboor et al. (2019). "STRFs in primary auditory cortex emerge from masking-based statistics of natural sounds". In: *PLOS Computational Biology* 15.1, e1006595.

Shelton, Jacquelyn A. et al. (2017). "GP-select: Accelerating EM using adaptive subspace preselection". In: *Neural Computation* 29.8. 1st version, arXiv:1412.3411, online since 2014, pp. 2177–2202.

Shen, Ying et al. (2018). "CBN: Constructing a clinical Bayesian network based on data from the electronic medical record". In: *Journal of Biomedical Informatics* 88, pp. 1–10.

Shickel, Benjamin et al. (2017). "Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis". In: *IEEE Journal of Biomedical and Health Informatics* 22.5, pp. 1589–1604.

Shivkumar, Sabyasachi et al. (2018). "A probabilistic population code based on neural samples". In: *Advances in Neural Information Processing Systems*. Vol. 31.

Shocher, Assaf, Nadav Cohen, and Michal Irani (2018). ""Zero-shot" Super-Resolution using Deep Internal Learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3118–3126.

Shwe, Michael A. et al. (1991). "Probabilistic Diagnosis Using a Reformulation of the INTERNIST-1/QMR Knowledge Base". In: *Methods of Information in Medicine* 30.04, pp. 241–255.

Šingliar, Tomáš and Miloš Hauskrecht (2006). "Noisy-OR Component Analysis and its Application to Link Analysis". In: *Journal of Machine Learning Research* 7, pp. 2189–2213.

Skrondal, Anders and Sophia Rabe-Hesketh (2007). "Latent Variable Modelling: A Survey". In: *Scandinavian Journal of Statistics* 34.4, pp. 712–745.

Spratling, Michael W. (2006). "Learning Image Components for Object Recognition". In: *Journal of Machine Learning Research* 7.5, pp. 793–815.

Suresh, Harini et al. (2017). "Clinical Intervention Prediction and Understanding using Deep Networks". In: *arXiv preprint arXiv:1705.08498*.

Tai, Ying et al. (2017). "MemNet: A Persistent Memory Network for Image Restoration". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4539–4547.

Tarantola, Albert (2005). *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM.

Teymurazyan, A. et al. (2013). "Properties of noise in positron emission tomography images reconstructed with filtered-backprojection and row-action maximum likelihood algorithm". In: *Journal of Digital Imaging* 26.3, pp. 447–456.

Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, pp. 267–288.

Tipping, Michael E. and Christopher M. Bishop (1999). "Probabilistic Principal Component Analysis". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3, pp. 611–622.

Titsias, Michalis K. and Miguel Lázaro-Gredilla (2011). "Spike and Slab Variational Inference for Multi-Task and Multiple Kernel Learning". In: *Advances in Neural Information Processing Systems*. Vol. 24.

Triguero, Isaac, Salvador García, and Francisco Herrera (2015). "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study". In: *Knowledge and Information Systems* 42.2, pp. 245–284.

Ueda, Naonori and Ryohei Nakano (1998). "Deterministic annealing EM algorithm". In: *Neural Networks* 11.2, pp. 271–82.

Vahdat, Arash, Evgeny Andriyash, and William G. Macready (2019). "Learning Undirected Posteriors by Backpropagation through MCMC Updates". In: *arXiv preprint arXiv:1901.03440*.

Valera, Isabel and Zoubin Ghahramani (2017). "Automatic Discovery of the Statistical Types of Variables in a Dataset". In: *International Conference on Machine Learning*, pp. 3521–3529.

Valera, Isabel et al. (2020). "General Latent Feature Models for Heterogeneous Datasets". In: *Journal of Machine Learning Research* 21.100, pp. 1–49.

van der Linden, Wim J. and Ronald K. Hambleton (2013). *Handbook of Modern Item Response Theory*. Springer Science & Business Media.

van Esch, Thamar E. M. et al. (2013). "Evaluation of the preliminary auditory profile test battery in an international multi-centre study". In: *International Journal of Audiology* 52.5, pp. 305–321.

van Hateren, J. Hans and Arjen van der Schaaf (1998). "Independent component filters of natural images compared with simple cells in primary visual cortex". In: vol. 265. 1394. The Royal Society, pp. 359–366.

Vergari, Antonio et al. (2019). "Automatic Bayesian Density Analysis". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 5207–5215.

Wainwright, Martin J. and Michael I. Jordan (2008). *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers, Inc.

Wainwright, Martin J. and Eero P. Simoncelli (1999). "Scale Mixtures of Gaussians and the Statistics of Natural Images". In: *Advances in Neural Information Processing Systems*. Vol. 12, pp. 855–861.

Wang, Fei et al. (2014). "Clinical Risk Prediction with Multilinear Sparse Logistic Regression". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 145–154.

Wei, Greg C. G. and Martin A. Tanner (1990). "A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms". In: *Journal of the American Statistical Association* 85.411, pp. 699–704.

Wonnacott, Thomas H. and Ronald J. Wonnacott (1990). *Introductory Statistics for Business and Economics*. Vol. 4. Wiley New York.

Xie, Yusheng et al. (2016). "Variational hybridization and transformation for large inaccurate noisy-or networks". In: *preprint arXiv:1605.06181*.

Zhang, Kai, Wangmeng Zuo, and Lei Zhang (2018). "FFDNet: Toward a Fast and Flexible Solution for CNN-based Image Denoising". In: *IEEE Transactions on Image Processing* 27.9, pp. 4608–4622.

Zhang, Kai et al. (2017). "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising". In: *IEEE Transactions on Image Processing* 26.7, pp. 3142–3155.

Zhou, Mingyuan et al. (2012). "Beta-Negative Binomial Process and Poisson Factor Analysis". In: *Artificial Intelligence and Statistics*, pp. 1462–1471.

Zoran, Daniel and Yair Weiss (2009). "The 'tree-dependent components' of natural scenes are edge filters". In: *Advances in Neural Information Processing Systems* 22, pp. 2340–8.

— (2011). "From learning models of natural image patches to whole image restoration". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 479–486.