Approximate Variational Inference For Mixture Models

Annäherungsweise Variationsinferenz für Mischungsmodelle
Master thesis by Janosch Moos
Date of submission: January 30, 2021

1. Review: Hany Abdulsamad
2. Review: Svenja Stark
3. Review: Jan Peters
Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Erklärung zur Abschlussarbeit gemäß §22 Abs. 7 und §23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Janosch Moos, die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß §23 Abs. 7 APB überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 30. Januar 2021

J. Moos

Abstract

Learning truths behind real, relevant data is faced with uncertainty. A probabilistic view on unsupervised learning considers this uncertainty in its learning objectives through probability distributions and inference. The expressiveness of probability distributions is further enhanced by mixture models, a linear combination of such distributions. This thesis discusses the concept of variational inference in Bayesian Gaussian mixture models. We present different approaches to modeling the variational posterior of such mixtures with Normalizing Flows and provide proof of concept experimental results. The approaches have been devised as improvements and extensions to one another, ultimately leading to collapsed variational inference, where the assignment variables of the Bayesian Gaussian mixture model have been marginalized. We further derived a decomposable update procedure for mixtures of Normalizing Flows based on work presented in density estimation for Gaussian mixtures.

Zusammenfassung

Um Zusammenhänge in realen, relevanten Daten zu finden müssen Unsicherheiten in den Daten berücksichtigt werden, was in unüberwachten Lernmethoden durch Wahrscheinlichkeitsverteilungen und Inferenz gemacht wird. Diese Arbeit diskutiert das Konzept der Variationsinferenz in Bayes'schen Gaußschen Mischungsmodellen, welche die Aussagekraft der Gaußverteilung weiter verstärken. Es werden verschiedene Ansätze zur Modellierung nötiger Variations-A-posteriori-Verteilungen solcher Mischungsmodelle mit Normalizing Flows vorgestellt und experimentelle Ergebnisse als Konzeptnachweis präsentiert. Die Ansätze sind Weiterentwicklungen voneinander und führen letztendlich zu kollabierter Variationsinferenz, bei der die Zuweisungsvariablen des Bayes'schen Gaußschen Mischungsmodells marginalisiert wurden. Ein weiteres Ergebnis dieser Arbeit sind separierbare Updates für Mischungsmodelle von Normalizing Flows, die auf Arbeiten in der Dichteschätzung für Gaußsche Mischungen basieren.

Contents

Figures

List of Figures

Abbreviations, Symbols and Operators

List of Abbreviations

Notation	Description
ELBO	*Evidence Lower Bound*
EM	*Expectation-Maximization*
i.i.d.	*independently and identically distributed*
I-Projection	*Information-Projection*
KL	*Kullback-Leibler-Divergence*
LVM	*Latent Variable Model*
MAP	*maximum a-posteriori*
MCMC	*Markov Chain Monte-Carlo*
ML	*maximum likelihood*
M-Projection	*Moment-Projection*

SGD	*Stochastic Gradient Descent*
VAE	*Variational Auto-Encoder*

1. Introduction

With the rapidly increasing computational capabilities of our modern age, the amount of data gathered on a daily basis is staggering [1, 2]. Unsupervised learning is the study of finding and extracting valuable information and patterns in such data [2, 3]. The data, however, may well be just a finite set of samples gathered from some unknown random process [3]. Furthermore, we might only have some noisy observation of the true data [3]. Machine learning adopts a probabilistic view to quantify such uncertainty or possible ambiguity in the data, which means involving probabilities [2–4]. A probability is a measure of the likelihood of a certain outcome when sampling a random process, where the outcome is defined as a specific value a random variable takes on. The function that assigns a probability to such an observed value is a probability distribution [2]. A central problem of unsupervised learning is finding the best probabilistic model to describe a finite set of data [2]. Such models are also referred to as *generative* models for their ability to generate new data [1, 5]. However, suppose we assume a probabilistic model with a specified functional form. How do we choose the parameters of this model to best describe the observed data within the constraints of its functional form? This thesis evolves around reasoning or *inferring* the parameters of such models.

Let $\mathbf{X}$ be a set of data we have observed. We will define a probabilistic model as a probability distribution $p(\mathbf{X} \mid \boldsymbol{\theta})$ whose parameters are denoted by $\boldsymbol{\theta}$. This probability distribution describes the *likelihood* of the observed data $\mathbf{X}$ under the parameters $\boldsymbol{\theta}$ [5]. Maximizing the parameters as a point estimate such that

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{X} \mid \boldsymbol{\theta}),$$

is known as *maximum likelihood* (ML) estimation [2,3,5]. While appealing for its simplicity the ML estimate carries the risk of *overfitting* the given data. Given e.g., an overly complex model with an insufficient amount of data, when learning the underlying truth of the data we may find that we have learned to represent the noise overlaying the data instead [2, 5].

By incorporating additional prior information about the state of the parameters we alleviate this problem [1–3]. This prior information may e.g., be some assumption or belief about the parameters described by a probability distribution $p(\boldsymbol{\theta})$ known as the *prior* [2, 3, 5]. This concept is referred to as the *Bayesian* approach [3]. Given a likelihood $p(\mathbf{X} \mid \boldsymbol{\theta})$ and a prior $p(\boldsymbol{\theta})$ we infer a posterior distribution

$$p(\boldsymbol{\theta} \mid \mathbf{X}) = \frac{p(\mathbf{X} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\mathbf{X})} = \frac{p(\mathbf{X} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{\int p(\mathbf{X} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}},$$

where the relation of the posterior to likelihood and prior is known as *Bayes Rule* or *Bayes Theorem* [2–5]. The denominator represents a normalization to ensure that the posterior is a proper distribution with probabilities in $[0, 1]$ [2, 3]. Optimizing the posterior w.r.t. a point estimate of the parameters

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathbf{X}) = \arg\max_{\boldsymbol{\theta}} p(\mathbf{X} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta}),$$

is known as the *maximum a-posteriori* (MAP) estimate [1–5]. For uniform priors, where all values of the parameters are assigned the same probability, the MAP estimate converges to the ML estimate [1, 2]. The same behaviour occurs in the limit of infinite data as the likelihood overwhelms the prior [2]. Even though the MAP estimate incorporates prior information, by optimizing a point estimate we find the mean of a maximizing mode of the posterior distribution.

The *full* Bayesian perspective further considers possible uncertainty or variability in estimating the parameters [1, 2, 5]. Rather than just optimizing the parameter as a point estimate, we marginalize over all possible values of the parameters $\boldsymbol{\theta}$ such that

$$p(\mathbf{X}) = \int p(\mathbf{X} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta},$$

which is known as the *marginal likelihood* of a probabilistic model for a set of given data [2, 3]. When we want to infer parameters such that a probabilistic model best describes the data while facing uncertainty, the marginal likelihood is the key quantity we need to consider [3, 5]. For most probabilistic models, however, this quantity is intractable [5]. Instead, we simplify the problem by approximating the integral [2, 3, 5]. A traditional approach applicable to a wide range of different probabilistic models are sampling methods such as *Markov Chain Monte-Carlo* (MCMC) [6–8] [2, 3, 5]. Sampling techniques, however, are computationally intensive in large scale problems [2, 3]. For large scale problems focus has shifted to more recent deterministic approximation methods one of which is *variational inference* [9–13] [2, 3, 5]. Since optimizing the intractable

marginal likelihood is impossible, variational inference formulates a lower bound on the integral by introducing an approximate posterior to be optimized instead [2, 3, 5, 9–13]. Even then, optimization might still not be possible directly without further simplifications through, e.g., independencies between random variables of the posterior distribution such as the *Mean-Field* assumption [9, 14–16] [2, 3, 5]. However, in recent years a new highly flexible and complex but easy to use family of parametric generative models has emerged, known as *Normalizing Flows* [17–28]. These generative models allow tractable posterior approximations in the lower bound without introducing independencies between the posterior distribution's random variables. This thesis discusses the application of Normalizing Flows for variational inference where the probabilistic model $p(\mathbf{X} \mid \boldsymbol{\theta})$ is a linear combination of a set of probability distributions called mixture model.

The thesis is organized as follows. In Chapter 2 we will discuss some fundamental concepts required for the rest of this thesis. The chapter introduces probability distributions and their role in formulating the mixture model and its priors that we consider for this thesis. Further, we will shortly discuss the *Kullback-Leibler-Divergence* (KL) [29] and its importance when optimizing probability distributions. Chapter 3 will give an in-depth introduction to variational methods and discuss related topics such as MCMC and *Latent Variable Model* (LVM) [30]. We will describe the well known *Expectation-Maximization* (EM) algorithm [31, 32] from a Bayesian perspective and its relation to variational inference. We will derive Mean-Field inference, the importance of the independence assumptions required, and how structures in probabilistic models have been exploited to relieve the independence assumptions [15, 16, 33–35]. The concept of Normalizing Flows as generative models, a short overview of a variety of involved transformations, its application in variational inference will be discussed in Chapter 4. In Chapter 5, we will present the contribution of this thesis to the scientific community and show proof of concept results on various small datasets. In addition to results on variational inference with approximate posterior distributions for mixture models, we will introduce decomposable update procedures for mixtures of Normalizing Flows in density estimation with complementary results and possible extensions to posterior estimation based on work in [36, 37]. The results are further discussed in Chapter 6 including related works such as the *Variational Auto-Encoder* (VAE) [38–40] and *variational boosting* [41–43]. Finally, in Chapter 7 we will describe possible future extensions to this thesis.

2. Preliminaries

Probability distributions and divergences are at the core of the probabilistic view to unsupervised learning and the algorithms we discuss in this thesis. This chapter introduces certain distributions from the exponential family and their function in mixture models with prior information. We will further discuss the concept of the KL [29] required for the rest of this thesis as a measure of similarity between probability distributions.

2.1. Distributions And Mixtures

Probability distributions are functions that provide a probability measure of the likelihood of a certain outcome when sampling a random process [2]. An outcome is defined as a certain value a random variable takes on. The probability measure, thus, quantifies uncertainty about random variables in random processes or observed data. While there is a myriad of different distributions designed for discrete or continuous, skalars, vectors or matrices, this section focuses on a specific group of distributions relevant for this thesis.

2.1.1. Gaussian Distribution

The *Gaussian* distribution, denoted by $\mathcal{N}\left(\mu, \lambda^{-1}\right)$, is the most important distribution in probability theory due to its versatility [2]. Here μ is the mean of the distribution, and $\lambda = 1/\sigma^2$ denotes its precision and is the inverse of the variance σ^2. High precision relates to a very narrow distribution around the mean [2]. Literature often refers to this distribution as *Normal* distribution [2, 3]. In the univariate case, the probability density function is defined by

$$p\left(x\right) = \frac{\sqrt{\lambda}}{\sqrt{2\pi}} \exp\left(-\frac{\lambda}{2}\left(x-\mu\right)^2\right),$$

where $x \in \mathcal{R} \sim \mathcal{N}\left(\mu, \lambda^{-1}\right)$ is a continuous random variable. The multivariate case is defined similarly for a random variable $\mathbf{x} \in \mathcal{R}^D$ with $\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}\right)$ and a probability density function

$$p\left(\mathbf{x}\right) = \frac{|\boldsymbol{\Lambda}|^{\frac{1}{2}}}{(2\pi)^{\frac{D}{2}}} \exp\left(-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}\right)^T \boldsymbol{\Lambda}\left(\mathbf{x}-\boldsymbol{\mu}\right)\right),$$

where $\boldsymbol{\Lambda} \in \mathcal{R}^{D\times D}$ is the positive semi-definite precision matrix [2, 3]. Any given data of a continuous random variable can be appropriately represented with a sufficiently large linear combination of Gaussian distributions known as a mixture distribution (see Section 2.1.3) [3]. This thesis is focused on the multivariate Gaussian distribution, and we will, for the rest of this thesis, always refer to the multivariate Gaussian. Its functional form places the Gaussian distribution into the exponential family of distributions, which significantly simplifies some of the problems we discuss in this thesis, even allowing for analytical solutions [2, 3]. These simplifications mainly evolve around the concept of conjugate priors, which for the exponential family are themselves from the exponential family. Conjugate priors are distributions for which the posterior takes on the same functional form as the prior. For a single Gaussian, there are two such priors, one for each parameter of the distribution. While the conjugate prior for the mean $\boldsymbol{\mu}$ is again a Gaussian, which is the case for both the univariate and multivariate definition, the prior for the precision matrix $\boldsymbol{\Lambda}$ is a *Wishart* distribution [3].

2.1.2. Wishart Distribution

The Wishart distribution $\mathcal{W}\left(\nu, W\right)$ is the multivariate extension of the *Gamma* distribution whose probability density function is defined as

$$p\left(\boldsymbol{\Lambda}\right) = \frac{|\ \boldsymbol{\Lambda}\ |^{\frac{(\nu-D-1)}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{W}^{-1}\boldsymbol{\Lambda}\right)\right)}{2^{\frac{\nu D}{2}}\ |\ \boldsymbol{\Lambda}\ |^{\frac{\nu}{2}}\ \Gamma\left(\frac{\nu}{2}\right)},$$

where $\nu > D-1$ denotes the degrees of freedom and $\mathbf{W} \in \mathcal{R}^{D\times D}$ is a positive definite scale matrix [44–46]. Further, $\mathrm{tr}\left(\cdot\right)$ is the trace and $\Gamma\left(\cdot\right)$ is the gamma function. The parameters ν and $\mathbf{W}$ relate to the mean of the Wishart distribution as $\mathbb{E}\left[\boldsymbol{\Lambda}\right] = \nu\mathbf{W}$. Literature often combines the priors of the Gaussian distribution as a *Gaussian-Wishart* or *Normal-Wishart* denoted by $\mathcal{NW}\left(\boldsymbol{\mu}_0, \beta, \nu, \mathbf{W}\right)$. This combined notation describes a

factorized distribution of the form

$$p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = p(\boldsymbol{\mu} \mid \boldsymbol{\Lambda})\, p(\boldsymbol{\Lambda}) \tag{2.1}$$

$$= \mathcal{N}\left(\boldsymbol{\mu} \,\middle|\, \boldsymbol{\mu}_0, (\beta\boldsymbol{\Lambda})^{-1}\right) \mathcal{W}(\boldsymbol{\Lambda} \mid \nu, \mathbf{W}), \tag{2.2}$$

where $\boldsymbol{\mu}_0$ is the mean of the Gaussian prior on the mean and β is a scaling for the precision matrix in the Gaussian prior [45]. The sampling process, therefore, is sequential and described as follows

$$\boldsymbol{\Lambda} \sim \mathcal{W}(\nu, \mathbf{W}),$$
$$\boldsymbol{\mu} \mid \boldsymbol{\Lambda} \sim \mathcal{N}\left(\boldsymbol{\mu}_0, (\beta\boldsymbol{\Lambda})^{-1}\right),$$

with $\boldsymbol{\Lambda}$ being sampled first and $\boldsymbol{\mu}$ being drawn conditioned on $\boldsymbol{\Lambda}$.

Given a Gaussian distribution with a Gaussian-Wishart prior, one can obtain a predictive density $p(\mathbf{x} \mid \boldsymbol{\mu}_0, \beta, \nu, \mathbf{W})$ by marginalizing over the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ of the Gaussian distribution

$$p(\mathbf{x} \mid \boldsymbol{\mu}_0, \beta, \nu, \mathbf{W}) = \int\int p\left(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}\right) p\left(\boldsymbol{\mu} \,\middle|\, \boldsymbol{\mu}_0, (\beta\boldsymbol{\Lambda})^{-1}\right) p(\nu, \mathbf{W})\, \mathrm{d}\boldsymbol{\mu}\mathrm{d}\boldsymbol{\Lambda}$$

which corresponds to a *Student-t* distribution, whose probability density function can be described in terms of the parameters of the priors as

$$p(\mathbf{x}) = \frac{\Gamma\left(\frac{\nu_t + D}{2}\right) \mid \boldsymbol{\Lambda}_t \mid^{\frac{1}{2}}}{(\nu_t \pi)^{\frac{D}{2}}\, \Gamma\left(\frac{\nu_t}{2}\right)} \left(1 + \frac{1}{\nu_t}(\mathbf{x} - \boldsymbol{\mu}_t)^T \boldsymbol{\Lambda}_t (\mathbf{x} - \boldsymbol{\mu}_t)\right)^{-\frac{\nu_t + D}{2}},$$

where $\nu_t = \nu - D + 1$, $\boldsymbol{\mu}_t = \boldsymbol{\mu}_0$ and $\boldsymbol{\Lambda}_t = ((\beta + 1) / (\beta\nu_t))\, \mathbf{W}^{-1}$. The same relation holds for the parameters of the posterior as it has the same functional form. Therefore, given a set of *independently and identically distributed* (i.i.d.) observations $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, which we assume are distributed according to a Gaussian distribution with Gaussian-Wishart prior, we can compute a Bayesian update for the parameters of the posterior in closed form as

$$\nu_{new} = \nu + N, \tag{2.3}$$

$$\beta_{new} = \beta + N, \tag{2.4}$$

$$\boldsymbol{\mu}_{new} = \frac{\beta\boldsymbol{\mu}_0 + N\bar{\mathbf{x}}}{\beta + N}, \tag{2.5}$$

$$\mathbf{W}_{new} = \left(\mathbf{W}^{-1} + \mathbf{S} + \frac{N\beta}{\beta + N}(\bar{\mathbf{x}} - \boldsymbol{\mu}_0)(\bar{\mathbf{x}} - \boldsymbol{\mu}_0)^T\right)^{-1}, \tag{2.6}$$

where $\bar{\mathbf{x}}$ is the mean and

$$\mathbf{S} = \sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})^T ,$$

is the scatter matrix of the observations. The predictive density $p(\mathbf{x} \mid X)$ for a new data point $\mathbf{x}$ is then defined as a Student-t distribution in relation to the updated parameters [44–46].

2.1.3. Gaussian Mixture Models

In the context of this thesis, we will consider data that is highly multi-modal. Matching such data with a single Gaussian distribution is insufficient as the Gaussian is a uni-modal distribution. Instead, we need to consider mixture distributions, which are a linear combination of some basic distribution such as the Gaussian for which the mixture is defined as

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N} (\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) ,$$

where $\pi_k \in \mathcal{R}$ is the *mixing coefficient* of the k^{th} component [3]. The mixing coefficients can be seen as a prior probability on the components with the basic properties of a distribution, namely

$$\sum_{k=1}^{K} \pi_k = 1, \qquad 0 \leq \pi_k \leq 1 \, \forall \, k = 1, ..., K.$$

On this basis, the mixture is redefined as

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(z = k) \, p (\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) ,$$

where $p(z = k) = \pi_k$ with z being a discrete random variable. The distribution over z represents a *Categorical* distribution whose parameters are $\boldsymbol{\pi} = \{\pi_1, ..., \pi_K\}$ such that $z \sim \text{Cat}(\boldsymbol{\pi})$.

The full set of parameters of the Gaussian mixture distribution are $\theta = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\}$, where $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K\}$ and $\boldsymbol{\Lambda} = \{\boldsymbol{\Lambda}_1, ..., \boldsymbol{\Lambda}_K\}$ are sets over the parameters of each component. As each component of the mixture distribution is a basic Gaussian distribution, the conjugate prior over each of the parameters in $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ are still of the Gaussian-Wishart

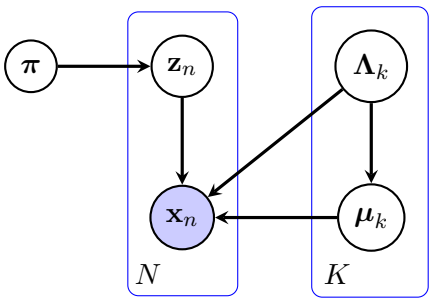


Figure 2.1.: Shown is the graphical model of a Bayesian Gaussian mixture model with K components. It is assumed that $\mathbf{x}_n$ is the n^{th} observation of a set of N observations. For each observation n there is an assignment to a mixture component denoted by $\mathbf{z}_n \in \mathcal{Z}^K$ as a one-hot vector. The mixture is parameterized by the mixing coefficients $\boldsymbol{\pi} = \{\pi_1, ..., \pi_K\}$, the component means $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K\}$ and the components' precision matrices $\boldsymbol{\Lambda} = \{\boldsymbol{\Lambda}_1, ..., \boldsymbol{\Lambda}_K\}$.

type. However, for a full Bayesian view on this model an additional prior over $\boldsymbol{\pi}$ is to be considered for which the conjugate takes on the form of a *Dirichlet* distribution [3]. The Dirichlet distribution, denoted by $\mathrm{Dir}(\boldsymbol{\alpha})$ with $\boldsymbol{\alpha} \in \mathcal{R}^K$, is a multivariate generalization of the *Beta* distribution defined as

$$p(\boldsymbol{\pi}) = \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_i)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1},$$

where $\Gamma(a) = (a-1)!$ is the gamma function.

In summary, we will define a full *Bayesian Gaussian mixture model* as a combination of the basic mixture of K Gaussian distributions and all of its priors, which are assumed to be conjugate. The graphical representation over all continuous random variables $\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}$ and the discrete random variable $\mathbf{z}$ is shown in Figure 2.1 [2, 3]. We will, for this model, assume that $\mathbf{z} \in \mathcal{Z}^K$ now represents an assignment as a one-hot vector of length K for which we define

$$p(\mathbf{z} \mid \boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{z_k},$$

where z_k is the k^{th} element of $\mathbf{z}$. The discrete variable $\mathbf{z}$ can be seen as an assignment of an

observation of $\mathbf{x}$ to the k^{th} component in the mixture, which is assumed to be responsible for the creation of that observation. We can recover the non-Bayesian mixture of Gaussians by marginalizing out $\mathbf{z}$ when no other priors are present as

$$\begin{aligned} p(\mathbf{x}) &= \sum_{\mathbf{z}} p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) \\ &= \sum_{\mathbf{z}} \prod_{k=1}^{K} \pi_k^{z_k} \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)^{z_k} \\ &= \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \end{aligned}$$

which we will use later in this thesis. To describe the Bayesian Gaussian mixture distribution in its entirety, we define the conjugate priors

$$\begin{aligned} \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k &\sim \mathcal{NW}(\boldsymbol{\mu}_0, \beta, \nu, \mathbf{W}) \qquad \forall\, k = 1, ..., K, \\ \boldsymbol{\pi} &\sim \mathrm{Dir}(\boldsymbol{\alpha}), \\ \mathbf{z} &\sim \mathrm{Cat}(\boldsymbol{\pi}). \end{aligned}$$

It follows the joint probability over all random variables

$$\begin{aligned} p(\mathbf{x}, \mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) &= p(\mathbf{z} \mid \boldsymbol{\pi})\, p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) \prod_{k=1}^{K} p(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)^{z_k}\, p(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \\ &= C_\alpha \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \mathcal{NW}(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k \mid \boldsymbol{\mu}_0, \beta, \nu, \mathbf{W})\, \pi_k^{z_k} \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)^{z_k}, \end{aligned} \tag{2.7}$$

where we marginalize over all parameters $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}$ and the assignments to define the probability for an observation $\mathbf{x}$ as

$$p(\mathbf{x}) = \sum_{\mathbf{z}} \int\int\int p(\mathbf{x}, \mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})\, \mathrm{d}\boldsymbol{\pi} \mathrm{d}\boldsymbol{\mu} \mathrm{d}\boldsymbol{\Lambda}, \tag{2.8}$$

which, however, is intractable [3]. Solving such a marginalization will be the central issue discussed in the following chapter, where most methods presented rely on minimizing the KL.

2.2. Kullback-Leibler Divergence

A fundamental concept at the core of inference when dealing with variational methods is the KL, which represents a measure for the distance between probability distributions.

Let $p(\mathbf{x})$ be a probability distribution over some random variable $\mathbf{x} \in \mathcal{R}^D$. We will, without loss of generality, assume that the random variable is continuous. Further, let $q(\mathbf{x})$ be a second distribution, which, for correlation purposes to later chapters, we will assume is some model on the same space as $p(\mathbf{x})$. Given the densities of these distributions, the family of f-Divergences is defined as

$$\mathrm{D}(p(\mathbf{x}) \,\|\, q(\mathbf{x})) = \int q(\mathbf{x}) f\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) \mathrm{d}\mathbf{x},$$

where $f(\cdot)$ is convex on $\mathbf{x} > 0$ with $f(1) = 0$. The KL is as an f-Divergence whose function $f(\cdot)$ is defined as $f(t) = t \cdot \log t$ leading to

$$\mathrm{KL}(p(\mathbf{x}) \,\|\, q(\mathbf{x})) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x},$$

which is also known as the relative entropy between $q(\mathbf{x})$ and $p(\mathbf{x})$. The name comes from the integral as it describes an expectation of the log-term w.r.t. to $p(\mathbf{x})$ as

$$\begin{aligned}\mathrm{KL}(p(\mathbf{x}) \,\|\, q(\mathbf{x})) &= \mathbb{E}_{p(\mathbf{x})}[\log p(\mathbf{x})] - \mathbb{E}_{p(\mathbf{x})}[\log q(\mathbf{x})] \\ &= H(p(\mathbf{x}), q(\mathbf{x})) - H(p(\mathbf{x}))\end{aligned}$$

where $\mathbb{E}_{p(\mathbf{x})}[\log p(\mathbf{x})] = -H(p(\mathbf{x}))$ is the *Shannon* entropy [47] of $p(\mathbf{x})$ and $\mathbb{E}_{p(\mathbf{x})}[\log q(\mathbf{x})] = -H(p(\mathbf{x}), q(\mathbf{x}))$ is the cross entropy [4,48–51]. The KL is defined for $\mathrm{KL}(p(\mathbf{x}) \,\|\, q(\mathbf{x})) \geq 0$. With $p(\mathbf{x}) = q(\mathbf{x})$ corresponding to

$$\mathrm{KL}(p(\mathbf{x}) \,\|\, p(\mathbf{x})) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{p(\mathbf{x})} \mathrm{d}\mathbf{x} = 0,$$

minimizing the model $q(\mathbf{x})$ over an objective defined as KL leads to equality between $p(\mathbf{x})$ and $q(\mathbf{x})$ [51]. An important property of the KL to consider, however, is *asymmetry*, meaning that $\mathrm{KL}(p(\mathbf{x}) \,\|\, q(\mathbf{x})) \neq \mathrm{KL}(q(\mathbf{x}) \,\|\, p(\mathbf{x}))$ [49,51]. In the optimization we will be discussing in the course of this thesis, asymmetry has an important effect. Optimizing the original form $\mathrm{KL}(p(\mathbf{x}) \,\|\, q(\mathbf{x}))$ or *Moment-Projection* (M-Projection) w.r.t. $q(\mathbf{x})$ enforces

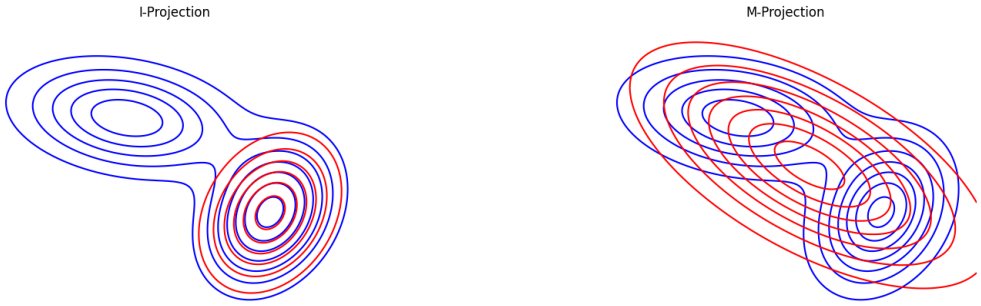


Figure 2.2.: Shown is an example of the described I- and M-Projection. The red contour depicts a uni-modal model $q(\mathbf{x})$ on top of a bi-modal target distribution $p(\mathbf{x})$. The KL is optimized w.r.t. $q(\mathbf{x})$. The figure is inspired by a figure presented in [3].

an over-estimation of the target distribution $p(\mathbf{x})$. This effect comes from the log-term in the integral of the divergence as

$$\log \frac{p(\mathbf{x})}{q(\mathbf{x})} \to \infty \qquad \text{for } q(\mathbf{x}) \to 0,$$

such that $\mathrm{KL}(p(\mathbf{x}) \,\|\, q(\mathbf{x})) \to \infty$, which is also known as *zero avoidence* [52]. Due to asymmetry, taking what is known as the *Information-Projection* (I-Projection) or *reverse* KL $\mathrm{KL}(q(\mathbf{x}) \,\|\, p(\mathbf{x}))$ when optimizing w.r.t. the model $q(\mathbf{x})$ enforces an under-estimation of the target distribution for the same reasons as

$$\log \frac{q(\mathbf{x})}{p(\mathbf{x})} \to \infty \qquad \text{for } p(\mathbf{x}) \to 0.$$

This property is known as *zero forcing* as any point in $\mathcal{R}^D$ where $q(\mathbf{x}) > 0$ while $p(\mathbf{x}) \to 0$ causes $\mathrm{KL}(q(\mathbf{x}) \,\|\, p(\mathbf{x})) \to \infty$ such that $q(\mathbf{x}) \to 0 \,\forall\, \mathbf{x}$ where $p(\mathbf{x}) \to 0$ when optimized. An example of these effects is shown in Figure 2.2 with a bi-modal distribution $p(\mathbf{x})$, represented by a two component Gaussian mixture, and a uni-modal model $q(\mathbf{x}) = \mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}, \boldsymbol{\Lambda})$.

3. Inference

Statistical inference is concerned with reasoning about and drawing conclusions from data, which involves two types of inference. One aims to answer queries about or predict possible states of the world based on some observed data. However, to make predictions, we need to find some underlying truth of the observed data. In inference, we define this truth or nature of the data as a probabilistic model for which we infer parameters such that the model best describes the observed data. This chapter discusses the fundamental ideas and concepts of inferring parameters of an assumed probabilistic model given a set of observed data.

Let $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ be a set of N i.i.d. observations of a random variable $\mathbf{x} \in \mathcal{R}^D$, for which we want to find the generating process. Literature often also refers to these observations as evidence. The generative process imposes some conditional dependency between the different observations, which, in statistical inference, takes on the form of some unknown distribution from which the observations are assumed to be drawn [2, 3]. A flexible concept of describing such distributions and dependencies are *Latent Variable Models* (LVM).

3.1. Latent Variable Models

Let $p(\mathbf{x} \mid \boldsymbol{\theta})$ be a distribution conditioned on a set of unknown, latent variables $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M\}$ [5, 30]. We will refer to these latent variables $\boldsymbol{\theta}$ as parameters of the conditional distribution. However, generally the latent variables are not necessarily parameters of the model but can be any type of additional unknown information we assume exists. A corresponding graphical model showing the relation between the evidence of $\mathbf{x}$ and these latent variables is depicted in Figure 3.1. Optimizing this construct to match the

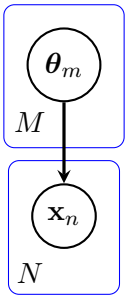


Figure 3.1.: Graphical Model For Latent Variable Models

underlying nature of the evidence means solving the marginal likelihood

$$\begin{aligned} \log p\left(\mathbf{X}\right) &= \log \int p\left(\mathbf{X}, \boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} \\ &= \log \int p\left(\mathbf{X} \,|\, \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta}, \end{aligned} \tag{3.1}$$

where $p\left(\boldsymbol{\theta}\right)$ is a prior over the latent variables $\boldsymbol{\theta}$. Solving this integral is generally intractable, and instead, we need to find the closest approximation.
The easiest and straight forward approaches to approximating this integral, such that the model best describes the underlying nature of the evidence, are ML and MAP estimation

$$\begin{aligned} \boldsymbol{\theta}_{\mathrm{ML}} &= \arg\max_{\boldsymbol{\theta}} p\left(\mathbf{X} \,|\, \boldsymbol{\theta}\right), \\ \boldsymbol{\theta}_{\mathrm{MAP}} &= \arg\max_{\boldsymbol{\theta}} p\left(\mathbf{X} \,|\, \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right), \end{aligned}$$

which correspond to finding a maximizing point estimate of $\boldsymbol{\theta}$ [5]. From a mathematical perspective, finding a single point estimate means finding the single value of $\boldsymbol{\theta}$ for which the integrands contribute the most to the integral and choosing these integrands as an approximation

$$\begin{aligned} \log p\left(\mathbf{X}\right) &\approx \log p\left(\mathbf{X} \,|\, \boldsymbol{\theta}_{\mathrm{ML}}\right), \\ \log p\left(\mathbf{X}\right) &\approx \log p\left(\mathbf{X} \,|\, \boldsymbol{\theta}_{\mathrm{MAP}}\right) p\left(\boldsymbol{\theta}_{\mathrm{MAP}}\right). \end{aligned}$$

Both approaches potentially ignore a majority of the contributing probability mass of the integral. The ML estimation further ignores the prior density $p\left(\boldsymbol{\theta}\right)$, which, for non-uniform priors, means neglecting important prior information [5].

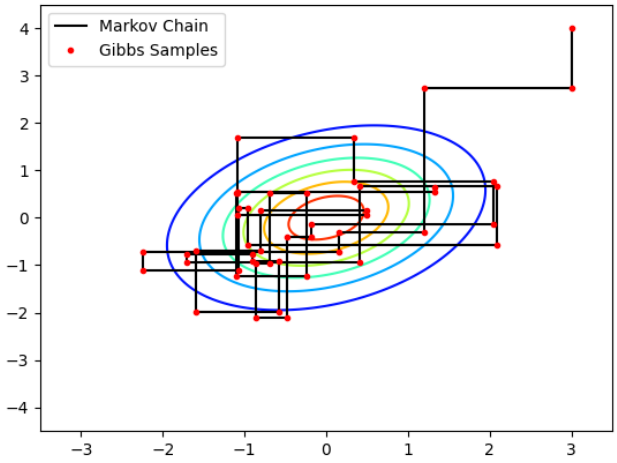


Figure 3.2.: Shown is an example of Gibbs sampling applied to a 2D multivariate Gaussian distribution. Each of the dimensions is treated as a separate random variable. The process is initialized at $\mathbf{x}^{(0)} = (3, 4)$ with the black lines and red dots visualizing the Markov Chain of samples in front of the true distribution.

Other methods that find more accurate approximations of the integral are mainly split into two categories, MCMC sampling techniques, and variational methods. MCMC sampling will only be discussed briefly in the next section for completeness as the focus lies in variational methods.

3.2. Markov Chain Monte-Carlo

MCMC sampling combines the fundamental concepts of Monte-Carlo estimation with Markov Chains. The idea of Monte-Carlo estimation is to estimate the inherent properties of a distribution by analyzing samples drawn from the distribution [50, 53]. In MCMC these samples are assumed to possess the Markov Property, meaning each sample is only depending on the previous sample generating a Markov Chain of samples [8]. In potentially heavily correlated distributions such as posterior distributions of LVMs a common choice of

MCMC technique is *Gibbs* sampling. In theory, Gibbs sampling produces complete samples of a possibly correlated distribution by consecutively sampling every random variable conditioned on the last, fixed sample of all other random variables [7, 8]. Assuming an LVM of the form shown in Figure 3.1 with a marginal probability as described by Equation (3.1) we can approximate the integral by obtaining an estimate of the true posterior. The posterior is some unknown distribution $p(\boldsymbol{\theta} \mid \mathbf{X})$ over the set of random variables $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M\}$ conditioned on the evidence $\mathbf{X}$. To perform Gibbs sampling, we have to define all conditional distributions, of which samples of the separate latent variables are drawn as

$$
\begin{aligned}
\boldsymbol{\theta}_1^{(i)} &\sim p\left(\boldsymbol{\theta}_1 \,\middle|\, \mathbf{X}, \boldsymbol{\theta}_2 = \boldsymbol{\theta}_2^{(i-1)}, ..., \boldsymbol{\theta}_M = \boldsymbol{\theta}_M^{(i-1)}\right), \\
\boldsymbol{\theta}_2^{(i)} &\sim p\left(\boldsymbol{\theta}_2 \,\middle|\, \mathbf{X}, \boldsymbol{\theta}_1 = \boldsymbol{\theta}_1^{(i)}, \boldsymbol{\theta}_3 = \boldsymbol{\theta}_3^{(i-1)}, ..., \boldsymbol{\theta}_M = \boldsymbol{\theta}_M^{(i-1)}\right), \\
&\vdots \\
\boldsymbol{\theta}_M^{(i)} &\sim p\left(\boldsymbol{\theta}_M \,\middle|\, \mathbf{X}, \boldsymbol{\theta}_1 = \boldsymbol{\theta}_1^{(i)}, ..., \boldsymbol{\theta}_{M-1} = \boldsymbol{\theta}_{M-1}^{(i)}\right),
\end{aligned}
$$

where $\boldsymbol{\theta}_m^i$ is the new sample obtained in iteration i. The process is initialized by obtaining a single sample from the prior distribution $p(\boldsymbol{\theta})$. A few iterations between each sample have to be made to produce a set of i.i.d. samples, as the samples are correlated through a Markov Chain [7]. Given enough iterations, a sample $\boldsymbol{\theta}^{(n)} = \left\{\boldsymbol{\theta}_1^{(n)}, ..., \boldsymbol{\theta}_M^{(n)}\right\}$ will eventually match a sample of the true posterior distribution [6, 7]. However, reaching this point may require a large number of iterations, which is why in high dimensional distributions, Gibbs sampling is quite slow. Further, defining the conditional distributions for arbitrarily complex joint distributions may be impossible. An example of Gibbs sampling applied to a $2D$ multivariate Gaussian distribution is given in Figure 3.2, where we try to find true samples of the Gaussian distribution shown in the background from which we have drawn some initial observations.

3.3. Variational Methods

Instead of finding a suitable approximation to the integral directly, variational methods formulate a tractable lower or upper bound to the integral, shifting the optimization to bound optimization. Tightening the bound results in a closer approximation to the true solution [3, 5].

Again, let $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ be a set of i.i.d. observations from a random variable x, which we are assuming is distributed according to an LVM of the form

$$\log p(\mathbf{X}) = \log \int p(\mathbf{X} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta},$$

where $\boldsymbol{\theta}$ represents the parameters of the model. Further, we will introduce an additional set of latent variables $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$. The problem thus changes into

$$\log p(\mathbf{X}) = \log \int \int p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})\, \mathrm{d}\mathbf{Z}\mathrm{d}\boldsymbol{\theta}, \tag{3.2}$$

where each latent variable $\mathbf{z}_n$ in $\mathbf{Z}$ corresponds to an observation $\mathbf{x}_n$.

3.3.1. Expectation-Maximization

First, we will consider the problem of maximum likelihood for which Equation (3.2) simplifies to

$$\log p(\mathbf{X} \mid \boldsymbol{\theta}) = \log \int p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})\, \mathrm{d}\mathbf{Z} = \sum_{n=1}^{N} \log \int p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\theta})\, \mathrm{d}\mathbf{z}_n,$$

as we are not taking the prior over the unknown parameters $\boldsymbol{\theta}$ into account. The goal is to find a point estimate $\boldsymbol{\theta}_{\mathrm{ML}}$ that maximizes the log-likelihood $\mathcal{L}(\boldsymbol{\theta}) = \log p(\mathbf{X} \mid \boldsymbol{\theta})$ [2, 3, 5]. Due to the existence of the unobserved variables $\mathbf{Z}$ direct optimization of $\boldsymbol{\theta}$ is difficult as $\mathbf{Z}$ induce new dependencies. Further, solving the integral may be intractable [5]. However, if $\mathbf{z}_1, ..., \mathbf{z}_N$ were to be observed the problem would again be solvable, which is achieved by introducing auxiliary distributions $q(\mathbf{Z}) = \{q(\mathbf{z}_n)\}_{n=1}^{N}$, often also referred to as variational distributions [3], such that

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log \int q(\mathbf{z}_n)\, \frac{p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\theta})}{q(\mathbf{z}_n)} \mathrm{d}\mathbf{z}_n,$$

where $q(\mathbf{Z})$ represents the posterior over $\mathbf{Z}$ and $p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\theta})$ is the complete-data likelihood function [3]. The name of complete-data comes from the idea that the observations are assumed incomplete and that $\mathbf{z}_n$ is some missing information in the observation $\mathbf{x}_n$.

Thus, $\mathbf{x}_n$ and $\mathbf{z}_n$ combined are a complete data point with $\mathbf{z}_n$ being an augmentation of the observed data. We then define a lower bound on $\mathcal{L}(\boldsymbol{\theta})$ as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &\geq \sum_{n=1}^{N} \int q(\mathbf{z}_n) \log \frac{p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\theta})}{q(\mathbf{z}_n)} \mathrm{d}\mathbf{z}_n, \\ &= \sum_{n=1}^{N} \left[\int q(\mathbf{z}_n) \log p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\theta}) \, \mathrm{d}\mathbf{z}_n - \int q(\mathbf{z}_n) \log q(\mathbf{z}_n) \, \mathrm{d}\mathbf{z}_n \right], \\ &= \mathcal{F}(q(\mathbf{Z}), \boldsymbol{\theta})\end{aligned}$$

using *Jensen's inequality* [54] for concave functions [2, 3, 5, 11, 13]. This lower bound is known as the *Evidence Lower Bound* (ELBO) or *variational free energy* [2, 3, 5].

Optimizing the ELBO follows an sequential procedure known as *Expectation-Maximization* (EM) [32] or *Baum-Welch algorithm* [31], where an inference step (E-Step) and a maximization step (M-Step) are alternated [3]. For some parameters $\boldsymbol{\theta}^{(i)}$ at step i one first performs the E-Step, inferring a new posterior distribution $q(\mathbf{Z})^{(i+1)}$ such that

$$q(\mathbf{Z})^{(i+1)} = \arg\max_{q(\mathbf{Z})} \mathcal{F}\left(q(\mathbf{Z}), \boldsymbol{\theta}^{(i)}\right),$$

which is optimal when $q(\mathbf{Z}) = p\left(\mathbf{Z} \,\middle|\, \mathbf{X}, \boldsymbol{\theta}^{(i)}\right)$. Afterwards, the M-Step maximizes the lower bound w.r.t. to $\boldsymbol{\theta}$

$$\begin{aligned}\boldsymbol{\theta}^{(i+1)} &= \arg\max_{\boldsymbol{\theta}} \mathcal{F}\left(q(\mathbf{Z})^{(i+1)}, \boldsymbol{\theta}\right) \\ &= \arg\max_{\boldsymbol{\theta}} \int p\left(\mathbf{Z} \,\middle|\, \mathbf{X}, \boldsymbol{\theta}^{(i)}\right) \log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}),\end{aligned}$$

where we substitute $q(\mathbf{Z}) = p\left(\mathbf{Z} \,\middle|\, \mathbf{X}, \boldsymbol{\theta}^{(i)}\right)$ with $\boldsymbol{\theta}^{(i)}$ being kept fixed in the posterior [3, 5].

We can get a better understanding of the optimization by looking at a slightly different perspective of this problem. The goal was to maximize the log-likelihood $\mathcal{L}(\boldsymbol{\theta}) = \log p(\mathbf{X} \mid \boldsymbol{\theta})$ w.r.t. to $\boldsymbol{\theta}$, which might not be possible if the data is incomplete. Instead we defined a

complete data likelihood $p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\theta})$ to optimize over. Starting from the ELBO we derive

$$\begin{aligned}
\mathcal{F}(q(\mathbf{Z}), \boldsymbol{\theta}) &= \int q(\mathbf{Z}) \log \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})}{q(\mathbf{Z})} \mathrm{d}\mathbf{Z} \\
&= \int q(\mathbf{Z}) \log \frac{p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})\, p(\mathbf{X} \mid \boldsymbol{\theta})}{q(\mathbf{Z})} \mathrm{d}\mathbf{Z} \\
&= \int q(\mathbf{Z}) \log \frac{p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})}{q(\mathbf{Z})} \mathrm{d}\mathbf{Z} + \int q(\mathbf{Z}) \log p(\mathbf{X} \mid \boldsymbol{\theta}) \, \mathrm{d}\mathbf{Z} \\
&= \int q(\mathbf{Z}) \log \frac{p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})}{q(\mathbf{Z})} \mathrm{d}\mathbf{Z} + \log p(\mathbf{X} \mid \boldsymbol{\theta}) \\
&= \log p(\mathbf{X} \mid \boldsymbol{\theta}) - \mathrm{KL}\left(q(\mathbf{Z}) \,\|\, p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})\right),
\end{aligned}$$

where we applied the Bayes Rule

$$p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) = p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})\, p(\mathbf{X} \mid \boldsymbol{\theta}).$$

By performing the E-Step, we minimize the KL between the variational posterior $q(\mathbf{Z})$ and the true posterior $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})$, closing the gap between the ELBO and the true objective $\log p(\mathbf{X} \mid \boldsymbol{\theta})$. Maximizing the ELBO afterwards in the M-Step for a fixed posterior $q(\mathbf{Z})$ effectively maximizes $\log p(\mathbf{X} \mid \boldsymbol{\theta})$, re-widening the gap between variational and true posterior [3, 5, 11]. This process is illustrated in Figure 3.3.

3.3.2. Variational Bayesian Expectation-Maximization

While the EM-Algorithm is easily extended to a maximum-a-posteriori estimate of the parameters $\boldsymbol{\theta}$, we are not interested in a point estimate but rather in the posterior distribution over the parameters. We will again assume the problem described in Equation (3.2). The ulterior goal remains of maximizing the log-likelihood $\log p(\mathbf{X})$ for a set of N i.i.d. observations $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, for which we need to solve the integral

$$\begin{aligned}
\log p(\mathbf{X}) &= \log \int \int p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta}) \, \mathrm{d}\mathbf{Z} \mathrm{d}\boldsymbol{\theta}, \\
&= \log \int \int p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\theta})\, p(\mathbf{Z} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta}) \, \mathrm{d}\mathbf{Z} \mathrm{d}\boldsymbol{\theta}.
\end{aligned}$$

Following a similar derivation as in the EM-Algorithm we will introduce an auxiliary distribution $q(\mathbf{Z}, \boldsymbol{\theta})$ to approximate the true posterior, obtaining a lower bound on $\log p(\mathbf{X})$

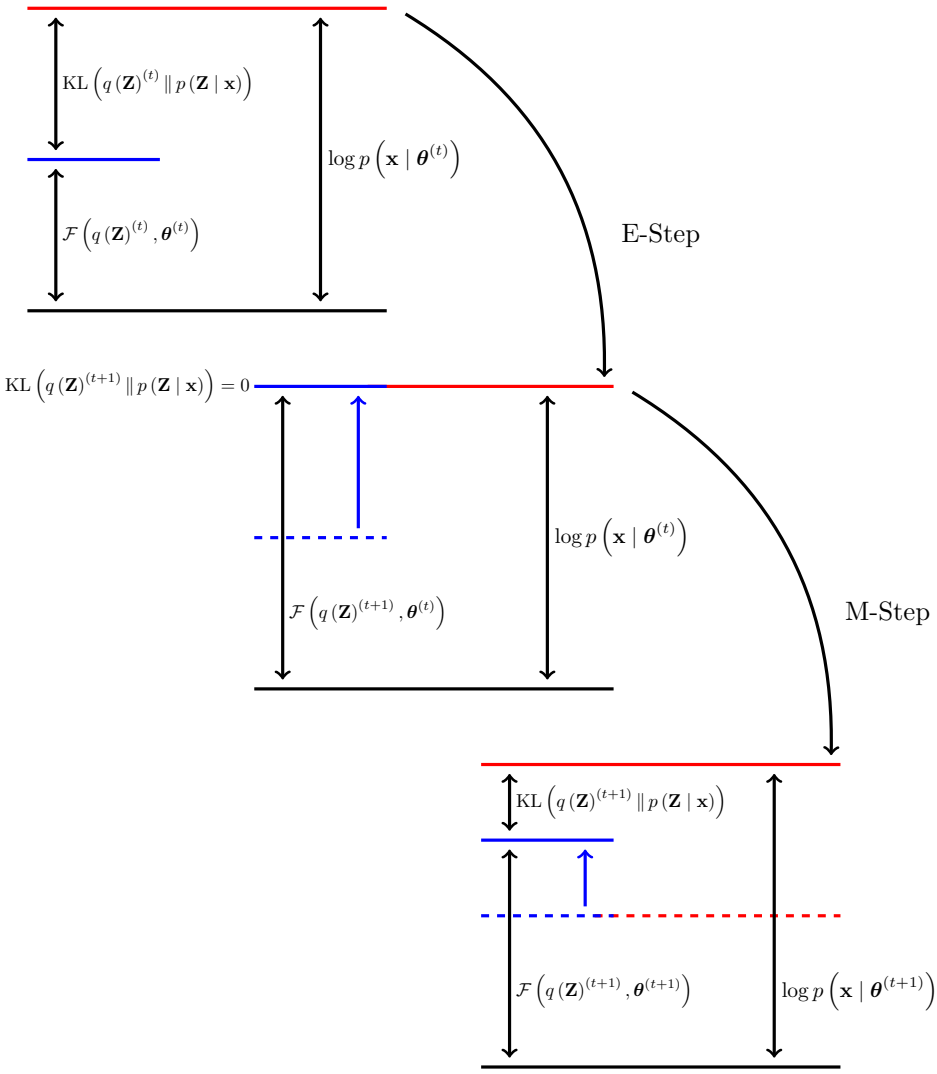


Figure 3.3.: This figure visualizes the EM algorithm. The E-Step minimizes the KL between variational and true posterior such that the lower bound matches the log likelihood $\log p\left(\mathbf{x} \mid \boldsymbol{\theta}\right)$. Through the M-Step the log likelihood is maximized, causing a new discrepancy between variational and true posterior. These steps are repeated until convergence. This figure is created based on three separate figures in [3].

as

$$\begin{aligned}\log p(\mathbf{X}) &= \log \int\int \frac{q(\mathbf{Z},\boldsymbol{\theta})}{q(\mathbf{Z},\boldsymbol{\theta})} p(\mathbf{X},\mathbf{Z},\boldsymbol{\theta})\,\mathrm{d}\mathbf{Z}\mathrm{d}\boldsymbol{\theta}, \\ &\geq \int\int q(\mathbf{Z},\boldsymbol{\theta}) \log \frac{p(\mathbf{X},\mathbf{Z},\boldsymbol{\theta})}{q(\mathbf{Z},\boldsymbol{\theta})}\mathrm{d}\mathbf{Z}\mathrm{d}\boldsymbol{\theta}, \\ &= \mathcal{F}\left(q(\mathbf{Z},\boldsymbol{\theta})\right),\end{aligned}$$

applying Jensen's inequality [2, 3, 5]. We again recognize the relation

$$\log p(\mathbf{X}) = \mathcal{F}\left(q(\mathbf{Z},\boldsymbol{\theta})\right) - \mathrm{KL}\left(q(\mathbf{Z},\boldsymbol{\theta}) \,\|\, p(\mathbf{Z},\boldsymbol{\theta} \,|\, \mathbf{X})\right),$$

such that maximizing the lower bound w.r.t. $q(\mathbf{Z},\boldsymbol{\theta})$ minimizes the KL to the true posterior. However, this time, substituting the approximate posterior with the true posterior when the KL reaches its optimum will not simplify the problem as the true posterior in most models is an intractable distribution [3, 5]. Instead, to find tractable solutions, we have to restrict the approximate posterior $q(\mathbf{Z},\boldsymbol{\theta})$ to some tractable distribution while still being sufficiently flexible to approximate the true posterior appropriately [3].

3.3.3. Mean-Field Inference

A traditional method to the variational posterior is to assume factorized distributions, known as *Mean-Field* inference, which is derived from *Mean-Field Theory* [14] [2, 3]. The approach utilizes the Mean-Field assumption that all random variables from the posterior distribution are partitioned into a set of disjoint groups $\mathbf{Z}_i$. For this definition, we summarize the random variables $\{\mathbf{Z},\boldsymbol{\theta}\}$ as $\mathbf{Z}$, such that

$$q(\mathbf{Z}) = \prod_{\mathbf{Z}_i \in \mathbf{Z}} q_i(\mathbf{Z}_i),$$

where each factor $q_i(\mathbf{Z}_i)$ has no assumed functional form, following the notation in [3]. The Mean-Field assumption introduces independencies between all random variables from different groups $\mathbf{Z}_i$ [3]. The best solution we can find using the Mean-Field assumption is when we find the least amount of disjoint groups such that the problem still remains tractable [3]. For the slightly adjusted problem, where all latent variables and parameters are jointly denoted by $\mathbf{Z}$, we have

$$\log p(\mathbf{X}) = \mathcal{F}\left(q(\mathbf{Z})\right) - \mathrm{KL}\left(q(\mathbf{Z}) \,\|\, p(\mathbf{Z} \,|\, \mathbf{X})\right),$$

with

$$\mathcal{F}\left(q\left(\mathbf{Z}\right)\right) = \int q\left(\mathbf{Z}\right) \log \frac{p\left(\mathbf{X}, \mathbf{Z}\right)}{q\left(\mathbf{Z}\right)} \mathrm{d}\mathbf{Z}.$$

Similar to the EM-algorithm we will alternate between separate updates for each of the factors $q_i\left(\mathbf{Z}_i\right)$ [2,3]. The update rule for a specific factor $q_j\left(\mathbf{Z}_j\right)$ is derived from the lower bound

$$\begin{aligned}
\mathcal{F}\left(q\left(\mathbf{Z}\right)\right) &= \int \log \frac{p\left(\mathbf{X}, \mathbf{Z}\right)}{\prod_i q_i} \prod_i q_i \mathrm{d}\mathbf{Z} \\
&= \int \log p\left(\mathbf{X}, \mathbf{Z}\right) \prod_i q_i \mathrm{d}\mathbf{Z} - \int \sum_i \log q_i \prod_i q_i \mathrm{d}\mathbf{Z} \\
&= \int q_j \int \log p\left(\mathbf{X}, \mathbf{Z}\right) \prod_{i \neq j} q_i \mathrm{d}\mathbf{Z}_i \mathrm{d}\mathbf{Z}_j \\
&\quad - \int q_j \left[\log q_j + \int \sum_{i \neq j} \log q_i \prod_{i \neq j} q_i \mathrm{d}\mathbf{Z}_i \right] \mathrm{d}\mathbf{Z}_j \\
&= \int q_j \log \tilde{p}\left(\mathbf{X}, \mathbf{Z}_j\right) \mathrm{d}\mathbf{Z}_j - \int q_j \log q_j \mathrm{d}\mathbf{Z}_j + C_j \\
&= -\operatorname{KL}\left(q_j \,\|\, \tilde{p}\left(\mathbf{X}, \mathbf{Z}_j\right)\right) + C_j,
\end{aligned}$$

while keeping all other factors fixed, where $q_j = q_j\left(\mathbf{Z}_j\right)$ and $q_i = q_i\left(\mathbf{Z}_i\right)$ with an auxiliary distribution

$$\begin{aligned}
\log \tilde{p}\left(\mathbf{X}, \mathbf{Z}_j\right) &= \int \log p\left(\mathbf{X}, \mathbf{Z}\right) \prod_{i \neq j} q_i \mathrm{d}\mathbf{Z}_i \\
&= \mathbb{E}_{\prod_{i \neq j} q_i}\left[\log p\left(\mathbf{X}, \mathbf{Z}\right)\right].
\end{aligned}$$

Further, C_j represents a constant that corresponds to the normalization constant of the updated factor q_j and can be retrieved if necessary [3]. Thus maximizing the lower bound w.r.t. q_j means minimizing the negative KL between q_j and the auxiliary distribution. The optimum of the KL defines the update for q_j as

$$\begin{aligned}
q_j\left(\mathbf{Z}_j\right) &= C_j \exp \mathbb{E}_{\prod_{i \neq j} q_i}\left[\log p\left(\mathbf{X}, \mathbf{Z}\right)\right] \\
&= \frac{\exp \mathbb{E}_{\prod_{i \neq j} q_i}\left[\log p\left(\mathbf{X}, \mathbf{Z}\right)\right]}{\int \exp \mathbb{E}_{\prod_{i \neq j} q_i}\left[\log p\left(\mathbf{X}, \mathbf{Z}\right)\right] \mathrm{d}\mathbf{Z}_j},
\end{aligned}$$

which simply is taking the exponential of the expectation over the full joint distribution $\log p(\mathbf{X}, \mathbf{Z})$ w.r.t. all other, fixed factors with a normalization constant [3].

Returning to the problem defined previously

$$\log p(\mathbf{X}) = \mathcal{F}(q(\mathbf{Z}, \boldsymbol{\theta})) - \mathrm{KL}(q(\mathbf{Z}, \boldsymbol{\theta}) \,\|\, p(\mathbf{Z}, \boldsymbol{\theta} \,|\, \mathbf{X})),$$

we can follow the same derivation by first factorizing the auxiliary distribution $q(\mathbf{Z}, \boldsymbol{\theta}) = q(\mathbf{Z})\, q(\boldsymbol{\theta})$. Adjusting the notation, we define the update of both factors at step i as E- and M-Step

$$q(\mathbf{Z})^{(i+1)} = C_{\mathbf{Z}} \mathbb{E}_{q(\boldsymbol{\theta})^{(i)}} [\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta})], \tag{3.3}$$

$$q(\boldsymbol{\theta})^{(i+1)} = C_{\boldsymbol{\theta}} \mathbb{E}_{q(\mathbf{Z})^{(i+1)}} [\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta})], \tag{3.4}$$

which are guaranteed to converge to a local optimum in a way that the lower bound can never decrease [5].

From this point it is easy to notice the relation of the assumed LVM to the Bayesian Gaussian mixture model derived in the previous chapter (see Section 2.1.3). Let $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ be a set of N i.i.d. observations of a random variable $\mathbf{x} \in \mathcal{R}^D$ which we assume are distributed according to a Bayesian Gaussian mixture model as described by Figure 2.1 with conjugate priors. The marginal probability is defined as

$$\begin{aligned}
\log p(\mathbf{X}) &= \log \sum_{\mathbf{Z}} \int\int\int p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) \, \mathrm{d}\boldsymbol{\pi} \mathrm{d}\boldsymbol{\mu} \mathrm{d}\boldsymbol{\Lambda} \\
&= \log \sum_{\mathbf{Z}} \int\int\int p(\mathbf{X} \,|\, \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})\, p(\mathbf{Z} \,|\, \boldsymbol{\pi})\, p(\boldsymbol{\pi})\, p(\boldsymbol{\mu} \,|\, \boldsymbol{\Lambda})\, p(\boldsymbol{\Lambda}) \, \mathrm{d}\boldsymbol{\pi} \mathrm{d}\boldsymbol{\mu} \mathrm{d}\boldsymbol{\Lambda},
\end{aligned}$$

where the joint probability factorizes according to the graphical model. Introducing a factorized variational posterior $q(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\mathbf{Z})\, q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ in accordance with the Mean-Field assumption, we recognize that $q(\mathbf{Z})\, q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\mathbf{Z})\, q(\boldsymbol{\theta})$ with $\theta = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\}$ and $\mathbf{Z}$ now being discrete. Therefore, we can follow the update equations in Equation (3.3) to define an E-Step over the assignment variable in step i as

$$\begin{aligned}
\log q(\mathbf{Z})^{(i+1)} &= \mathbb{E}_{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}} [\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] + C_{\mathbf{Z}}, \\
&= \mathbb{E}_{\boldsymbol{\pi}} [\log p(\mathbf{Z} \,|\, \boldsymbol{\pi})] + \mathbb{E}_{\boldsymbol{\mu}, \boldsymbol{\Lambda}} [\log p(\mathbf{X} \,|\, \boldsymbol{\mu}, \boldsymbol{\Lambda})] + C_{\mathbf{Z}},
\end{aligned}$$

where all terms independent of $\mathbf{Z}$ were absorbed by $C_{\mathbf{Z}}$ and with $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \sim q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ [3]. Due to the model being a Gaussian mixture with conjugate priors in conjunction with

the factorization of q there exists a closed-form analytical solution to both update steps. Replacing the remaining factors of the joint distribution in the E-Step with their density functions leads to

$$\begin{aligned}
\log q\left(\mathbf{Z}\right)^{(i+1)} =& \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\Lambda}}\left[\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\left(\frac{1}{2}\left(\mathbf{x}_n-\boldsymbol{\mu}_k\right)^T\boldsymbol{\Lambda}_k\left(\mathbf{x}_n-\boldsymbol{\mu}_k\right)+\log\sqrt{\left(2\pi\right)^D\left|\boldsymbol{\Lambda}_k^{-1}\right|}\right)\right] \\
&+\mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\log\pi_k\right]+C_{\mathbf{Z}} \\
=&\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\left[\mathbb{E}_{\pi_k}\left[\log\pi_k\right]+\frac{1}{2}\mathbb{E}_{\boldsymbol{\Lambda}_k}\left[\log\left|\boldsymbol{\Lambda}_k\right|\right]-\frac{D}{2}\log 2\pi\right. \\
&\left.-\frac{1}{2}\mathbb{E}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}\left[\left(\mathbf{x}_n-\boldsymbol{\mu}_k\right)^T\boldsymbol{\Lambda}_k\left(\mathbf{x}_n-\boldsymbol{\mu}_k\right)\right]\right]+C_{\mathbf{Z}} \\
=&\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\log\rho_{nk}+C_{\mathbf{Z}},
\end{aligned}$$

such that the update of $q\left(\mathbf{Z}\right)$ is defined in terms of ρ [2, 3]. It follows the updated distribution described as

$$q\left(\mathbf{Z}\right)^{(i+1)}=\prod_{n=1}^{N}\prod_{k=1}^{K} r_{nk}^{z_{nk}},$$

with the normalization constant defined in a way that

$$r_{nk}=\frac{\rho_{nk}}{\sum_{j=1}^{K}\rho_{nj}},$$

which corresponds to the responsibility the k^{th} component of the mixture has in generating the n^{th} observation [2, 3]. Noticing that the responsibilities are subject to the same constraints as the parameters $\boldsymbol{\pi}$ we see that $q\left(\mathbf{Z}\right)$ is again a Categorical distribution with parameters $\mathbf{r}_n=\{r_{n1},...,r_{nK}\}$ [2, 3].

Similarly, we find a closed-form update for the parameters in an M-Step from

$$\begin{aligned}
\log q\left(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}\right)^{(i+1)} =& \mathbb{E}_{\mathbf{Z}}\left[\log p\left(\mathbf{X},\mathbf{Z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}\right)\right]+C_\theta \\
=&\mathbb{E}_{\mathbf{Z}}\left[\log p\left(\mathbf{Z}\,|\,\boldsymbol{\pi}\right)\right]+\mathbb{E}_{\mathbf{Z}}\left[\log p\left(\mathbf{X}\,|\,\boldsymbol{\mu},\boldsymbol{\Lambda}\right)\right] \\
&+\log p\left(\boldsymbol{\pi}\right)+\log p\left(\boldsymbol{\mu}\,|\,\boldsymbol{\Lambda}\right)+\log p\left(\boldsymbol{\Lambda}\right)+C_\theta,
\end{aligned}$$

where the prior distributions are independent of $\mathbf{Z}$. Noticing that there are no terms with both $\boldsymbol{\pi}$ and $\boldsymbol{\mu},\boldsymbol{\Lambda}$ at the same time, we can conclude that $q\left(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}\right)$ further factorizes into $q\left(\boldsymbol{\pi}\right)q\left(\boldsymbol{\mu},\boldsymbol{\Lambda}\right)$ such that $q\left(\boldsymbol{\pi}\right)$ and $q\left(\boldsymbol{\mu},\boldsymbol{\Lambda}\right)$ can be updated separately [2, 3]. Considering

only the terms involving $\boldsymbol{\pi}$ and again supplementing the functional form of the different factors in the joint distribution, we obtain

$$\begin{aligned}\log q\left(\boldsymbol{\pi}\right)^{(i+1)} &= \log \prod_{k=1}^{K} \pi_k^{(\alpha_k - 1)} + \log \prod_{k=1}^{K} \prod_{n=1}^{N} \pi_k^{r_{nk}} + C_{\boldsymbol{\pi}}, \\ &= \log \prod_{k=1}^{K} \pi_k^{\left(\alpha_k + \sum_{n=1}^{N} r_{nk} - 1\right)} + C_{\boldsymbol{\pi}},\end{aligned}$$

for which we recover the functional form of the Dirichlet distribution. This Dirichlet distribution is parameterized by

$$\alpha = \left\{ \alpha_k + \sum_{n=1}^{N} r_{nk} \right\}_{k=1}^{K},$$

where the responsibilities computed in the E-Step represent the expectation over $\mathbf{Z}$ [2, 3]. For $q\left(\boldsymbol{\mu}, \boldsymbol{\Lambda}\right)$ we obtain a further factorization into $\prod_{k=1}^{K} q\left(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right)$ by recognizing that all factors appearing in the update equation that depend on $\boldsymbol{\mu}, \boldsymbol{\Lambda}$ are sums over the mixture components such that

$$\log q\left(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right)^{(i+1)} = \log p\left(\boldsymbol{\Lambda}_k\right) + \log p\left(\boldsymbol{\mu}_{\mathbf{k}} \,|\, \boldsymbol{\Lambda}_k\right) + \sum_{n=1}^{N} r_{nk} \log p\left(\mathbf{x}_n \,\middle|\, \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right) + C_{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k},$$

which again is a Gaussian-Wishart distribution whose parameters are defined by a weighted Bayesian update [2, 3] using Equation (2.3) with

$$\begin{aligned}N_k &= \sum_{n=1}^{N} r_{nk}, \\ \bar{\mathbf{x}}_k &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \mathbf{x}_n, \\ \mathbf{S}_k &= \sum_{n=1}^{N} r_{nk} \left(\mathbf{x}_n - \bar{\mathbf{x}}_k\right)^T \left(\mathbf{x}_n - \bar{\mathbf{x}}_k\right).\end{aligned}$$

This derivation shows that we get closed-form analytical solutions for all factors for a Bayesian Gaussian mixture model in conjunction with the Mean-Field assumption. While appealing, these solutions are only possible by introducing independence between the

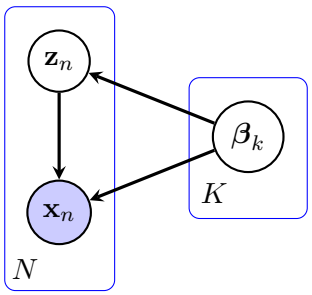


Figure 3.4.: This figure shows the graphical model considered in [34, 35]. The blue colored node $\mathbf{x}_n$ represents the n^{th} observation as a known variable. For each observation a set of local latent variables $\mathbf{z}_n$ is introduced. Observations and local latent variables are dependent on a set of global latent variables $\boldsymbol{\beta} = \{\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K\}$.

assignments $\mathbf{Z}$ and the mixture's parameters $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}$. If we consider that the Categorical distribution, as the prior on these assignments, is not independent of the parameters, we must conceit that the Mean-Field assumption potentially causes significant approximation errors. This problem is not a specific case of the Bayesian Gaussian mixture but is a cause for potential approximation errors independent of the model. Still, since Gibbs sampling is slow in high dimensional problems, Mean-Field inference offers a viable alternative.

While we have considered Mean-Field as a separation into disjoint groups of the random variables, this point of view is per definition already considered a *Structured Mean-Field* [15] approach as naive Mean-Field considers fully factorized distributions [2, 16, 33]. In [55] Structured Mean-Field is presented for two types of substructures. The first substructure represents the approach we have discussed for deriving Mean-Field inference by defining disjoint groups of random variables, where only dependencies between the variables of the same group are retained in the variational distribution. The other type describes substructures, where the groups of variables might not be entirely disjoint, meaning there is some dependency between the different groups remaining. This dependency is typically induced by some global variable on which all groups depend. An example of such a structure is a hierarchical model where all groups depend on some global random variable drawn from a prior distribution [56].

The approach we want to focus on here is presented in the context of *Stochastic Variational Inference* [57]. Given exceedingly large datasets computing the expectations comprising

the lower bound becomes prohibitively expensive [57]. SVI is proposed as an approach for variational inference with such large datasets by repeatedly subsampling mini-batches from the dataset to compute noisy estimates of the expectations for which stochastic optimization is performed [57]. In [34, 35] SVI is extended to Structured Mean-Field for graphical models of the form shown in Figure 3.4. Let $\boldsymbol{\beta} = \{\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K\}$ be a set of K global latent variables and $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$ be a set of N local latent variables, where each variable may be comprised of a subset of multiple variables. The local variables are in correspondence to a set $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ of N i.i.d. observations of a random variable $\mathbf{x}$. Further, we define a model

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\beta}),$$

where all local variables $\mathbf{Z}$ are conditionally independent of each other given the global variables $\boldsymbol{\beta}$ [34]. The objective is to maximize the marginal probability for the observations

$$p(\mathbf{X}) = \int \int p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\beta}) \, \mathrm{d}\boldsymbol{\beta} \mathrm{d}\mathbf{Z},$$

where, without loss of generality, we have assumed that both sets of latent variables are sets of continuous variables. In cases where either variable is discrete, the integral is replaced by a sum. While Mean-Field inference would assume a factorized variational posterior

$$q(\boldsymbol{\beta}, \mathbf{Z}) = \left(\prod_{k=1}^{K} q(\boldsymbol{\beta}_k) \right) \prod_{n=1}^{N} q(\mathbf{z}_n),$$

where $q(\boldsymbol{\beta}, \mathbf{Z})$ is fully factorized, [34, 35] propose a more structured factorization of the form

$$q(\boldsymbol{\beta}, \mathbf{Z}) = \left(\prod_{k=1}^{K} q(\boldsymbol{\beta}_k) \right) \prod_{n=1}^{N} q(\mathbf{z}_n \mid \boldsymbol{\beta}).$$

Here, the local latent variables are no longer assumed independent from each other but are only conditionally independent given the global latent variables as described by $p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\beta})$ [34]. While this approach recovers some dependency between the local latent variables there is still a full factorization between all global variables to remain tractable [34], which, for models where $\boldsymbol{\beta}$ are not independent of each other, might be detrimental.

We have discussed the general idea of Bayesian inference in LVMs, highlighting the different approaches to approximate the marginal distribution described by the LVM. We described

the idea behind Gibbs sampling as a representative for MCMC methods, pointed out its strength of finding true posterior samples given enough time and its weakness of being slow in high dimensional problems. Further, we have thoroughly discussed the concept of variational methods deriving the EM-algorithm from a Bayesian perspective, first for maximum likelihood point estimates of the parameters and afterward for full posterior distributions over the parameters and assignment variables. In this context, we derived Mean-Field inference for general LVMs and specifically derived closed-form solutions for Bayesian Gaussian mixture models. Next, we have described how Structured Mean-Field alleviates the problem of approximation errors induced by the Mean-Field assumption, however still requiring some potentially devastating independence assumption to remain tractable.

While Mean-Field inference offers far better viability in high dimensional problems than Gibbs sampling, our goal is to find a variational posterior over the parameters of the assumed model, for which we will consider a Bayesian Gaussian mixture model without any independence assumptions. To achieve this goal, we will consider a parametric model of the form

$$q(\mathbf{Z}) = q(\mathbf{Z} \mid \boldsymbol{\omega}),$$

where $\boldsymbol{\omega}$ is a set of parameters. While Mean-Field inference induces independence by factorizing to restrict the variational posterior to a tractable form, it does so without assuming any functional form of these factors. Instead, the functional form of the factors was derived from the lower bound. In contrast, the parametric model restricts the variational posterior to a tractable form by predefining some functional form the posterior is restricted to [3]. However, we no longer have to assume independence between the random variables of the variational posterior. The quality of the approximation relies entirely on the representational power of the parametric model.

4. Normalizing Flows As Parametric Generative Models

Applying variational inference to relevant real-world applications often requires highly complex models not only for approximate posterior but also for density estimation tasks. Representing these complex distributions, however, while remaining tractable for inference is challenging. In the previous chapter, we have discussed the idea of parametric models in variational inference. A concept of parametric generative models that gained a lot of attention in modern variational inference is *Normalizing Flows*.

4.1. Normalizing Flows

Normalizing flows represent a form of generative models that originated in the context of density estimation from Tabak and Vanden-Eijnden in 2010 [17]. The idea behind Normalizing Flows is to describe complex distributions using a base distribution in combination with parameterized, invertible, and differentiable transformations [18, 26]. While typical choices for the base distribution are the Gaussian or Uniform distribution, one is not restricted to either one of the two. Rather the base distribution can be any form of distribution with a tractable probability density function from which one can draw samples. In practice, the base distribution acts as some form of a prior on the resulting complex distribution, which can potentially simplify the following optimization of the transformation [26].

Let $\mathbf{x} \in \mathbb{R}^D$ be a sample from a random variable drawn from the base distribution whose probability density function q_x is known and tractable. Further, we define an invertible, and differentiable transformation $g(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^D$ mapping samples from the base distribution q_x to samples of a more complex distribution q_y, such that

$$\mathbf{x} \sim q_x(\mathbf{x}) \rightarrow \mathbf{y} = f(\mathbf{x}) \sim q_y(\mathbf{y}).$$

With f being the inverse of g and following the *change of variables* formula [58] we can define the probability density function of the complex distribution q_y as

$$\begin{aligned} q_y(\mathbf{y}) &= q_x(\mathbf{x}) \left| \det \mathbf{J}_g(\mathbf{x}) \right|^{-1}, \\ &= q_x(f(\mathbf{y})) \mid \det \mathbf{J}_f(\mathbf{y}) \mid, \end{aligned} \tag{4.1}$$

where $\mathbf{J}_g(\mathbf{x})$ and $\mathbf{J}_f(\mathbf{y})$ are the respective Jacobians of the transformation and its inverse with [18, 25, 26]

$$\mathbf{J}_g(\mathbf{x}) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_D}{\partial x_1} & \cdots & \frac{\partial g_D}{\partial x_D} \end{bmatrix}.$$

An intuitive way of imagining a transformation is to think of a function that expands and contracts the space of the base distribution q_x to warp and twist it into the shape of the complex distribution q_y [25]. In this context, the absolute determinant of the Jacobian is a measure for the relative change of volume around a sample $\mathbf{x}$ caused by the transformation [25].

As Normalizing Flows represent a form of generative models, literature often refers to $\mathbf{y} = g(\mathbf{x})$ as the generative direction with g being used to generate data points from the complex distribution by sampling the base distribution and transforming the samples. The inverse function or *flow* f is instead seen as the normalizing direction as it maps the complex distribution onto a simple, "more normal" base distribution [26]. The only restriction imposed on the transformation is for it to be a *diffeomorphism*, which is a bijective and differentiable function whose inverse is differentiable as well [25, 26, 59]. Under this restriction, we may construct arbitrarily complex distributions from any base distribution by defining an arbitrarily complex transformation as shown formally in [60, 61] [25, 26]. However, as we are interested in finding a tractable model for the complex distribution $q_y(\mathbf{y})$, we have to find transformations whose inverse and Jacobian are easy to compute [25, 26]. A key property that is exploited in the construction of such transformations is composability.

Let $g_1, ..., g_L$ be a set of invertible and differentiable transformations, then their composition

$$g = g_L \circ ... \circ g_2 \circ g_1,$$

is also invertible and differentiable. The inverse of such a composition is similarly defined as

$$f = g^{-1} = g_L^{-1} \circ ... \circ g_2^{-1} \circ g_1^{-1}.$$

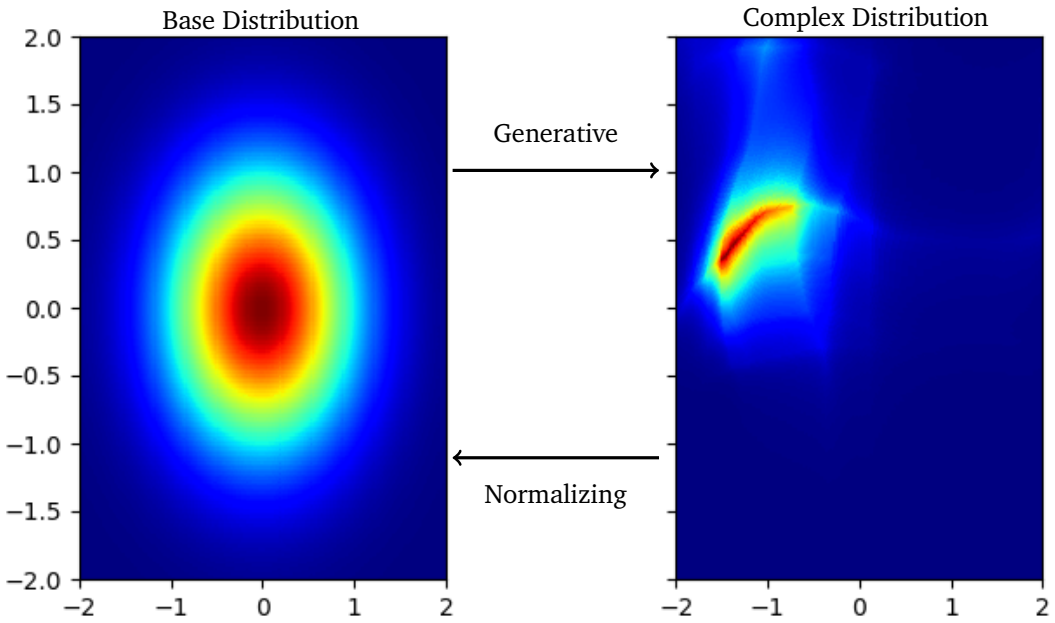


Figure 4.1.: This figure visualizes the generative and normalizing direction. Samples from the base distribution are transformed into samples from the complex distribution forming a generative model. Its inverse normalizes the complex samples to fit a "more normal" base distribution.

More importantly, the determinant of the Jacobian is given by

$$\det \mathbf{J}_g(\mathbf{x}) = \det \mathbf{J}_{g_1}(\mathbf{x}) \prod_{l=2}^{L} \det \mathbf{J}_{g_l}(g_{l-1} \circ ... \circ g_1(\mathbf{x})). \tag{4.2}$$

This composability allows for complex transformations to be represented through a set of simpler transformations [25,26]. Inherently this also means, for a complex distribution q_y, that is described by the same set of transformations g, we can easily compute the density $q_y(\mathbf{y})$ as long as we can easily compute the determinant of the Jacobian of every single transformation $g_1, ..., g_L$.

To summarize, a Normalizing Flow transforms a simple base distribution, e.g., a standard Gaussian, into a much more complex distribution of the same dimensions through a set of bijective and differentiable transformations whose inverses are differentiable as well. A Normalizing Flow is thereby described as

$$\begin{aligned} \mathbf{x} \sim q_x, \qquad \mathbf{y} &= g_L \circ ... \circ g_1(\mathbf{x}), \\ \log q_y(\mathbf{y}) &= \log q_x(f(\mathbf{y})) - \sum_{l=1}^{L} \log \mid \det \mathbf{J}_{g_l}(y_{l-1}) \mid , \end{aligned} \tag{4.3}$$

where $\mathbf{x} = f(\mathbf{y})$ is the inverse of g.

4.2. Bijective And Differentiable Transformations

Finding transformations that meet the requirements we have discussed thus far is an ongoing research topic. This section, will roughly discuss the designs of several approaches, highlighting certain properties required for application in variational inference. A more comprehensive review of various transformations is given in [25, 26]. The following transformations are described in terms of the generative direction.

The most simplistic forms of transformations are elementwise transformations. Let a transformation $g : \mathcal{R}^D \to \mathcal{R}^D$ be defined by a set of scalar, bijective, non-linear functions

$$g(\mathbf{x}) = (h_1(x_1), h_2(x_2), ..., h_D(x_D))^T,$$

where $h_d(x_d) : \mathcal{R} \to \mathcal{R}$ maps the d^{th} input dimension to the d^{th} output dimension [26]. The inverse is defined by separately inverting all function h_d to form a similar set

while the Jacobian is a diagonal matrix. The big disadvantage, however, of elementwise transformations is the separate transformation of every dimension which prevents any potential inter-dimensional dependencies [26]. While limited in representational power, we can construct transformations with inter-dimensional dependencies as linear functions of the form

$$g(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b},$$

with $\mathrm{A} \in \mathcal{R}^{D \times D}$ and $\mathbf{b} \in \mathcal{R}^{D}$ [25, 26].

To represent complex posterior distributions, neither elementwise nor linear transformation are sufficient as the posterior distribution may contain a set of several multidimensional latent variables that require inter-dimensional dependencies while being arbitrarily complex for which linear transformations are too limited.

A concept that is not widely used in practice are planar, and radial Flows for their limited access of the inverse transformation [18, 26]. However, optimizing an approximate posterior distribution does not necessarily require the computation of the inverse. Let a planar transformation be defined as

$$g(\mathbf{x}) = \mathbf{x} + \mathbf{u}h\left(\mathbf{w}^T\mathbf{x} + b\right),$$

with $\mathbf{u} \in \mathcal{R}^{D}$, $\mathbf{w} \in \mathcal{R}^{D}$, $b \in \mathcal{R}$ and h being an elementwise, non-linear function [18]. The inverse of such transformations does not have a closed-form solution and might not even exist [26]. However, the absolute determinant of the Jacobian is easily computed as

$$\begin{aligned} \psi(\mathbf{x}) &= h'\left(\mathbf{w}^T\mathbf{x} + b\right)\mathbf{w}, \\ \left|\det \mathbf{J}_g(\mathbf{x})\right| &= \left|1 + \mathbf{u}^T\psi(\mathbf{x})\right|, \end{aligned}$$

where h' is the derivative of h. Such transformations stretch and compress the base distribution perpendicular to a hyperplane [18]. A similar effect is created around a reference point $\mathbf{x}_0 \in \mathcal{R}^{D}$ using radial transformations defined by

$$g(\mathbf{x}) = \mathbf{x} + \beta h(\alpha, r)(\mathbf{x} - \mathbf{x}_0),$$

with $\alpha \in \mathcal{R}^{+}$, $\beta \in \mathcal{R}$ and

$$\begin{aligned} r &= \left|\mathbf{x} - \mathbf{x}_0\right|, \\ h(\alpha, r) &= 1/(\alpha + r). \end{aligned}$$

The corresponding Jacobian is given as

$$\left|\det \mathbf{J}_g(\mathbf{x})\right| = \left[1 + \beta h(\alpha, r)\right]^{d-1}\left[1 + \beta h(\alpha, r) + \beta h'(\alpha, r)\, r\right],$$

where h' is the derivative of h [18].

While significantly more powerful than linear and elementwise transformations, planar and radial flows still require large sets of successive transformations to represent complex distribution, especially in the posterior estimation, where highly multi-modal distributions are to be expected. Most state-of-the-art approaches, therefore, rely on *coupling functions* [19,24] and *auto-regressive networks* [20,24,62] to define more expressive transformations [26]. Let $\mathbf{x} \in \mathcal{R}^D$ be divided into two disjoint sets $\mathbf{x}_{1:d-1} \in \mathcal{R}^d$ and $\mathbf{x}_{d:D} \in \mathcal{R}^{D-d}$. Further, let $h(\mathbf{x}_{d:D}; \phi) : \mathcal{R}^{D-d} \to \mathcal{R}^{D-d}$ be a bijective function, then a coupling transformation g is defined as

$$g\left(\mathbf{x}\right) = \begin{cases} \mathbf{y}_{1:d-1} = \mathbf{x}_{1:d-1} \\ \mathbf{y}_{d:D} = h\left(\mathbf{x}_{d:D}; \phi\right) \end{cases}, \tag{4.4}$$

with $\phi = \Phi\left(\mathbf{x}_{d:D}\right)$, where Φ is some arbitrary function [19, 24–26] (see Figure 4.2). Typically this function is represented by a fully connected neural network or an auto-regressive network [19, 24]. The Jacobian for coupling transformations is a lower, block triangular matrix of the form

$$\mathbf{J}_g\left(\mathbf{x}\right) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{A} & \mathbf{D} \end{bmatrix}.$$

whose determinant is the product of the diagonal elements, which are the identity matrix $\mathbf{I} \in \mathcal{R}^{d \times d}$ and a diagonal matrix $\mathbf{D} \in \mathcal{R}^{(D-d)\times(D-d)}$ containing the derivatives of h [25,26].

Early approaches of coupling transformations rely on affine mappings, limiting the expressiveness of the transformation [28]. Instead, [24, 28] present monotonic rational spline functions of linear and quadratic order as coupling functions. Spline couplings follow the idea of using piece-wise polynomial functions as coupling functions introduced in [63] [24, 28]. The spline itself serves as an elementwise bijective function with each dimension x_d of $\mathbf{x}$ being transformed separately.

Let a spline be defined by a set of K linear or quadratic rational functions bounded by a set of $K+1$ monotonically increasing points $\left\{\left(x^{(k)}, y^{(k)}\right)\right\}_{k=0}^{K}$ called knots in an interval $\left(x^{(0)}, y^{(0)}\right) = (-B, -B)$ and $\left(x^{(K)}, y^{(K)}\right) = (B, B)$ [24, 28]. Each function k maps the points in the corresponding interval $\left[x^{(k)}, x^{(k+1)}\right]$ known as bin. The derivative of the spline is the derivative of the rational functions inside the bins and some positive scalar $\{\delta(k) > 0\}_{k=0}^{K}$ at the knots. At the bounds of each bin, its function must match the

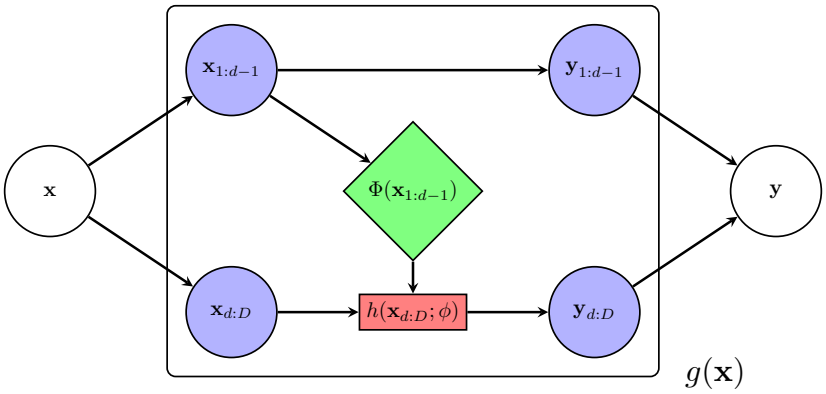


Figure 4.2.: Schematic of a coupling transform. Shown is a single transform $\mathbf{y} = g(\mathbf{x})$ as described by Equation (4.4). Figure inspired by Hadi M. Dolatabadi.

bounding knots while its derivative matches the positive scalars $[\delta(k), \delta(k+1)]$ [24, 28]. In the quadratic case, the derivatives at the boundaries $x^{(0)}$ and $x^{(K)}$ are set to $\delta(0) = \delta(K) = 1$ for numerical stability due to discontinuities at these points [24]. The inverse of both, linear and quadratic couplings has an analytic and easy to compute solution.

In [28] the set of monotonic linear rational functions is defined by following the method described in [64]. To match all required constraints, each bin is further split in two intervals $\left[x^{(k)}, x^{(m)}\right]$ and $\left[x^{(m)}, x^{(k+1)}\right]$, where $x^{(m)} = \left(1 - \lambda^{(k)}\right) x^{(k)} + \lambda^{(k)} x^{(k+1)}$ is an intermediate point with $0 \leq \lambda \leq 1$. Therefore, a linear rational function is described by

$$h\left(\boldsymbol{\zeta}\right) = \begin{cases} \frac{\omega^{(k)} y^{(k)} \left(\lambda^{(k)} - \zeta\right) + \omega^{(m)} y^{(m)} \zeta}{\omega^{(k)} \left(\lambda^{(k)} - \zeta\right) + \omega^{(m)} \zeta} & 0 \leq \zeta \leq \lambda^{(k)} \\ \frac{\omega^{(m)} y^{(m)} (1-\zeta) + \omega^{(k+1)} y^{(k+1)} \left(\zeta - \lambda^{(k)}\right)}{\omega^{(m)} (1-\zeta) + \omega^{(k+1)} \left(\zeta - \lambda^{(k)}\right)} & \lambda^{(k)} \leq \zeta \leq 1 \end{cases},$$

where $\zeta = \left(x - x^{(k)}\right) / \left(x^{(k+1)} - x^{(k)}\right)$ with k being the bin in which x lies [28].

The construction of monotonic quadratic rational splines as proposed in [24] follows a method introduced in [65] for parameterizing the splines. This method constructs the quadratic function of the k^{th} bin as

$$h\left(\zeta\right) = y^{(k)} + \frac{\left(y^{(k+1)} - y^{(k)}\right)\left[s^{(k)}\zeta^2 + \delta(k)\zeta\left(1-\zeta\right)\right]}{s^{(k)}\left[\delta(k+1) + \delta(k) - 2s^{(k)}\right]\zeta\left(1-\zeta\right)},$$

where $\zeta = \left(x - x^{(k)}\right) / \left(x^{(k+1)} - x^{(k)}\right)$ and $s^{(k)} = \left(y^{(k+1)} - y^{(k)}\right) / \left(x^{(k+1)} - x^{(k)}\right)$. For a detailed explanation on the construction of these splines, their derivatives and inverse please refer to [24] and [28].

Taking the concept of coupling transformations even further are auto-regressive flows. Exploiting the *chain rule of probability*, which decomposes a density $p(\mathbf{y})$ with $\mathbf{y} \in \mathcal{R}^D$ into a product of conditionals with each conditional being one-dimensional

$$p(\mathbf{y}) = p(y_1) \prod_{d=2}^{D} p(y_d \,|\, y_{1:d-1}),$$

these flows decompose an input variable $\mathbf{x} \in \mathcal{R}^D$ into D scalars. Each dimension is transformed by a scalar differentiable bijective function $y_d = h(x_d; \phi_d)$ whose parameters are determined by some arbitrary function $\phi_d = \Phi\left(\mathbf{y}_{1:d-1}\right)$ known as conditioner. Therefore, the transformation of the d^{th} dimension of the input variable $\mathbf{x}$ is conditioned on all previous dimensions of the output $\mathbf{y}$ [25, 26, 62]. A schematic of the transform's design is shown in Figure 4.3. By conditioning the output of a transformation on its own lower dimensions, the transformation cannot be computed in parallel and is inherently slow [26]. However, its inverse $x_d = h^{-1}(y_d; \phi_d)$ can be computed efficiently due to the dependency of x_d only on $\phi_d = \Phi\left(\mathbf{y}_{1:d-1}\right)$. Therefore, depending on the direction, the transformation is implemented in, we have either fast sampling or fast evaluation. The Jacobian of the auto-regressive transformation is again a lower triangular whose determinant is the sum of the derivatives of h and, thus, easy to compute. Flows following this design are for example *Neural Auto-Regressive Flows* [21], *Block Neural Auto-Regressive Flows* [27], *Masked Auto-Regressive Flows* [62] and *Inverse Auto-Regressive Flows* [20]. For density estimation, we need a fast normalizing direction to maximize the likelihood of the complex distribution on a fixed set of samples projected onto the base distribution. We can think of density estimation as minimizing $\mathrm{KL}\left(p(\mathbf{x}) \,\|\, q(\mathbf{x})\right)$ w.r.t. $q(\mathbf{x})$ which corresponds to

$$\min_{q(\mathbf{x})} \mathbb{E}_{\mathbf{X}}\left[\log p(\mathbf{x}) - \log q(\mathbf{x})\right] \propto \min_{q(\mathbf{x})} \mathbb{E}_{\mathbf{X}}\left[-\log q(\mathbf{x})\right],$$

where $\mathbf{X}$ is a set of samples from the unknown distribution $p(\mathbf{x})$. In turn, for variational inference, we compute an expectation w.r.t. samples drawn from the variational distribution $q(\mathbf{x})$ which requires a fast generative direction. The required densities $q(\mathbf{x})$ for these samples are computed alongside the transformation following Equation (4.1) and Equation (4.2). Since $q(\mathbf{x})$ does not need to be evaluated on any other samples, fast computation of the inverse or even the existence of an analytically computable inverse is not strictly necessary [26]. Therefore, depending on the application, an auto-regressive

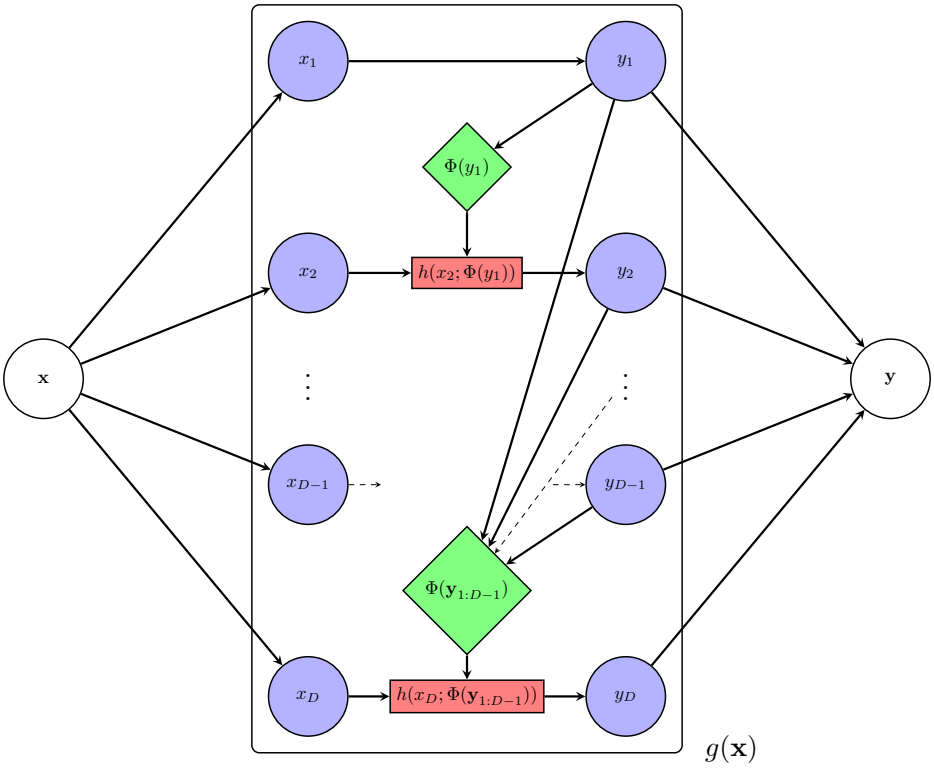


Figure 4.3.: Schematic for auto-regressive coupling transforms. Each dimension d of the output vector $\mathbf{y} \in \mathcal{R}^D$ is a scalar bijective functional mapping h of the corresponding input dimension x_d. The function's parameters are determined through another functional mapping Φ whose input are all previous output dimensions $\mathbf{y}_{1:d-1}$, creating an auto-regressive structure.

transformation must be implemented in the correct direction. The next section will describe variational inference with Normalizing Flows in more detail as a continuation of Chapter 3. However, later in this thesis, we will also consider a concept for updating mixtures of Normalizing Flows, where both directions need to be easy to compute.

4.3. Variational Inference With Normalizing Flows

Applying Normalizing Flows to variational inference is straightforward. Let $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ be a set of N i.i.d. observations of a continuous random variable $\mathbf{x}$. We will assume that this data is distributed according to a Latent Variable Model as described by Figure 3.1 for which we define the marginal log-probability

$$\log p\left(\mathbf{X}\right) = \log \int p\left(\mathbf{X} \mid \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta}.$$

Introducing a variational distribution $q\left(\boldsymbol{\theta} \mid \boldsymbol{\omega}\right)$ we derive the lower bound

$$\begin{aligned} \log p\left(\mathbf{X}\right) &\geq \int q\left(\boldsymbol{\theta} \mid \boldsymbol{\omega}\right) \log p\left(\mathbf{X}, \boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} - \int q\left(\boldsymbol{\theta} \mid \boldsymbol{\omega}\right) \log q\left(\boldsymbol{\theta} \mid \boldsymbol{\omega}\right) \mathrm{d}\boldsymbol{\theta} \\ &= \mathcal{F}\left(q\left(\boldsymbol{\theta} \mid \boldsymbol{\omega}\right)\right), \end{aligned}$$

where $\boldsymbol{\omega}$ denote the parameters of the variational distribution. Using Normalizing Flows gives $q\left(\boldsymbol{\theta} \mid \boldsymbol{\omega}\right)$ a tractable functional form through which we can draw samples which in turn allows for an optimization of the lower bound using *Monte-Carlo* estimates. It follows an approximation of the lower bound as

$$\begin{aligned} \mathcal{F}\left(q\left(\boldsymbol{\theta} \mid \boldsymbol{\omega}\right)\right) &= \mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta} \mid \boldsymbol{\omega})}\left[\log p\left(\mathbf{X}, \boldsymbol{\theta}\right) - \log q\left(\boldsymbol{\theta} \mid \boldsymbol{\omega}\right)\right] \\ &\approx \frac{1}{N_s} \sum_{i=1}^{N_s} \log p\left(\mathbf{X}, \boldsymbol{\theta}^i\right) - \log q\left(\boldsymbol{\theta}^i \mid \boldsymbol{\omega}\right), \end{aligned}$$

where $\boldsymbol{\theta}^i \sim q\left(\boldsymbol{\theta} \mid \boldsymbol{\omega}\right)$ for a total of N_s samples [18]. Typically the approximation is done using a single sample. By computing the gradient $\nabla_{\boldsymbol{\omega}} - \mathcal{F}\left(q\left(\boldsymbol{\theta} \mid \boldsymbol{\omega}\right)\right)$ of the negative lower bound we optimize the objective using *Stochastic Gradient Descent* (SGD) [18]. Supplementing $q\left(\boldsymbol{\theta} \mid \boldsymbol{\omega}\right)$ with the formulation for the complex density described by the

Flow (see Equation (4.3)) leads to

$$\begin{aligned}
-\mathcal{F}\left(q\left(\boldsymbol{\theta} \mid \boldsymbol{\omega}\right)\right) =& \mathbb{E}_{q_0(\mathbf{y}_0)}\left[\log q_L\left(\mathbf{y}_L\right) - \log p\left(\mathbf{X}, \mathbf{y}_L\right)\right] \\
=& \mathbb{E}_{q_0(\mathbf{y}_0)}\left[\log q_0\left(\mathbf{y}_0\right)\right] - \mathbb{E}_{q_0(\mathbf{y}_0)}\left[\log p\left(\mathbf{X}, \mathbf{y}_L\right)\right] \\
& - \mathbb{E}_{q_0(\mathbf{y}_0)}\left[\sum_{l=1}^{L} \log \left|\, \det \mathbf{J}_{g_l}(\mathbf{y}_{l-1}) \,\right|\right],
\end{aligned}$$

where $\mathbf{y}_L = g_L \circ ... \circ g_1(\mathbf{y}_0) = \boldsymbol{\theta}$ is the complex transformed sample and $\mathbf{y}_0 \sim q_0\left(\mathbf{y}_0\right)$ is drawn from the base distribution. Then $\boldsymbol{\omega}$ describes the parameters of the L transformations. We notice that the lower bound can be computed for samples from the base distribution by computing the absolute determinant of the Jacobian along the generative direction such that the normalizing direction does not need to be easily computable for optimizing the lower bound.

The concept of variational inference with Normalizing Flows as parametric variation distributions has been considered mainly in the context of *Variational Auto-Encoders* (VAE) [18, 20, 39]. VAEs are comprised of an encoder and decoder. The encoder $q\left(\mathbf{z} \mid \mathbf{x}\right)$ represents a mapping or encoding of high dimensional samples $\mathbf{x}$ to lower dimensional feature representations $\mathbf{z}$ as in dimensionality reduction methods. In turn the decoder provides the counterpart by reconstructing a high dimensional sample $\hat{\mathbf{x}}$ from a lower dimensional feature representation equivalent to $p\left(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\theta}\right)$, where $\|\mathbf{x} - \hat{\mathbf{x}}\|^2$ is minimized. A more detailed introduction to VAEs is given in Chapter 6. Through this dimensionality reduction, VAEs provide a highly scalable approach for complex high dimensional datasets [18, 39]. However, VAEs define a variational posterior $q\left(\mathbf{z} \mid \mathbf{x}\right)$ as encoder only w.r.t. the lower dimensional feature representation $\mathbf{z}$ considering an additional prior $p\left(\mathbf{z}\right)$. The VAE is optimized w.r.t. the variational posterior over $\mathbf{z}$ and a point estimate of the parameters $\boldsymbol{\theta}$ of the model. The focus of this thesis, instead lies in the posterior over the parameters $\boldsymbol{\theta}$ of mixture models such as the Bayesian Gaussian mixture model.

5. Contribution And Results

So far, we have discussed traditional approaches to variational inference following the Mean-Field assumption and a generic parametric approach with Normalizing Flows. This chapter is split into two different contributions. The first part presents the exact designs and derivations we have considered for learning posterior distributions over the parameters of Bayesian Gaussian mixture models. For each design, experimental results are provided with a comparison of the different designs. Throughout this work, some of the designs have been developed as more sophisticated extensions of one another. In these cases, the reasoning behind the development is described such that the process is understood. The second part of this chapter discusses the concept of mixtures of Normalizing Flows. We describe a concept for separately updating each component in the mixture following an idea described in [36, 37]. While this concept is applicable to variational posterior distributions, the experiments focus on density matching. The experiments shown in this thesis are provided as a proof of concept.

Let $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ be a set of N i.i.d. observations of a continuous random variable $\mathbf{x} \in \mathcal{R}^D$. We will assume that the observations are distributed according to a Bayesian Gaussian mixture model as defined in Chapter 3 of the form

$$p\left(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) = p\left(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) p\left(\mathbf{Z} \mid \boldsymbol{\pi}\right) p\left(\boldsymbol{\pi}\right) p\left(\boldsymbol{\mu} \mid \boldsymbol{\Lambda}\right) p\left(\boldsymbol{\Lambda}\right),$$

where $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$ is a set of one-hot assignment variables $\mathbf{z} \in \mathcal{Z}^K$ with K being the number of components. Further, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K\}$ and $\boldsymbol{\Lambda} = \{\boldsymbol{\Lambda}_1, ..., \boldsymbol{\Lambda}_K\}$ denote the parameters of the K Gaussian components whose mixing coefficients are summarized by $\boldsymbol{\pi} = \{\pi_1, ..., \pi_K\}$. The mixture is constructed with uninformative conjugate priors such that its joint distribution is described by Equation (2.7) with

$$p\left(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) p\left(\mathbf{Z} \mid \boldsymbol{\pi}\right) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right)^{z_{nk}},$$

where z_{nk} denotes the k^{th} element of $\mathbf{z}_n$. The true posterior of this model $p(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ is assumed to be intractable.

While Mean-Field inference deals with the intractable posterior by introducing a factorized variational approximation where $\mathbf{Z}$ is assumed independent of the parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$, we are considering a parametric variational approximation. The posterior distribution of the Bayesian Gaussian mixture model covers both discrete and continuous variables. Constructing such a variational distribution over both types as a single distribution is difficult. Instead we will consider a factorization of the variational posterior as

$$q(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\mathbf{Z} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})\, q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}),$$

where $q(\mathbf{Z} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ further factorizes into $\prod_{n=1}^{N} q(\mathbf{z}_n \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$. We have implemented different approaches to the problem, which will be presented in succession, each with results and a short discussion on their performance.

5.1. Amortized Variational Inference For Bayesian GMMs

As the main interest lies in the posterior over the parameters of the mixture, namely $\boldsymbol{\pi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$, we will consider the case of *amortized* variational inference. Amortization is easily described with the case of N i.i.d. observations $\mathbf{X}$ for which we want to learn the posterior of the corresponding labels $\mathbf{Z}$ in a Gaussian mixture model with

$$p(\mathbf{X}) = \prod_{n=1}^{N} p(\mathbf{x}_n \mid \mathbf{z_n}, \boldsymbol{\theta})\, p(\mathbf{z}_n).$$

We would have to optimize a variational posterior $q_n(\mathbf{z}_n \mid \mathbf{x}_n)$ for every observation in $\mathbf{X}$. It is easy to see that the number of parameters to optimize increases with N. Amortized inference assumes some function $\mathbf{z}_n = f(\mathbf{x}_n)$ mapping the observations to labels such that the number of parameters to optimize is fixed to the parameters of $f(\cdot)$ allowing for much larger sets of observations [66]. In general, we can define amortization as replacing the optimization of a set of free variables with the optimization of a function such that the free variables are a result of the function, which fixes the number of parameters that are optimized to a certain level independent of the number of free variables. This constraint, however, comes with associated potential approximation errors. When optimizing free variables directly, these variables can take on any value in the space they are defined in. Amortization binds the variables to the limits for which the function is defined, causing

some variables never to be fully optimized. The resulting gap is known as the *amortization gap* [67–69].

Following this idea we will consider $\prod_{n=1}^{N} q\left(\mathbf{z}_n \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right)$ to be represented by a neural network mapping $\mathbf{z}_n = \mathrm{NN}\left(\mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right)$ for which the lower bound is defined as

$$\begin{aligned}\mathcal{F}\left(q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \boldsymbol{\omega}\right)\right) = &\sum_{\mathbf{Z}} \int\int\int q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \boldsymbol{\omega}\right) \log p\left(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) \mathrm{d}\boldsymbol{\pi}\mathrm{d}\boldsymbol{\mu}\mathrm{d}\boldsymbol{\Lambda} \\ &- \int\int\int q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \boldsymbol{\omega}\right) \log q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \boldsymbol{\omega}\right) \mathrm{d}\boldsymbol{\pi}\mathrm{d}\boldsymbol{\mu}\mathrm{d}\boldsymbol{\Lambda},\end{aligned}$$

where $\mathbf{Z} = \left\{\mathrm{NN}\left(\mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right)\right\}_{n=1}^{N}$. The variational posterior is defined as a Normalizing Flow with

$$\begin{aligned}\log q_L\left(\boldsymbol{\pi}^{(L)}, \boldsymbol{\mu}^{(L)}, \boldsymbol{\Lambda}^{(L)} \,\middle|\, \boldsymbol{\omega}\right) = &\log q_0\left(\boldsymbol{\pi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\Lambda}^{(0)}\right) \\ &- \sum_{l=1}^{L} \log \left|\, \det \mathbf{J}_{g_l}\left(\boldsymbol{\pi}^{(l-1)}, \boldsymbol{\mu}^{(l-1)}, \boldsymbol{\Lambda}^{(l-1)}\right) \right|,\end{aligned}$$

where $\left\{\boldsymbol{\pi}^{(L)}, \boldsymbol{\mu}^{(L)}, \boldsymbol{\Lambda}^{(L)}\right\}$ represents a sample of the complex transformed distribution. We have switched indexing here to not confuse an intermediate result of the transformation $\boldsymbol{\mu}^{(l-1)}$ with a sample of the parameters of the k^{th} component $\boldsymbol{\mu}_k$. The parameters $\boldsymbol{\omega}$ we will optimize are defined through the transformations $g = g_L \circ ... \circ g_2 \circ g_1$. It follows the negative lower bound

$$\begin{aligned}&-\mathcal{F}\ \left(q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \boldsymbol{\omega}\right)\right) \\ &\quad= \mathbb{E}_{q_0\left(\boldsymbol{\pi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\Lambda}^{(0)}\right)}\left[\log q_L\left(\boldsymbol{\pi}^{(L)}, \boldsymbol{\mu}^{(L)}, \boldsymbol{\Lambda}^{(L)}\right) - \log p\left(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}^{(L)}, \boldsymbol{\mu}^{(L)}, \boldsymbol{\Lambda}^{(L)}\right)\right] \\ &\quad= \mathbb{E}_{q_0\left(\boldsymbol{\pi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\Lambda}^{(0)}\right)}\left[\log q_0\left(\boldsymbol{\pi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\Lambda}^{(0)}\right) - \sum_{l=1}^{L} \log \left|\, \det \mathbf{J}_{g_l}\left(\boldsymbol{\pi}^{(l-1)}, \boldsymbol{\mu}^{(l-1)}, \boldsymbol{\Lambda}^{(l-1)}\right) \right|\right] \\ &\qquad - \mathbb{E}_{q_0\left(\boldsymbol{\pi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\Lambda}^{(0)}\right)}\left[\log p\left(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}^{(L)}, \boldsymbol{\mu}^{(L)}, \boldsymbol{\Lambda}^{(L)}\right)\right],\end{aligned}$$

for which we compute the gradients

$$\begin{aligned}&\nabla_{\boldsymbol{\omega}} - \mathcal{F}\left(q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \boldsymbol{\omega}\right)\right), \\ &\nabla_{\boldsymbol{\psi}} - \mathcal{F}\left(q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \boldsymbol{\omega}\right)\right),\end{aligned}$$

where ψ denotes the parameters of the neural network $\mathrm{NN}(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$. We again minimize the negative lower bound with SGD. This definition holds for any transformation efficient in the generative direction and is viable as long as sufficient representational power is provided.

As $\boldsymbol{\Lambda}$ denotes the set of precision matrices $\{\boldsymbol{\Lambda}_k\}_{k=1}^K$ with $\boldsymbol{\Lambda}_k \in \mathcal{R}^{D\times D}$, we will, in practice, redefine a complex sample from the Normalizing Flow as $\{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}\}$ with $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_k\}_{k=1}^K$. Here $\boldsymbol{\lambda}_k \in \mathcal{R}^{D_\lambda}$ denotes a vector of the elements in the lower triangular matrix L_k of the *Cholesky decomposition* of $\boldsymbol{\Lambda}_k$ with $\boldsymbol{\Lambda}_k = L_k L_k^T$ with

$$D_\lambda = D + \frac{D(D-1)}{2},$$

significantly reducing the number of dimension in the variational posterior. Further, to remain tractable when computing the gradients we will assume a continuous relaxation of the discrete labels $\mathbf{z}_n$. Instead of directly mapping onto the discrete labels we are using a temperature *relaxed one-hot Categorical* distribution parameterized by the functional mapping $\boldsymbol{\rho}_n = \mathrm{NN}(\mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ used to amortize the discrete labels. The relaxed one-hot categorical distribution relies on a *Gumbel Softmax* relaxation defined as

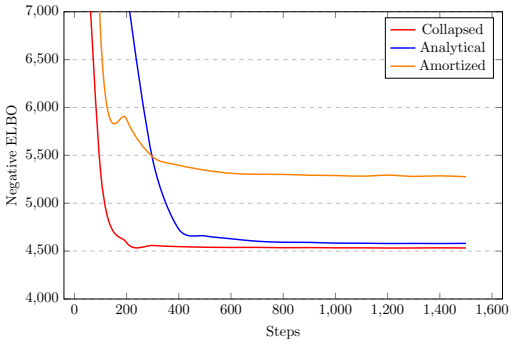
$$z_{nk} = \frac{\exp\left(\left(\log\left(\rho_{nk}\right) + g_i\right)/\tau\right)}{\sum_{j=1}^K \exp\left(\left(\log\left(\rho_{nj}\right) + g_j\right)/\tau\right)},$$

where τ denotes the temperature and g_k are i.i.d. samples drawn from a Gumbel distribution $\mathrm{Gumbel}(0, 1)$ [70, 71]. It follows the probability density of the relaxed categorical distribution

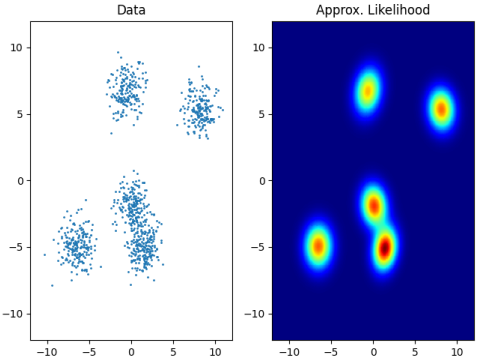
$$p\left(\mathbf{z}_n \mid \boldsymbol{\rho}_n\right) = \Gamma\left(K\right)\tau^{K-1}\left(\sum_{k=1}^K \frac{\rho_{nk}}{z_{nk}^\tau}\right)^{-1}\prod_{k=1}^K \frac{\rho_{nk}}{z_{nk}^{\tau+1}},$$

also known as *Gumbel-Softmax* [70] or *Concrete* [71] distribution, which were discovered independently [70]. For lower temperatures the distribution gets closer to the discrete categorical distribution while the variance of the gradient increases until vanishing completely [70]. Typically one follows an annealing schedule for the temperature to prevent high variance in the gradients to interfere with the optimization [70].

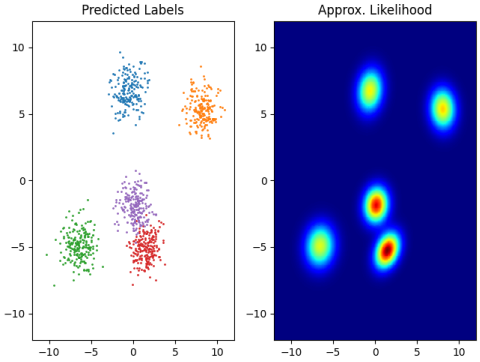
We have first applied this approach to evaluate its initial performance to basic datasets with $D = 2$ dimensions. In Figure 5.1 and Figure A.2 the results are shown for $N = 1000$ data points sampled from randomly generated Gaussian mixture distributions, each comprised of five equally weighted components. We have assumed the data is distributed according to a Bayesian Gaussian mixture model with $K = 5$ components. The hyperparameters



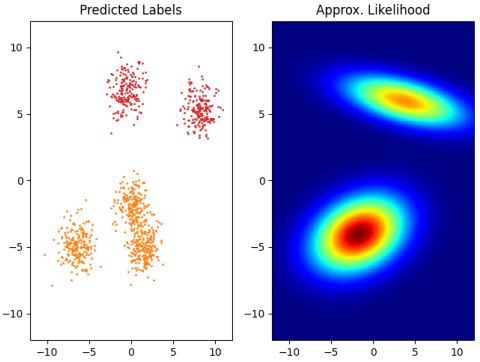
(a) Negative ELBO



(b) Collapsed VI



(c) Analytical $\mathbf{Z}$



(d) Amortized VI

Figure 5.1.: Shown are the results of the three presented approaches on data sampled from randomly generated Gaussian mixture distributions with five equally weighted components. The dataset contains $N = 1000$ samples. Figure (a) presents the minimization of the negative lower bound as an average of 100 posterior samples. In (b), the dataset is displayed in a single coloration as the assignment variables are collapsed. The other figures show the data colorized according to the assignment variables. The approximate likelihood corresponds to $p(\mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ parameterized by an average of 100 samples from the variational posterior $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$.

$p(\boldsymbol{\pi}) = \mathrm{Dir}(\alpha)$	$\alpha = \{1/K\}_{k=1}^{K}$
$p(\boldsymbol{\mu} \mid \boldsymbol{\Lambda}) = \mathcal{N}\left(\boldsymbol{\mu}_0 \,\middle|\, (\beta\boldsymbol{\Lambda}_k)^{-1}\right)$	$\boldsymbol{\mu}_0 = \mathbf{0}$, $\beta = 0.1 \; \forall \, k = 1, ..., K$
$p(\boldsymbol{\Lambda}) = \mathcal{W}(\nu, \mathbf{W})$	$\nu = D + 1$, $\mathbf{W} = \mathbf{I} \; \forall \, k = 1, ..., K$

Table 5.1.: Hyperparameters for the conjugate priors of every Bayesian Gaussian mixture model assumed for the experiments discussed in this chapter.

for the corresponding conjugate priors are listed in Table 5.1. The variational posterior $q(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ is factorized, where $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ is represented by a Block Neural Auto-Regressive Flow [27] with $L = 2$ transformations. The BNAFs are implemented in the generative direction for fast sampling in all experiments using the *Pyro.ai* library [72] and are optimized by an *Adam* [73] optimizer. The other factor $\{q(\mathbf{z}_n \mid \mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})\}_{n=1}^{N}$ is designed as a relaxed Categorical distribution whose parameters are computed by a dense neural network $\boldsymbol{\rho}_n = \mathrm{NN}(\mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ with two hidden-layers where each layer consists of $K \cdot D \cdot 50$ neurons. The temperature relaxation of the discrete assignments follows an annealing schedule according to a linear function

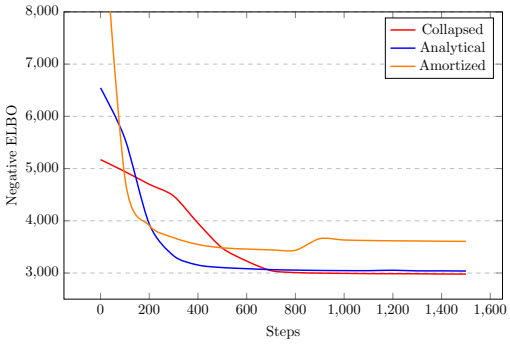
$$\tau = \max\left[0.01\,, 1 - \frac{t}{0.75T}\right], \tag{5.1}$$

where t is the current episode and T is the maximum number of episodes for training. We have also considered a smoothing of the negative lower bound of the form

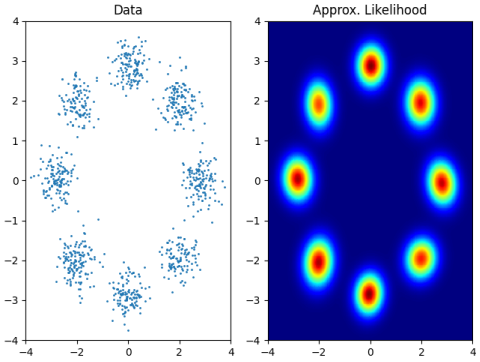
$$\begin{aligned} &-\mathcal{F}\;(q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \boldsymbol{\omega})) \\ &\quad = \mathbb{E}_{q_0\left(\boldsymbol{\pi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\Lambda}^{(0)}\right)}\left[\log q_L\left(\boldsymbol{\pi}^{(L)}, \boldsymbol{\mu}^{(L)}, \boldsymbol{\Lambda}^{(L)}\right) - \eta \log p\left(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}^{(L)}, \boldsymbol{\mu}^{(L)}, \boldsymbol{\Lambda}^{(L)}\right)\right], \end{aligned} \tag{5.2}$$

where $\eta = \min\left[1\,, 0.01 + \frac{t}{1000}\right]$ [18].

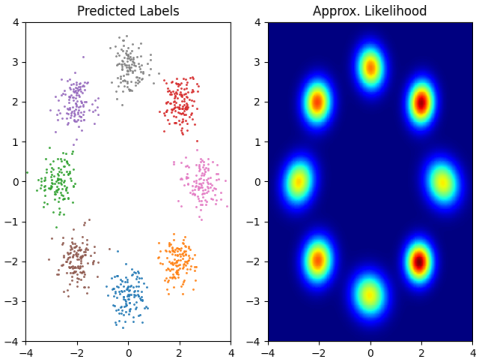
Further, a more structured example is given in Figure 5.2 for the *Eight Gaussians* data. In our example, the data contains $N = 1000$ observations. The Bayesian Gaussian mixture is now comprised of $K = 8$ components while keeping the priors' initialization fixed. We have reduced the variational posterior to $L = 1$ Block Neural Auto-Regressive transformation, which gave the best results. Smoothing and temperature relaxation were applied in the same way as for the random Gaussian mixture data. Further, the amortization was kept unchanged. In all three experiments, Figure 5.1, Figure A.2 and Figure 5.2, we use a



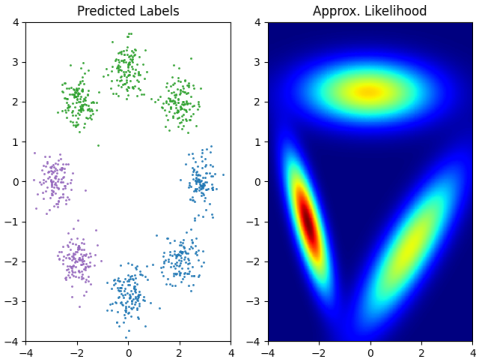
(a) Negative ELBO



(b) Collapsed VI



(c) Analytical $\mathbf{Z}$



(d) Amortized VI

Figure 5.2.: Shown are the results of the three presented approaches on the Eight Gaussians data. The dataset contains $N = 1000$ samples. Figure (a) presents the minimization of the negative lower bound as an average of 100 posterior samples. In (b), the dataset is displayed in a single coloration as the assignment variables are collapsed. The other figures show the data colorized according to the assignment variables. The approximate likelihood corresponds to $p(\mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ parameterized by an average of 100 samples from the variational posterior $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$.

batch of 100 samples from the variational posterior to compute the Monte-Carlo estimate of the lower bound during training.

The results in Figure 5.1, Figure A.2 and Figure 5.2 show an underwhelming performance of amortized variational inference as it was presented for even the most simplistic datasets. The main reason is that the model cannot fully separate clusters close to each other. The resulting variational posterior assigns small mixing coefficients to most components in the Bayesian Gaussian mixture, causing a mode-averaging behavior of the likelihood. In some cases, the posterior collapses entirely, causing all observations to be assigned to a single component. We have verified that each factor $q\left(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right)$ and $q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right)$ of the variational posterior by itself is capable of correctly representing its corresponding variables by fixing the other factor to the optimal solution. However, trained in conjunction, the data is not correctly separated. The problem persists independent of the training procedure. We have tried a separate EM-like update procedure where either factor was kept fixed while the other is being updated, including different hyperparameters for the neural network and different optimizers. Furthermore, the described approach becomes unbearably slow for more complex models. The dimensionality of the variational factor $q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right)$ not only increases with the growing dimensionality of the observations but also with the number of components in the Bayesian Gaussian mixture. Thus, due to the dependency placed on the mixture parameters $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}$ in the neural network used for amortization, the necessary parameters we have to optimize grows even further. Due to its poor performance, we have not applied the approach to more complex data. Instead, further approaches presented in this chapter are compared to Mean-Field inference.

5.2. Analytically Deriving Assignment Variables

The results for amortized variational inference, as discussed before, are unsatisfying. In an attempt to overcome the issues with the amortization, we have considered a different approach to computing the assignment variables. Let $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \sim q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \boldsymbol{\omega}\right)$ be a sample from the variational posterior. Given this sample we have conditional independency between the n^{th} assignment variable $\mathbf{z}_n$ and all other assignment variables $\mathbf{z}_{\neg n}$ as well as conditional independence between the n^{th} sample $\mathbf{x}_n$ and all other samples $\mathbf{x}_{\neg n}$. These independencies allow for a computation of $\mathbf{z}_n$ as a distribution

$$p\left(\mathbf{z}_n \mid \mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) = \frac{p\left(\mathbf{z}_n, \mathbf{x}_n \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right)}{p\left(\mathbf{x}_n \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right)}$$

using Bayes Rule. From the definition of the Bayesian Gaussian mixture model, we know that the numerator is given as

$$p\left(\mathbf{z}_n, \mathbf{x}_n \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) = \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right)^{z_{nk}},$$

while the denominator is obtained by marginalizing out the assignment variable $\mathbf{z}_n$

$$\begin{aligned} p\left(\mathbf{x}_n \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) &= \sum_{\mathbf{z}_n} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right)^{z_{nk}} \\ &= \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right), \end{aligned}$$

where we use the fact that $\mathbf{z}_n$ is a one-hot vector to regain the non-Bayesian Gaussian mixture density. As the denominator is independent of $\mathbf{z}_n$ we can consider it constant to arrive at the relation

$$\begin{aligned} p\left(\mathbf{z}_n \mid \mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) &\propto \prod_{k=1}^{K} \left(\pi_k \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right)\right)^{z_{nk}} \\ &= \prod_{k=1}^{K} \rho_k^{z_{nk}}, \end{aligned}$$

which is a Categorical distribution whose unnormalized weights are denoted by ρ [2]. With the normalized weights being under the constraints that

$$0 \leq r_k \leq 1, \qquad \sum_{k=1}^{K} r_k = 1,$$

and knowing that $\mathbf{z}_n$ is a one-hot vector, we can define

$$p\left(\mathbf{z}_n \mid \mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) = \prod_{k=1}^{K} r_k^{z_{nk}},$$

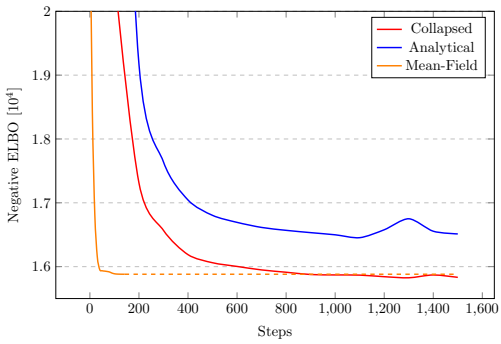
where the weights are normalized by the constant $p\left(\mathbf{x}_n \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right)$ such that

$$r_k = \frac{\pi_k \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right)}{\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right)}.$$

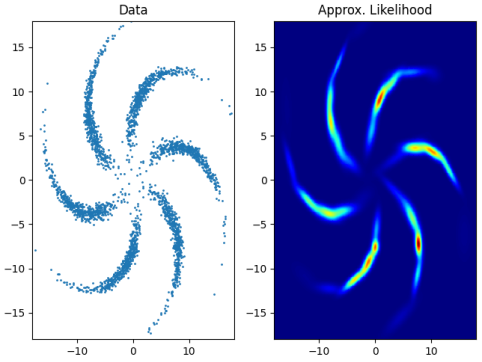
We use this relation to replace the amortization of the discrete assignments we have discussed before. Instead of parameterizing the relaxed one-hot Categorical distribution by some weights predicted through a neural network mapping, we use the unnormalized weights ρ to sample continuous relaxations of the discrete assignments. This approach allows for the assignments to be computed freely without being bound to a functional mapping while slowly converging to the discrete analytical solution using an annealing schedule for the temperature as described in Equation (5.1). In Figure 5.1 and Figure 5.2 we already see a significant improvement over the amortized variational inference approach. We refer to this approach as *Analytical* for clarity. The experiments are performed under the same constraints as the amortized counterpart with a Bayesian Gaussian mixture model of $K = 5$ and $K = 8$ respective components. The conjugate priors are again initialized according to Table 5.1. The same parametric model comprised of $L = 2$ for the random Gaussian mixture data and $L = 1$ Block Neural Auto-Regressive Flows for the Eight Gaussians data is used as the variational posterior with smoothing of the loss function as described for Equation (5.2). As for the amortized approach, we have used 100 samples to compute the Monte-Carlo estimate of the lower bound during training.

The approach scales better to more complex data such as the *Pinwheel* dataset (see Figure 5.3). In our experiments, the Pinwheel dataset contains $N = 3000$ observations distributed along six half-moon shapes. The Bayesian Gaussian mixture model is comprised of $K = 40$ components, where each components' prior is initialized by the parameters listed in Table 5.1. The variational posterior $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ is designed as a Normalizing Flow with $L = 3$ Block Neural Auto-Regressive transformations. We have further applied linear smoothing of the lower bound following the design in Equation (5.2). While the results show a reasonable recreation of the data distribution, we see in Figure 5.3 (a) that Mean-Field inference still provides a better approximation. Further applying the approach to the *Two Spirals* data, which is also comprised of $N = 3000$ observations shown in Figure 5.4 strengthens this statement. In this experiment, we used the exact same setting as described for the Pinwheel experiment. A summary of the final values for the negative lower bound is listed in Table 5.2. The only experiment listed where this approach outperforms Mean-Field inference is on the Eight Gaussians data.

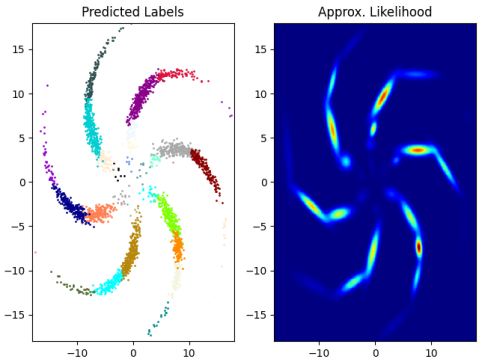
An interesting result we have encountered in some experiments is bi-modal or even multi-modal behavior of the posterior distribution. By defining the variational posterior as a Normalizing Flow, the posterior is theoretically capable of representing multi-modal data. Figure A.1 shows a variant of the Two Spirals data with a Bayesian Gaussian mixture comprised of $K = 60$ components where we have documented such behavior. While the other results are shown by computing an average of 100 posterior samples, here we use only a single sample to compute the assignments and approximate likelihood. We do so



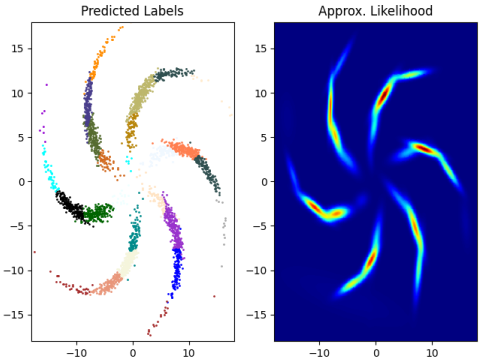
(a) Negative ELBO



(b) Collapsed VI



(c) Analytical $\mathbf{Z}$



(d) Mean-Field

Figure 5.3.: Shown is a comparison of two of the presented approaches and Mean-Field inference on the Pinwheel data. The dataset contains $N = 3000$ samples. Figure (a) presents the minimization of the negative lower bound as an average of 100 posterior samples. In (b), the dataset is displayed in a single coloration as the assignment variables are collapsed. The other figures show the data colorized according to the assignment variables. The approximate likelihood corresponds to $p(\mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ parameterized by an average of 100 samples from the variational posterior $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$.

Approach	Random GMM	Eight Gaussians	Pinwheel	Two Spirals
Amortized	5277	3606		
Analytical	4581	3038	16433	13357
Mean-Field		3084	15889	13139
Collapsed	4533	2980	15832	12939

Table 5.2.: Listed are the final negative lower bound values for the presented approaches as an average over 100 posterior samples. The table includes results for Mean-Field inference as presented in Chapter 3 where the variational posterior is initialized by a single step Gibbs Sampling.

as we have not found a reasonable way of separating the samples from the posterior to filter out samples from a single mode.

5.3. Collapsed Variational Inference For Bayesian GMMs

We have discussed before that our main interest is a posterior over the parameter of the mixture $\boldsymbol{\pi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$. So far, we have considered approaches to obtain some reasonable assumption about the assignment variables $\mathbf{Z}$. Another option is to assume a collapsed approach, where some of the latent variables are marginalized out beforehand [13,74]. In most cases, this technique is applied to marginalize out the parameters of the model such that the posterior is a distribution over the latent assignments $\mathbf{Z}$ [75–77]. However, we will apply the concept to marginalize out the assignment variables $\mathbf{Z}$ which corresponds to exact inference w.r.t. to those variables and removing them from the derivation of the lower bound. Considering such a collapsed model not only improves the lower bound but also speeds up learning as fewer variables need to be inferred [13]. We take the problem definition from the beginning of this chapter with a Bayesian Gaussian mixture model whose full joint distribution is described as

$$p\left(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) = C_\alpha \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \mathcal{NW}\left(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k \mid \boldsymbol{\zeta}_k\right) \prod_{n=1}^{N} \pi_k^{z_{nk}} \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right)^{z_{nk}},$$

where we have assumed conjugate priors. From this joint distribution we marginalize out the assignment variables $\mathbf{Z}$ such that

$$\begin{aligned}
p\left(\mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) &= \sum_{\mathbf{Z}} C_\alpha \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \mathcal{NW}\left(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k \,|\, \boldsymbol{\zeta}_k\right) \prod_{n=1}^{N} \pi_k^{z_{nk}} \mathcal{N}\left(\mathbf{x}_n \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right)^{z_{nk}} \\
&= C_\alpha \left[\prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \mathcal{NW}\left(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k \,|\, \boldsymbol{\zeta}_k\right)\right] \sum_{\mathbf{Z}} \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}\left(\mathbf{x}_n \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right)^{z_{nk}} \\
&= C_\alpha \left[\prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \mathcal{NW}\left(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k \,|\, \boldsymbol{\zeta}_k\right)\right] \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_n \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right),
\end{aligned}$$

where $\boldsymbol{\zeta}_k$ represents the parameters $\boldsymbol{\mu}_{0,k}, \beta_k, \nu_k, \mathbf{W}_k$ of the k^{th} Gaussian-Wishart prior. This marginalizing of the assignment variables is possible due to all $\mathbf{z}_n$ being one-hot vectors. Being given every possible state of $\mathbf{z}_n$ corresponds to the term $\pi_k \mathcal{N}\left(\mathbf{x}_n \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\right)$ being evaluated exactly once for every component k in the mixture distribution. Thus the sum over $\mathbf{Z}$ and the product over components k merges into a sum over components [3]. It follows the log marginal distribution for the observation $p\left(\mathbf{X}\right)$ as

$$\log p\left(\mathbf{X}\right) = \log \int\int\int p\left(\mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) \mathrm{d}\boldsymbol{\pi} \mathrm{d}\boldsymbol{\mu} \mathrm{d}\boldsymbol{\Lambda},$$

where we introduce the variational posterior $q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \,|\, \boldsymbol{\omega}\right)$ to derive the lower bound

$$\begin{aligned}
\mathcal{F}\left(q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \,|\, \boldsymbol{\omega}\right)\right) = &\int\int\int q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \,|\, \boldsymbol{\omega}\right) \log p\left(\mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) \mathrm{d}\boldsymbol{\pi} \mathrm{d}\boldsymbol{\mu} \mathrm{d}\boldsymbol{\Lambda} \\
&- \int\int\int q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \,|\, \boldsymbol{\omega}\right) \log q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \,|\, \boldsymbol{\omega}\right) \mathrm{d}\boldsymbol{\pi} \mathrm{d}\boldsymbol{\mu} \mathrm{d}\boldsymbol{\Lambda},
\end{aligned}$$

using Jensen's inequality. By marginalizing over the assignment variables $\mathbf{Z}$ beforehand, there no longer exists a closed-form solution for the variational posterior. However, by assuming a parametric variational distribution $q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \,|\, \boldsymbol{\omega}\right)$ we can still optimize the lower bound w.r.t. the parameters $\boldsymbol{\omega}$. Following similar steps as before, we arrive at the negative lower bound

$$\begin{aligned}
-\mathcal{F}\left(q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \,|\, \boldsymbol{\omega}\right)\right) = &\mathbb{E}_{q_0\left(\boldsymbol{\pi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\Lambda}^{(0)}\right)} \left[\log q_0\left(\boldsymbol{\pi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\Lambda}^{(0)}\right)\right] \\
&- \mathbb{E}_{q_0\left(\boldsymbol{\pi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\Lambda}^{(0)}\right)} \left[\sum_{l=1}^{L} \log \left| \det \mathbf{J}_{g_l}\left(\boldsymbol{\pi}^{(l-1)}, \boldsymbol{\mu}^{(l-1)}, \boldsymbol{\Lambda}^{(l-1)}\right) \right|\right] \\
&- \mathbb{E}_{q_0\left(\boldsymbol{\pi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\Lambda}^{(0)}\right)} \left[\log p\left(\mathbf{X}, \boldsymbol{\pi}^{(L)}, \boldsymbol{\mu}^{(L)}, \boldsymbol{\Lambda}^{(L)}\right)\right],
\end{aligned}$$

where the variational posterior is defined by a Normalizing Flow in terms of L bijective and differentiable transformations g and a base distribution q_0.

We have applied the collapsed variational inference approach to all experiments we have explained so far using the same designs for the variational posterior $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \boldsymbol{\omega})$ and the Bayesian Gaussian mixture. Across all experiments, the collapsed model performs at least as well as the previously presented approaches. While in Figure 5.1, Figure A.2 and Figure 5.2 we have used a Monte-Carlo estimate over 100 posterior samples per episode for the previously presented approaches, the collapsed model achieves at least as good of a performance on a single sample Monte-Carlo estimate. Further, the experiments on the Pinwheel (see Figure 5.3) and Two Spirals (see Figure 5.4) data show that the collapsed model also outperforms Mean-Field inference. The final estimated values of the lower bound are listed for all experiments in Table 5.2 as a Monte-Carlo estimate over 100 posterior samples for all approaches. We have experienced a more stable optimization during the experiments allowing for higher learning rates and thus faster convergence. Furthermore, the collapsed model shows a significant boost in per-episode computation time.

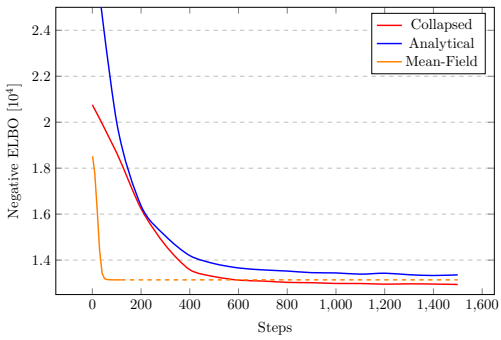
We have thoroughly discussed variational inference in the context of learning variational posterior distributions as approximations to the true posterior of an assumed model over given data. The following section of this chapter, however, is devoted to the concept of *Density Estimation* or *Density Matching* using variational inference. We do so to show a proof of concept for a separate update procedure for mixtures of Normalizing Flows based on ideas from [36, 37]. The goal is to learn highly multi-modal distributions rather as a mixture of less complex distributions than having a single highly complex distribution.

5.4. Variational Inference With Mixtures Of Normalizing Flows

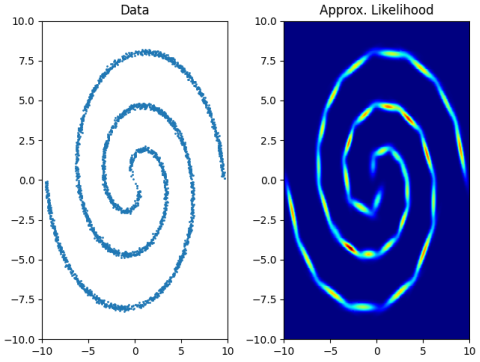
Distributions of real world data are almost certainly hideously complex to the point where inferring any valuable information becomes intractable. Such distributions $p^*(\mathbf{x})$ follow a general form such as

$$p^*(\mathbf{x}) = \frac{1}{C_{\boldsymbol{\theta}}} \prod_k \phi_k(G_k(\mathbf{x}); \boldsymbol{\theta}),$$

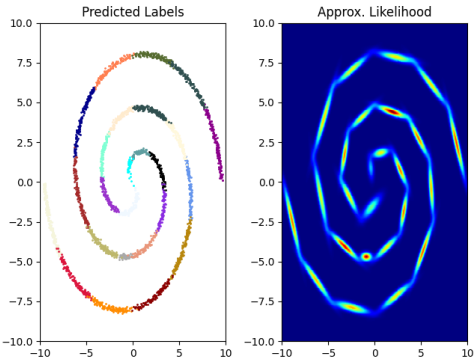
where $C_{\boldsymbol{\theta}}$ is a normalization constant representing a marginalization over all possible values of the random variables $\mathbf{x}$. If we consider undirected graphical models, each factor $\phi_k(G_k(\mathbf{x}); \boldsymbol{\theta})$ corresponds to a potential function over a clique or in other words fully



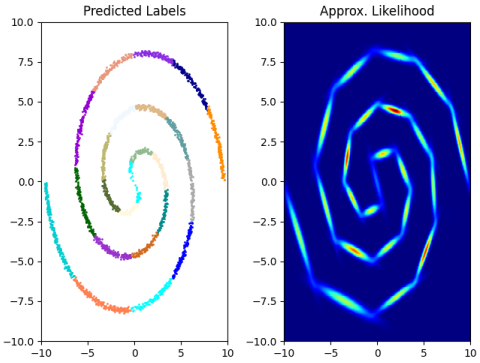
(a) Negative ELBO



(b) Collapsed VI



(c) Analytical $\mathbf{Z}$



(d) Mean-Field

Figure 5.4.: Shown is a comparison of two of the presented approaches and Mean-Field inference on the Two Spirals data. The dataset contains $N = 3000$ samples. Figure (a) presents the minimization of the negative lower bound as an average of 100 posterior samples. In (b), the dataset is displayed in a single coloration as the assignment variables are collapsed. The other figures show the data colorized according to the assignment variables. The approximate likelihood corresponds to $p(\mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ parameterized by an average of 100 samples from the variational posterior $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$.

connected subgraph $G_k(\mathbf{x})$ [5]. We assume that the normalization constant is intractable and thus sampling is impossible. For such models however, it is, in most cases, still possible to compute all potential functions $\phi_k(G_k(\mathbf{x});\boldsymbol{\theta})$ for a given sample of $\mathbf{x}$ such that the unnormalized density $p(\mathbf{x}) = \prod_k \phi_k(G_k(\mathbf{x});\boldsymbol{\theta})$ can be evaluated. It is, therefore, possible to learn a tractable generative model to represent the intractable true data distribution by density matching using variational inference.

Let $q(\mathbf{x} \mid \boldsymbol{\omega})$ be a tractable, parametric generative model, e.g. a Normalizing Flow, for which we want to minimize some distance measure to the true data distribution $p^*(\mathbf{x})$ such as the KL

$$\min_{q(\mathbf{x}\mid\boldsymbol{\omega})} \mathrm{KL}\left(q(\mathbf{x}\mid\boldsymbol{\omega}) \,\|\, p^*(\mathbf{x})\right),$$

where we consider an information projection to get an expectation w.r.t. the tractable model rather than the intractable true distribution. Minimizing this KL directly is impossible for either case as $p^*(\mathbf{x})$ is assumed to be intractable. Instead we reformulate the problem to

$$\mathrm{KL}\left(q(\mathbf{x}\mid\boldsymbol{\omega}) \,\|\, p^*(\mathbf{x})\right) = \mathrm{KL}\left(q(\mathbf{x}\mid\boldsymbol{\omega}) \,\|\, p(\mathbf{x})\right) + \log C_{\boldsymbol{\theta}},$$

by isolating the intractable normalization constant [36, 37, 42]. By recognizing that the $\log$ of $C_{\boldsymbol{\theta}}$ is again a constant we notice that $\mathrm{KL}\left(q(\mathbf{x}\mid\boldsymbol{\omega}) \,\|\, p(\mathbf{x})\right) \geq 0$ is a negative lower bound $-\mathcal{F}\left(q(\mathbf{x}\mid\boldsymbol{\omega})\right)$ on the original objective with

$$\mathcal{F}\left(q(\mathbf{x}\mid\boldsymbol{\omega})\right) = \int q(\mathbf{x}\mid\boldsymbol{\omega}) \log \frac{p(\mathbf{x})}{q(\mathbf{x}\mid\boldsymbol{\omega})} \mathrm{d}\mathbf{x}.$$

As discussed before, we minimize the true objective by minimizing the negative lower bound through a Monte-Carlo estimate

$$-\mathcal{F}\left(q(\mathbf{x}\mid\boldsymbol{\omega})\right) = \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x}\mid\boldsymbol{\omega})}\left[\log q(\mathbf{x}\mid\boldsymbol{\omega}) - \log p(\mathbf{x})\right],$$

where samples are drawn from the tractable approximation. For arbitrary tractable models a minimization is performed by SGD via the gradient $\nabla_{\boldsymbol{\omega}} - \mathcal{F}\left(q(\mathbf{x}\mid\boldsymbol{\omega})\right)$ of the negative lower bound w.r.t. the parameters of the tractable model.

It was shown that for Gaussian mixture models of the form $q(\mathbf{x}\mid\boldsymbol{\omega}) = \sum_{k=1}^{K} \pi_k q(\mathbf{x}\mid\boldsymbol{\omega}_k)$, where $q(\mathbf{x}\mid\boldsymbol{\omega}_k) = \mathcal{N}(\mathbf{x}\mid\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$ with $\boldsymbol{\omega}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\}$, that a closed-form solution exists for each component updated separately using a policy search method called *Model-Based Relative Entropy* [78] stochastic search [36,37]. While MORE introduces a trust-region and entropy constraint to stabilize the optimization, these constraints are not strictly necessary in the formulation of separate updates for the different components of the mixture. In fact

we can derive a separate update procedure for mixture models of arbitrary distributions by following the underlying concept presented in [36, 37].

Let $q(\mathbf{x} \mid \boldsymbol{\omega}) = \sum_{k=1}^{K} \pi_k q(\mathbf{x} \mid \boldsymbol{\omega}_k) = \sum_{k=1}^{K} q(\mathbf{x} \mid k)\, q(k)$ be a mixture model for an arbitrary parametric distribution where all components $q(\mathbf{x} \mid \boldsymbol{\omega}_k)$ have the same functional form, e.g. a Normalizing Flow. To keep the following derivations uncluttered we will, without loss of generality, hide the dependency on $\boldsymbol{\omega}$. Starting from the negative lower bound we derive

$$
\begin{aligned}
-\mathcal{F}(q(\mathbf{x})) &= \int q(\mathbf{x}) \left[\log q(\mathbf{x}) - \log p(\mathbf{x})\right] \mathrm{d}\mathbf{x} \\
&= \int \sum_{k=1}^{K} q(\mathbf{x} \mid k)\, q(k) \left[\log q(\mathbf{x}, k) - \log q(k \mid \mathbf{x}) - \log p(\mathbf{x})\right] \mathrm{d}\mathbf{x} \\
&= \sum_{k=1}^{K} q(k) \int q(\mathbf{x} \mid k) \left[\log q(\mathbf{x}, k) - \log q(k \mid \mathbf{x}) - \log p(\mathbf{x})\right] \mathrm{d}\mathbf{x},
\end{aligned}
$$

where $q(\mathbf{x}) = q(\mathbf{x}, k) \,/ q(k \mid \mathbf{x}) \;\forall\, k = 1, ..., K$ follows from Bayes Rule [36, 37]. Introducing an auxiliary distribution $\tilde{q}(k \mid \mathbf{x})$ for the posterior on the mixing coefficients leads to

$$
\begin{aligned}
-\mathcal{F}(q(\mathbf{x})) &= \sum_{k=1}^{K} q(k) \int q(\mathbf{x} \mid k) \left[\log q(\mathbf{x}, k) - \log \frac{q(k \mid \mathbf{x})\, \tilde{q}(k \mid \mathbf{x})}{\tilde{q}(k \mid \mathbf{x})} - \log p(\mathbf{x})\right] \mathrm{d}\mathbf{x} \\
&= \sum_{k=1}^{K} q(k) \int q(\mathbf{x} \mid k) \left[\log q(\mathbf{x}, k) - \log \tilde{q}(k \mid \mathbf{x}) - \log p(\mathbf{x})\right] \mathrm{d}\mathbf{x} \\
&\quad - \int \sum_{k=1}^{K} q(\mathbf{x} \mid k)\, q(k) \log \frac{q(k \mid \mathbf{x})}{\tilde{q}(k \mid \mathbf{x})} \mathrm{d}\mathbf{x} \\
&= \sum_{k=1}^{K} q(k) \int q(\mathbf{x} \mid k) \left[\log q(\mathbf{x}, k) - \log \tilde{q}(k \mid \mathbf{x}) - \log p(\mathbf{x})\right] \mathrm{d}\mathbf{x} \\
&\quad - \mathbb{E}_{q(\mathbf{x})} \left[\mathrm{KL}\left(q(k \mid \mathbf{x}) \,\|\, \tilde{q}(k \mid \mathbf{x})\right)\right],
\end{aligned}
$$

where the last step uses the *Chain Rule* of probability to obtain $q(\mathbf{x}) \sum_{k=1}^{K} q(k \mid \mathbf{x}) = \sum_{k=1}^{K} q(\mathbf{x} \mid k)\, q(k)$ through which we get an expectation of the KL between $q(k \mid \mathbf{x})$ and $\tilde{q}(k \mid \mathbf{x})$ w.r.t. $q(x)$ [36, 37]. By recognizing that

$$
\int \sum_{k=1}^{K} q(\mathbf{x} \mid k)\, q(k) \left[\log q(\mathbf{x}, k) - \log \tilde{q}(k \mid \mathbf{x}) - \log p(\mathbf{x})\right] \mathrm{d}\mathbf{x},
$$

is an upper bound on the negative lower bound $-\mathcal{F}(q(\mathbf{x}))$ we optimize the lower bound in an EM-like procedure, where the E-Step is simply defined as

$$\tilde{q}(k \mid \mathbf{x}) = \frac{q(\mathbf{x} \mid k)\, q(k)}{\sum_{k=1}^{K} q(\mathbf{x} \mid k)\, q(k)}.$$

As the upper bound no longer depends directly on $q(\mathbf{x})$ and decomposes into individual terms of the mixing coefficients $q(k)$ and the mixture components $q(\mathbf{x} \mid k)$, the M-Step is further divided into updates for each component, and the mixing coefficients [36, 37]. Using a Monte-Carlo estimate, the M-Step for the k^{th} component is defined as

$$\begin{aligned}
&\min_{q(\mathbf{x} \mid k)} \mathbb{E}_{\mathbf{x} \sim q(\mathbf{x} \mid k)} \left[\log q(\mathbf{x}, k) - \log \tilde{q}(k \mid \mathbf{x}) - \log p(\mathbf{x})\right] \\
= &\min_{q(\mathbf{x} \mid k)} \mathbb{E}_{\mathbf{x} \sim q(\mathbf{x} \mid k)} \left[\log q(\mathbf{x} \mid k) - \log \tilde{q}(k \mid \mathbf{x}) - \log p(\mathbf{x})\right] + H(q(k)) \\
\propto &\min_{q(\mathbf{x} \mid k)} \frac{1}{N} \sum_{n=1}^{N} \left[\log q(\mathbf{x}_n \mid k) - \log \tilde{q}(k \mid \mathbf{x}_n) - \log p(\mathbf{x}_n)\right],
\end{aligned}$$

where N is the number of samples $\mathbf{x}_n$ drawn from $q(\mathbf{x} \mid k)$ [36, 37]. The update is performed by SGD using the gradient $\nabla_{\boldsymbol{\omega}_k}$ of the Monte-Carlo estimates. After all components have been updated, the M-Step for the mixing coefficients is defined as

$$q(k) = \frac{\exp R(k)}{\sum_{k=1}^{K} \exp R(k)},$$

where $R(k)$ is the negative of the Monte-Carlo estimate

$$R(k) = -\frac{1}{N} \sum_{n=1}^{N} \left[\log q(\mathbf{x}_n \mid k) - \log \tilde{q}(k \mid \mathbf{x}_n) - \log p(\mathbf{x}_n)\right], \tag{5.3}$$

for which new samples from the updated components are drawn [36, 37]. Applying this decomposition to mixtures of Normalizing Flows allows for a separate update procedure of each individual Flow in the mixture.

Given sufficiently many sufficiently complex bijective transformations, a Flow can provably match any arbitrarily complex distribution. Though, in highly multi-modal distributions, it might become difficult to supply these transformations. Instead, we propose using mixture models through which a single component must not match the full multi-modal distribution but only a subset of modes. Matching only subsets of the modes is easier and

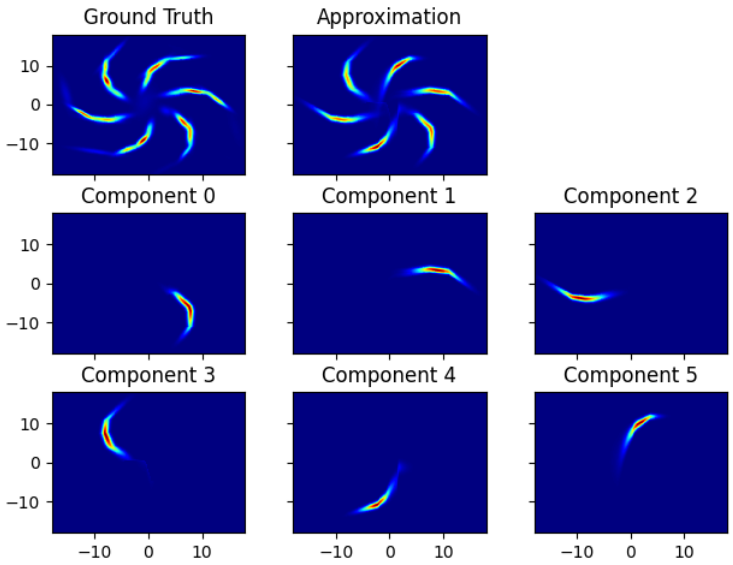


Figure 5.5.: The figure summarizes the results of a mixture of Normalizing Flows trained in the presented decomposable update procedure on a Pinwheel density. The mixture contains $K = 6$ components, where each component is comprised of $L = 2$ affine coupling transformations. The top left graphic shows the ground truth density $p(\mathbf{x})$, while right next to it, the full trained mixture density is plotted. The two lower rows show each component of the mixture separately to visualize how each component targeted a different mode.

requires less complex Flows such that basic transformations, e.g., affine coupling transformations, become viable for complex distributions without large numbers of successive transformations. As a proof of concept, we show the use of affine coupling transformations on a Pinwheel density in Figure 5.5. For the experiment, we provide a ground truth density matching the Pinwheel data. We initialize the variational distribution as a mixture of Normalizing Flows with $K = 6$ components where each component is comprised of a Gaussian base distribution and $L = 2$ affine coupling transformations. The base distribution is trainable and initialized with zero mean and identity precision matrix. Since we consider an example with only two dimensions, the input dimensions to the affine transformations are swapped after the first transformation in each component. The variational distribution is trained by an Adam optimizer for each component with a learning rate of 10^{-1} where

we use $N = 1000$ samples to compute the Monte-Carlo estimate of the lower bound. The results show that even with only $L = 2$ affine coupling transformations in each component, we achieve a reasonable representation of the ground truth density. This result is achieved by having each component only represent a single of the six modes in the ground truth density.

However, there is no guarantee with the presented approach that each component locks onto a different mode. It is entirely possible that some components converge onto the same mode causing one of the components' mixing coefficients to approach zero. Still, representing a distribution with mixture models whose components' updates are decomposable allows for parallel computation of the updates, significantly increasing available computational power. In addition, each component is less complex than a single distribution of equal representational power, further simplifying computations. Applying this decomposition to mixtures of Normalizing Flows brings an advantage compared to the Gaussian mixture models considered in [36, 37]. To fully represent all modes of a target density, the approach presented in [36, 37] relied on large initial mixtures with enough components to sufficiently represent the target density or on adding new components during the optimization. While rather simple coupling transformations such as affine couplings are inferior to auto-regressive couplings in representational power, they are still far superior to Gaussian distributions and capable of potentially modeling more than a single mode. As such a mixture of Normalizing Flows can compensate for missing components to a certain extend by representing multiple modes with a single component.

It is easy to see that the shown derivations also extend to variational posteriors defined as mixture models. Let $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ be a set of N i.i.d. observations of a random variable $\mathbf{x}$, where we will assume that this data is distributed according to a Latent Variable Model as described by Figure 3.1. For this setting, we have derived the negative lower bound as

$$-\mathcal{F}\left(q\left(\boldsymbol{\theta} \,|\, \boldsymbol{\omega}\right)\right) = \mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta} \,|\, \boldsymbol{\omega})}\left[\log q\left(\boldsymbol{\theta} \,|\, \boldsymbol{\omega}\right) - \log p\left(\mathbf{X}, \boldsymbol{\theta}\right)\right],$$

where $q\left(\boldsymbol{\theta} \,|\, \boldsymbol{\omega}\right)$ is the variational posterior and $p\left(\mathbf{X}, \boldsymbol{\theta}\right)$ is the joint probability of the LVM. By assuming that $q\left(\boldsymbol{\theta} \,|\, \boldsymbol{\omega}\right) = \sum_{k=1}^{K} \pi_k q\left(\boldsymbol{\theta} \,|\, \boldsymbol{\omega}_k\right)$ we arrive at a similar decomposition as presented for density matching. Thus, we can learn a multi-modal representation of the posterior over the parameters $\boldsymbol{\theta}$ as a mixture of Normalizing Flows with decomposable updates for the mixture components and mixing coefficients, given that each component is powerful enough to represent at least one of the posteriors' modes. Considering, however, that, in the parameter space of models such as the Bayesian Gaussian mixture, the modes represent a re-ordering of certain dimensions in the posterior distribution, there is no benefit to the lower bound itself.

6. Discussion And Related Works

We have presented results and ideas around variational inference for mixture models in Chapter 5. The contribution here was split into two parts, estimating the variational posterior over parameters of a Bayesian Gaussian mixture model and learning a density as an I-Projection for some unnormalized distribution with mixtures of Normalizing Flows. In this chapter, we will further discuss the results and approaches presented, including important related work.

The results presented in Chapter 5 show that the performance of variational inference as we have presented depends heavily on the quality of the prediction for the discrete assignment variables. Amortizing the assignment variables as a relaxed Categorical distribution parameterized by a dense neural network performed poorly on any given data we have tested. The approach was unable to separate clusters of data close to each other resulting in mode averaging behavior of the Bayesian Gaussian mixture model.

Deriving a closed-form solution for the assignment variables from the idea of Gibbs sampling where

$$p\left(\mathbf{z}_n \mid \mathbf{X}, \mathbf{z}_{\neg n}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right) = p\left(\mathbf{z}_n \mid \mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right),$$

with given samples $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} \sim q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right)$ provided significant improvement. Even though the derivation for the assignment variables included specific dependency on the parameters $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}$, the approach did still perform worse than Mean-Field inference for the Pinwheel and Two Spirals data. This result is surprising as we expected to see a better performance than Mean-Field, which assumes independence between the assignment variables and the parameters. Only on the Eight Gaussians data did we outperform Mean-Field (see Table 5.2).

To further improve the performance of variational inference with Normalizing Flows on Bayesian Gaussian mixture models, we collapsed the assignment variables. By collapsing a subset of the latent variables, we effectively perform exact inference over the collapsed latent variables. While there is no longer a closed-form solution for the parameters of the

mixture model, we can still perform variational inference given a parametric model for the variational posterior. The provided results show as a proof of concept that we achieve better performance than Mean-Field inference across all experiments. Not only did we improve the performance but also significantly increased the per-episode computation time allowing for higher dimensional problems and larger mixtures to be solved in a reasonable amount of time.

However, by defining the variational posterior over the Bayesian Gaussian mixture model's parameters, we have a rapidly increasing amount of dimensions to represent. Even with reduced dimensionality by only learning the lower triangular Cholesky decomposition of the precision matrix, we have

$$D_q = K(1 + 2D + \frac{D(D-1)}{2}),$$

dimensions in the posterior, where K is the number of components in the mixture, and D is the number of dimensions of the observed data $\mathbf{x}$. Due to the dependence of D_q on D and K, the approach does not scale well to very high dimensional data $\mathbf{x}$ without applying some form of dimensionality reduction beforehand. To achieve good scaling to high dimensional data, the literature around Normalizing Flows typically considers VAEs [39, 40, 79] [18, 20, 24, 27]. Due to their design, the number of dimensions over which the variational posterior is defined is much lower.

6.1. Variational Auto-Encoder

Let $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ be a set of N i.i.d. observations of a continuous random variable $\mathbf{x} \in \mathcal{R}^D$. Further, let $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$ be a set of latent variables with $\mathbf{z}_n \in \mathcal{R}^{D_z}$ where $D_z << D$. Introducing latent variables to augment the observations represents a latent variable model for which we define

$$\log p(\mathbf{X}) = \log \int p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\theta})\, p(\mathbf{Z})\, \mathrm{d}\mathbf{Z}, \tag{6.1}$$

where $\boldsymbol{\theta}$ denotes some set of parameters of the conditional distribution $p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\theta})$. We may recognize that this formulation is very close to what we have discussed for maximum likelihood expectation-maximization. The difference here lies in the definition of the latent space $\mathcal{R}^{D_z}$ and the conditional distribution $p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\theta})$. While in EM we have assumed some mixture model such that the latent variables represent assignment variables to a

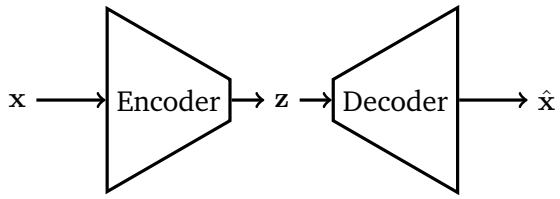


Figure 6.1.: Shown is a basic schematic for an Auto-Encoder. High dimensional samples $\mathbf{x} in \mathcal{R}^D$ are mapped to a lower dimensional feature representation $\mathbf{z} \in \mathcal{R}^{D_z}$ or latent space by the encoder, where $D_z << D$. The decoder reconstructs a sample $\hat{\mathbf{x}} \in \mathcal{R}^D$ from the latent variable $\mathbf{z}$ to be as close as possible to the original sample $\mathbf{x}$. Due to this design, Auto-Encoders are closely related to dimensionality reduction schemes such as principle component analysis.

component in the mixture for each observation $\mathbf{x}_n$, for VAEs we typically think of the latent space more as a reduced feature space of the observations [38, 79]. This point of view comes from the underlying concept of Auto-Encoders that are, in their behavior, closely related to dimensionality reduction (*Principle Component Analysis*) [80] [2, 38].

The Auto-Encoder is comprised of two elements, the encoder and the decoder, which are typically represented by some arbitrary neural network [38, 40]. The encoder takes as input an observation $\mathbf{x}_n$ from which it extracts a lower dimensional feature representation $\mathbf{z}_n$. Equivalently the decoder recovers an observation $\mathbf{x}_n$ from its latent representation $\mathbf{z}_n$ [38]. A representation of this construct is shown in Figure 6.1. Auto-Encoders are trained to minimize the error between the original observation and its reconstruction after going through the Auto-Encoder defined as

$$\text{loss} = \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|^2,$$

where $\hat{\mathbf{x}}_n = d\left(e\left(\mathbf{x}_n \,|\, \boldsymbol{\omega}\right) | \, \boldsymbol{\theta}\right)$ is the reconstructed observation with $e\left(\cdot \,|\, \boldsymbol{\omega}\right)$ being the encoder and $d\left(\cdot \,|\, \boldsymbol{\theta}\right)$ being the decoder [38]. The gradients for the two neural networks are described by

$$\nabla_{\boldsymbol{\omega}} \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|^2,$$
$$\nabla_{\boldsymbol{\theta}} \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|^2,$$

which are used to perform gradient descent.

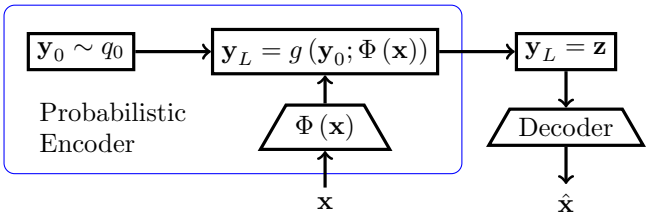


Figure 6.2.: Shown is the Variational Auto-Encoder for Normalizing Flows. The variational posterior is represented as a Normalizing Flow with a base distribution q_0 and a set of L transformations summarized by $g(\cdot)$. The parameters of the transformations are computed through some functional mapping $\Phi(\cdot)$ for an observation $\mathbf{x}$. The decoder $p(\mathbf{x} \mid \mathbf{z})$ reconstructs the observation as $\hat{\mathbf{x}}$ from the latent sample $\mathbf{z} = g(\mathbf{y}_0; \Phi(\mathbf{x}))$ [18].

With Equation (6.1) we have defined a latent variable model, where we can think of the conditional distribution $p(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\theta})$ as the decoder structure which generates a reconstruction of a sample $\mathbf{x}$ from a lower dimensional latent representation $\mathbf{z}$. The decoder is defined as some generative model, whose parameters are denoted by $\boldsymbol{\theta}$ [39, 40]. Further by considering a prior distribution $p(\mathbf{z})$ we can introduce a variational posterior $q(\mathbf{z} \mid \mathbf{X}, \boldsymbol{\omega})$ that replaces the encoder structure [39, 40]. By following the derivations of variational inference, we get a lower bound

$$\begin{aligned}\mathcal{F}(q(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\omega})) &= \int q(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\omega}) \log \frac{p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\theta})\, p(\mathbf{Z})}{q(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\omega})} \mathrm{d}\mathbf{Z} \\ &= \mathbb{E}_{q(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\omega})} \left[\log p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\theta}) + \log p(\mathbf{Z}) - \log q(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\omega})\right],\end{aligned}$$

which we compute as Monte-Carlo estimates. For the negative lower bound, we define the gradients

$$\begin{aligned}&\nabla_{\boldsymbol{\omega}} - \mathcal{F}(q(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\omega})), \\ &\nabla_{\boldsymbol{\theta}} - \mathcal{F}(q(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\omega})),\end{aligned}$$

to optimize a point estimate of the parameters of both encoder and decoder with SGD [39, 40]. Due to the relation to dimensionality reduction and by only considering point estimates over the parameters of encoder and decoder, VAEs scale much better to high dimensional problems compared to the design presented in this thesis. Especially considering that the variational posterior is described by a parametric model, the concept

of VAEs is easily enriched with Normalizing Flows [18, 20]. The idea was first presented in [18]. By defining a neural network with parameters $\boldsymbol{\omega}$, an observation $\mathbf{x}$ is mapped onto the parameters of a set of bijective and differentiable transformations g [18]. These transformations are used to transform a sample from a possibly trainable base distribution q_0 defined over the lower dimensional latent space into a sample $\mathbf{z}$ on the latent space [18]. A visualization of the approach is shown in Figure 6.2. The decoder in VAEs is designed as some *Deep Latent Variable Model* [40]. By defining the variational posterior over the reduced feature space $\mathbf{z}$ the dimensionality of the variational posterior is significantly lower compared to the variational posterior we defined for the parameters of a Bayesian Gaussian mixture. By further conditioning the parameters of the L transformations of the Normalizing Flow on observations $\mathbf{x}$ the number of dimensions in the variational posterior becomes independent of the dimensionality of $\mathbf{x}$ and the number of observations. This independence means that we can freely define the dimensions of $\mathbf{z}$ and thus the variational posterior in accordance with the decoder $p(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\theta})$ allowing for better scaling to high dimensional data. However, restricting the variational posterior to the lower dimension features $\mathbf{z}$ also entails that only a point-estimate over the parameters is optimized while the approaches we presented consider the full posterior.

The second part of Chapter 5 discussed the idea of separately updating each component in a mixture of Normalizing Flows using a decomposition of the lower bound. This decomposition follows the concept presented in [36, 37]. We have presented the decomposition for a mixture of $K = 6$ Normalizing Flows, each comprised of $L = 2$ affine coupling transformations on a Pinwheel density as a proof of concept. As a comparison in Figure A.3 we have added results on the same density for a single Normalizing Flow with $L = 12$ affine coupling transformations. The figure shows that the single Normalizing Flow is incapable of fully matching the ground truth density despite the same total amount of transformations. While learning mixtures comes with increased computational costs by having to differentiate every component w.r.t. the parameters of the components that get updated, the decomposition allows for parallel computation of the updates, alleviating the increased computational cost. However, the presented approach is still bound to certain restrictions. We rely on fast computation of the Normalizing Flows in both generative and normalizing direction, preventing efficient use of auto-regressive transformations. Further, the decomposition we presented is bound to the I-Projection. Thus, it is only applicable to density matching and posterior estimation in variational inference. A slightly different type of decomposition for mixtures of Normalizing Flows has been presented in [43] based on variational boosting [41, 42].

6.2. Boosted Normalizing Flows

Variational boosting improves upon the lower bound of the variational objective by iteratively refining the approximation with newly added components to the variational mixture distribution. We will consider the lower bound of the form

$$\mathcal{F}\left(q\left(\mathbf{x}\,|\,\boldsymbol{\omega}\right)\right) = \int q\left(\mathbf{x}\,|\,\boldsymbol{\omega}\right) \log \frac{p\left(\mathbf{x}\right)}{q\left(\mathbf{x}\,|\,\boldsymbol{\omega}\right)} \mathrm{d}\mathbf{x},$$

where $q\left(\mathbf{x}\,|\,\boldsymbol{\omega}\right) = \sum_{k=1}^{K} q\left(\mathbf{x}\,|\,k, \boldsymbol{\omega}_k\right) q\left(k\right)$ is some mixture distribution [41, 42]. Supplementing the definition of the mixture model, the lower bound changes to

$$\begin{aligned}
\mathcal{F}\left(q\left(\mathbf{x}\,|\,\boldsymbol{\omega}\right)\right) &= \int \sum_{k=1}^{K} q\left(\mathbf{x}\,|\,k, \boldsymbol{\omega}_k\right) q\left(k\right) \log \frac{p\left(\mathbf{x}\right)}{q\left(\mathbf{x}\,|\,\boldsymbol{\omega}\right)} \mathrm{d}\mathbf{x} \\
&= \sum_{k=1}^{K} q\left(k\right) \int q\left(\mathbf{x}\,|\,k, \boldsymbol{\omega}_k\right) \log \frac{p\left(\mathbf{x}\right)}{q\left(\mathbf{x}\,|\,\boldsymbol{\omega}\right)} \mathrm{d}\mathbf{x} \\
&= \sum_{k=1}^{K} q\left(k\right) \mathbb{E}_{q(\mathbf{x}\,|\,k, \boldsymbol{\omega}_k)} \left[\log p\left(\mathbf{x}\right) - \log q\left(\mathbf{x}\,|\,\boldsymbol{\omega}\right)\right],
\end{aligned}$$

which allows for a separate computation of the gradients for each component [42]. While, in the presented approach in Chapter 5 we iterate through the updates of all components in every episode, with boosting, only the most recently added component in the mixture is being updated. All previously existing components are kept fix as soon as a new component is added to the mixture [41–43]. Given we are at the point of adding component $K+1$ to the existing approximation $q\left(\mathbf{x}\,|\,\boldsymbol{\omega}\right)$ we define a new approximation as

$$q^{(K+1)}\left(\mathbf{x}\,|\,\boldsymbol{\omega}_{K+1}\right) = \left(1 - q\left(K+1\right)\right) q\left(\mathbf{x}\,|\,\boldsymbol{\omega}\right) + q\left(K+1\right) q\left(\mathbf{x}\,|\,K+1, \boldsymbol{\omega}_{K+1}\right),$$

where $q\left(K+1\right) \in [0, 1]$ [42]. This leads to a new lower bound

$$\begin{aligned}
\mathcal{F}\left(q^{(K+1)}\left(\mathbf{x}\,|\,\boldsymbol{\omega}_{K+1}\right)\right) =& \mathbb{E}_{q^{(K+1)}(\mathbf{x}\,|\,\boldsymbol{\omega}_{K+1})} \left[\log p\left(\mathbf{x}\right) - \log q^{(K+1)}\left(\mathbf{x}\,|\,\boldsymbol{\omega}_{K+1}\right)\right] \\
=& \left(1 - q\left(K+1\right)\right) \mathbb{E}_{q(\mathbf{x}\,|\,\boldsymbol{\omega})} \left[\log p\left(\mathbf{x}\right) - \log q^{(K+1)}\left(\mathbf{x}\,|\,\boldsymbol{\omega}_{K+1}\right)\right] \\
&+ q\left(K+1\right) \mathbb{E}_{q(\mathbf{x}\,|\,K+1, \boldsymbol{\omega}_{K+1})} \left[\log p\left(\mathbf{x}\right) - \log q^{(K+1)}\left(\mathbf{x}\,|\,\boldsymbol{\omega}_{K+1}\right)\right],
\end{aligned}$$

which is optimized w.r.t. $\boldsymbol{\omega}_{K+1}$ and $q\left(K+1\right)$ as the preexisting approximation $q\left(\mathbf{x}\,|\,\boldsymbol{\omega}\right)$ is kept fix [42]. The approach has been extended to mixtures of Normalizing Flows in [43].

7. Outlook

In our final approach to learning variational posterior distributions for a Bayesian Gaussian mixture model, we have considered a collapsed model, marginalizing out the discrete assignment variables $\mathbf{Z}$. While we achieve better results than traditional Mean-Field inference as derived in Chapter 3 the approach does not scale well to high dimensional problems and large mixture models because of the rapid increase in the number of dimensions of the variational posterior distribution. An interesting idea for the future is to consider a collapse of the parameters instead such that the posterior is a distribution over the assignment variables $q(\mathbf{Z})$. By further considering an amortized approach as presented in [18] introducing a functional mapping $\boldsymbol{\phi}_n = \Phi(\mathbf{x}_n)$ to parameterize the transformations of a Normalizing Flow we reduce the variational posterior $q(\mathbf{Z})$ to $q(\mathbf{z}_n \mid \boldsymbol{\phi})$ with $\mathbf{z} \in \mathcal{Z}^K$. Thus, the number of dimensions of the variational posterior only depends on the number of components in the Bayesian Gaussian mixture. By considering a collapse of the parameters, we further get an exact inference estimate of the parameters, which is an improvement compared to the point estimate of VAEs. However, as the assignment variables are discrete random variables, we cannot rely on any of the transformations presented in Chapter 4. Instead, we need some form of discrete Normalizing Flow.

7.1. Discrete Normalizing Flows

For discrete Normalizing Flows, the underlying concept of Normalizing Flows remains, that a complex distribution is described by transforming samples from a base distribution through a set of bijective transformations. Here the base distribution is some discrete distribution, such as a Categorical distribution. The difference to continuous Normalizing Flows lies in non-differentiable transformations [22, 23]. Instead of defining the density function of the complex distribution q_y as

$$q_y(\mathbf{y}) = q_x(\mathbf{x}) \left| \det \mathbf{J}_g(\mathbf{x}) \right|^{-1},$$

where q_x is the base distribution and $\mathbf{J}_g$ is the Jacobian of the transformation, the complex distribution for a discrete Normalizing Flow is described as

$$q_y(\mathbf{y}) = q_x(\mathbf{x}),$$

with $\mathbf{y} = g(\mathbf{x})$ [22, 23]. We no longer need to compute the absolute determinant of the Jacobian of the transformation, simplifying the computation.

A discrete Normalizing Flow based on an extension of the XOR function for a Categorical base distribution is presented in [22] called *Modulo location-scale* transform. This transformation is defined as

$$y_d = (\boldsymbol{\mu}_d + \boldsymbol{\sigma}_d x_d) \bmod K,$$

with $\mathbf{x} \in \mathcal{Z}_+^D$, where each element x_d takes on values in $0, 1, ..., K-1$. Further $\boldsymbol{\mu}_d$ and $\boldsymbol{\sigma}_d$ are some auto-regressive functions with discrete output [22]. For gradient tracking a continuous temperature-softmax relaxation of the auto-regressive functions is used during the backward pass [22].

In [23] a discrete coupling transform is proposed. The concept is similar to discrete coupling transformations. Given a sample $\mathbf{x} \in \mathcal{Z}^D$ split in two smaller vectors $\mathbf{x}_{1:d-1}$ and $\mathbf{x}_{d:D}$ at dimension d the transform is defined as

$$\begin{aligned} \mathbf{y}_{1:d-1} &= \mathbf{x}_{1:d-1}, \\ \mathbf{y}_{d:D} &= \mathbf{x}_{d:D} + \lfloor \mathrm{NN}(\mathbf{x}_{1:d-1}) \rceil, \end{aligned}$$

with $\lfloor \cdot \rceil$ being the nearest rounding operation and NN being a dense neural network [23]. To reduce gradient bias in the optimization the split dimension is chosen as $d \approx 0.75D$ [23].

It may be interesting to see how a collapsed variational inference approach for Bayesian Gaussian mixture models with discrete Normalizing Flows compares to the commonly used VAE in high dimensional data. However, while collapsing the parameters of the Bayesian Gaussian mixture might scale better to high dimensional data, we can also improve on the collapsed approach presented in Chapter 5. For the experiments, we have considered a Monte-Carlo estimate of the lower bound to compute an estimate of the gradient. The quality of the gradient estimate can be improved by considering *Importance Weighting* [81]. Introducing importance weighted updates might increase the speed and overall performance of the presented approach.

The approach presented in [36, 37] considers several ways to improve the performance of their algorithm for mixtures of Gaussians. Similarly, we could benefit from the same improvements for mixtures of Normalizing Flows presented in the second part of Chapter 5. To boost sample efficiency, O. Arenz et al. introduce a *replay buffer* to use past samples during the update of the mixing coefficients. A set of N samples is drawn from the replay buffer where for each sample an importance weight ϵ_n is computed as

$$\epsilon_n(k) = \frac{1}{C}\frac{q(\mathbf{x}_n \mid k)}{c(\mathbf{x}_n)}, \qquad C = \sum_{n=1}^{N}\frac{q(\mathbf{x}_n \mid k)}{c(\mathbf{x}_n)}.$$

Here k refers to the k^{th} component in the mixture model and $c(\mathbf{x}_n)$ denotes the distribution that generated the sample $\mathbf{x}_n$ [36, 37]. As shown in [36, 37] the Monte-Carlo estimate in Equation (5.3) is, therefore, replaced by

$$\tilde{R}(k) = \sum_{n=1}^{N}\epsilon_n\left[\log\tilde{q}(k \mid \mathbf{x}_n) + \log p(\mathbf{x}_n)\right] + H\left(q(\mathbf{x} \mid k)\right).$$

While the entropy $H(q(\mathbf{x} \mid k))$ for Gaussian components is computed easily, we cannot compute the entropy of arbitrary Normalizing Flows. In Chapter 5 we have, therefore, included the entropy in the Monte-Carlo estimate. We have yet to determine whether the entropy can still be included when using an importance weighted estimate to really benefit from this formulation.

As it might not be possible to determine the correct number of mixture components in advance, [36, 37] dynamically adjust the mixture's size during training by adding and removing components. As we have discussed before, components may converge onto the same modes of the target density. In these cases, one of these components dominates the contribution to the lower bound, causing the mixing coefficients of the other components to converge to zero. In [36, 37] all components with mixing coefficients below a certain threshold are removed as their contribution to the optimization is negligible. Though Normalizing Flows are less prone to overlap entirely due to their flexibility, it is still possible such that we could speed up learning by removing these components. Further, [36, 37] initialize learning with small mixtures and expand the mixture sequentially by adding new components at a certain rate. New components are added based on a heuristic that determines whether the new component is initialized close to the existing approximation to refine areas of the target density that have already been discovered or far away to discover new areas of high density in the target distribution [36, 37]. While we have considered a mixture distribution with a fixed size and with identical base distributions,

slowly introducing new components to an initially small mixture could further increase computation speed and allow the algorithm to initialize the new components with a base distribution in areas where the target density is not well matched.

Bibliography

[1] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001.

[2] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[3] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[4] D. Barber, *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[5] M. J. Beal, *Variational algorithms for approximate Bayesian inference*. PhD thesis, UCL (University College London), 2003.

[6] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. London: Chapman and Hall, 1996.

[7] I. Yildirim, "Bayesian inference: Gibbs sampling," *Technical Note, University of Rochester*, 2012.

[8] D. Van Ravenzwaaij, P. Cassey, and S. D. Brown, "A simple introduction to markov chain monte–carlo sampling," *Psychonomic bulletin & review*, 2018.

[9] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, 1999.

[10] M. J. Wainwright and M. I. Jordan, *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

[11] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for bayesian inference," *IEEE Signal Processing Magazine*, 2008.

[12] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, 2017.

[13] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt, "Advances in variational inference," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[14] G. Parisi, *Statistical field theory*. Addison-Wesley, 1988.

[15] L. K. Saul and M. I. Jordan, "Exploiting tractable substructures in intractable networks," in *Advances in neural information processing systems*, 1996.

[16] Z. Ghahramani and M. I. Jordan, "Factorial hidden markov models," *Machine learning*, 1997.

[17] E. G. Tabak, E. Vanden-Eijnden, *et al.*, "Density estimation by dual ascent of the log-likelihood," *Communications in Mathematical Sciences*, 2010.

[18] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," *arXiv preprint arXiv:1505.05770*, 2015.

[19] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2015.

[20] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," in *Advances in neural information processing systems*, 2016.

[21] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville, "Neural autoregressive flows," *arXiv preprint arXiv:1804.00779*, 2018.

[22] D. Tran, K. Vafa, K. Agrawal, L. Dinh, and B. Poole, "Discrete flows: Invertible generative models of discrete data," in *Advances in Neural Information Processing Systems*, 2019.

[23] E. Hoogeboom, J. Peters, R. van den Berg, and M. Welling, "Integer discrete flows and lossless compression," in *Advances in Neural Information Processing Systems*, 2019.

[24] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, "Neural spline flows," in *Advances in Neural Information Processing Systems*, 2019.

[25] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *arXiv preprint arXiv:1912.02762*, 2019.

[26] I. Kobyzev, S. Prince, and M. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[27] N. De Cao, W. Aziz, and I. Titov, "Block neural autoregressive flow," in *Uncertainty in Artificial Intelligence*, PMLR, 2020.

[28] H. M. Dolatabadi, S. Erfani, and C. Leckie, "Invertible generative modeling using linear rational splines," *arXiv preprint arXiv:2001.05168*, 2020.

[29] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, 1951.

[30] C. M. Bishop, "Latent variable models," in *Learning in graphical models*, Springer, 1998.

[31] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *The annals of mathematical statistics*, 1970.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, 1977.

[33] D. Barber and W. Wiegerinck, "Tractable variational structures for approximating graphical models," in *Advances in neural information processing systems*, 1999.

[34] M. D. Hoffman, "Stochastic structured mean-field variational inference," *arXiv preprint arXiv:1404.4114*, 2014.

[35] M. D. Hoffman and D. M. Blei, "Structured stochastic variational inference," in *Artificial Intelligence and Statistics*, 2015.

[36] O. Arenz, G. Neumann, and M. Zhong, "Efficient gradient-free variational inference using policy search," in *International conference on machine learning*, PMLR, 2018.

[37] O. Arenz, M. Zhong, and G. Neumann, "Trust-region variational inference with gaussian mixture models," *Journal of Machine Learning Research*, 2020.

[38] A. Ng *et al.*, "Sparse autoencoder," *CS294A Lecture notes*, 2011.

[39] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[40] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *arXiv preprint arXiv:1906.02691*, 2019.

[41] F. Guo, X. Wang, K. Fan, T. Broderick, and D. B. Dunson, "Boosting variational inference," *arXiv preprint arXiv:1611.05559*, 2016.

[42] A. C. Miller, N. J. Foti, and R. P. Adams, "Variational boosting: Iteratively refining posterior approximations," in *International Conference on Machine Learning*, PMLR, 2017.

[43] R. Giaquinto and A. Banerjee, "Gradient boosted flows," *arXiv preprint arXiv:2002.11896*, 2020.

[44] T. Minka, "Inferring a gaussian distribution," *Media Lab Note*, 1998.

[45] K. P. Murphy, "Conjugate bayesian analysis of the gaussian distribution," 2007.

[46] T. S. Haines, "Gaussian conjugate prior cheat sheet," 2011.

[47] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, 1948.

[48] A. Rényi *et al.*, "On measures of entropy and information," in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, The Regents of the University of California, 1961.

[49] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, 1991.

[50] D. J. MacKay and D. J. Mac Kay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[51] O. Johnson, *Information theory and the central limit theorem*. World Scientific, 2004.

[52] T. Van Erven and P. Harremos, "Rényi divergence and kullback-leibler divergence," *IEEE Transactions on Information Theory*, 2014.

[53] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[54] J. L. W. V. Jensen *et al.*, "Sur les fonctions convexes et les inégalités entre les valeurs moyennes," *Acta mathematica*, 1906.

[55] A. Bouchard-Côté and M. I. Jordan, "Optimization of structured mean field objectives," *arXiv preprint arXiv:1205.2658*, 2012.

[56] R. Ranganath, D. Tran, and D. Blei, "Hierarchical variational models," in *International Conference on Machine Learning*, 2016.

[57] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *The Journal of Machine Learning Research*, 2013.

[58] V. I. Bogachev, *Measure theory*. Springer Science & Business Media, 2007.

[59] J. Milnor and D. W. Weaver, *Topology from the differentiable viewpoint*. Princeton university press, 1997.

[60] V. I. Bogachev, A. V. Kolesnikov, and K. V. Medvedev, "Triangular transformations of measures," *Sbornik: Mathematics*, 2005.

[61] K. V. Medvedev, "Certain properties of triangular transformations of measures," 2008.

[62] G. Papamakarios, "Neural density estimation and likelihood-free inference," *arXiv preprint arXiv:1910.13233*, 2019.

[63] T. Müller, B. McWilliams, F. Rousselle, M. Gross, and J. Novák, "Neural importance sampling," *ACM Transactions on Graphics (TOG)*, 2019.

[64] R. D. Fuhr and M. Kallay, "Monotone linear rational spline interpolation," *Computer Aided Geometric Design*, 1992.

[65] J. Gregory and R. Delbourgo, "Piecewise rational quadratic interpolation to monotonic data," *IMA Journal of Numerical Analysis*, 1982.

[66] Z. Dong, B. Seybold, K. Murphy, and H. Bui, "Collapsed amortized variational inference for switching nonlinear dynamical systems," in *International Conference on Machine Learning*, PMLR, 2020.

[67] J. Marino, Y. Yue, and S. Mandt, "Iterative amortized inference," *arXiv preprint arXiv:1807.09356*, 2018.

[68] C. Cremer, X. Li, and D. Duvenaud, "Inference suboptimality in variational autoencoders," *arXiv preprint arXiv:1801.03558*, 2018.

[69] R. Shu, H. H. Bui, S. Zhao, M. J. Kochenderfer, and S. Ermon, "Amortized inference regularization," in *Advances in Neural Information Processing Systems*, 2018.

[70] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[71] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv preprint arXiv:1611.00712*, 2016.

[72] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. A. Szerlip, P. Horsfall, and N. D. Goodman, "Pyro: Deep universal probabilistic programming," *J. Mach. Learn. Res.*, 2019.

[73] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[74] J. Hensman, M. Rattray, and N. Lawrence, "Fast variational inference in the conjugate exponential family," *Advances in neural information processing systems*, 2012.

[75] Y. Teh, D. Newman, and M. Welling, "A collapsed variational bayesian inference algorithm for latent dirichlet allocation," *Advances in neural information processing systems*, 2006.

[76] K. Kurihara, M. Welling, and Y. W. Teh, "Collapsed variational dirichlet process mixture models.," in *IJCAI*, 2007.

[77] J. Sung, Z. Ghahramani, and S.-Y. Bang, "Latent-space variational bayes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

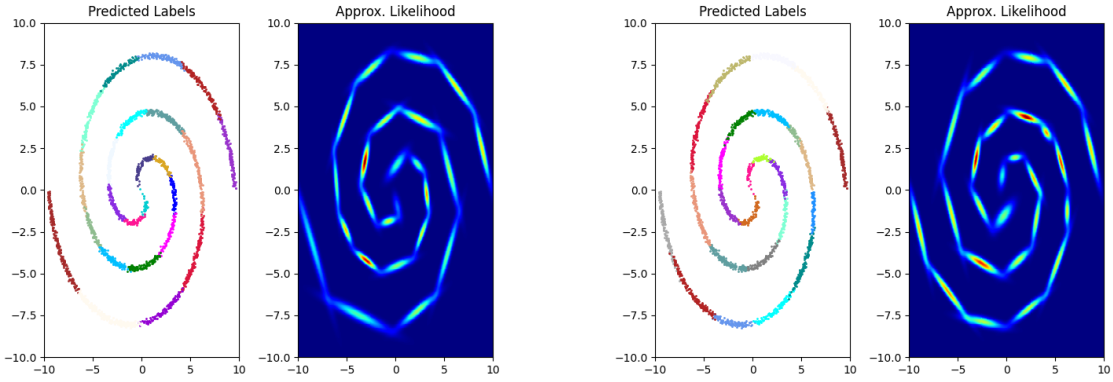
[78] A. Abdolmaleki, R. Lioutikov, J. R. Peters, N. Lau, L. Pualo Reis, and G. Neumann, "Model-based relative entropy stochastic search," *Advances in Neural Information Processing Systems*, 2015.

[79] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised learning with deep generative models," *Advances in neural information processing systems*, 2014.

[80] J. Lucas, G. Tucker, R. Grosse, and M. Norouzi, "Understanding posterior collapse in generative latent variable models," 2019.

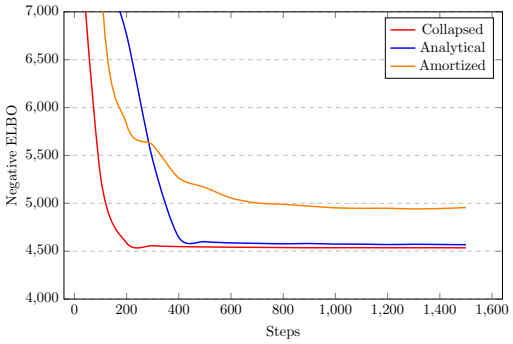
[81] Y. Burda, R. Grosse, and R. Salakhutdinov, "Importance weighted autoencoders," *arXiv preprint arXiv:1509.00519*, 2015.

A. Appendix

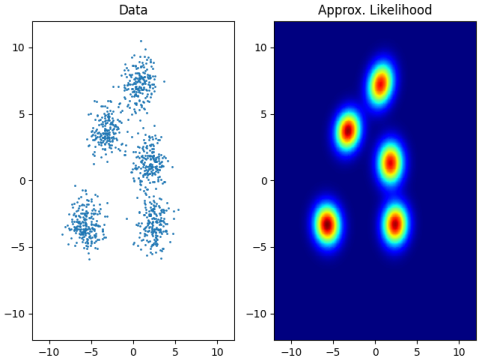


(a) Mode 1 (b) Mode 2

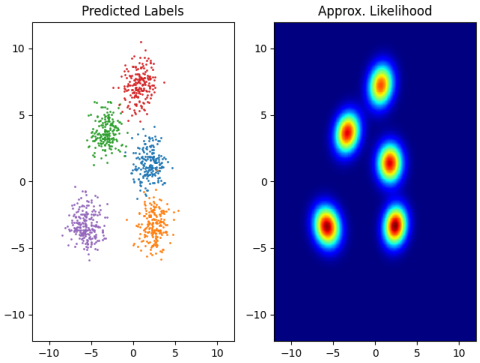
Figure A.1.: Shown are the results of variational inference with the analytically derived solution of the assignment variables on an up-scaled Two Spirals dataset. The dataset contains $N = 3000$ observations. The data is assumed to be distributed according to a Bayesian Gaussian mixture comprised of $K = 60$ components. The left side of each figure (a) and (b) visualizes the data, colorized according to the assignment variables. The right shows the corresponding approximate likelihood parameterized by the variational posterior $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$. Since the variational posterior is designed as a Normalizing Flow capable of representing multi-modal distributions we have encountered bi-modal behaviour in some experiments. This shows such an example where the analytical approach to the assignment variables trained two modes in parallel displayed by (a) and (b).



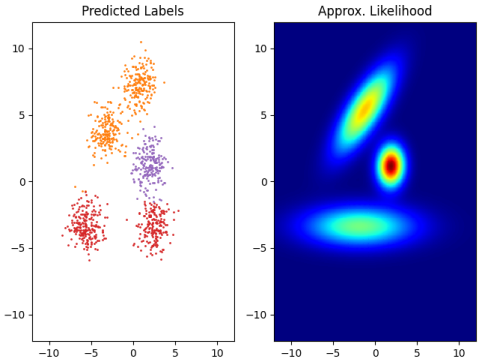
(a) Negative ELBO



(b) Collapsed VI



(c) Analytical $\mathbf{Z}$



(d) Amortized VI

Figure A.2.: Shown are the results of the three presented approaches on data sampled from randomly generated Gaussian mixture distributions with five equally weighted components. The dataset contains $N = 1000$ samples. Figure (a) presents the minimization of the negative lower bound. In (b) the dataset is displayed in a single coloration as the assignment variables are collapsed. The other figures show the data colorized according to the assignment variables. The approximate likelihood corresponds to $p(\mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ parameterized by the variational posterior $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$.

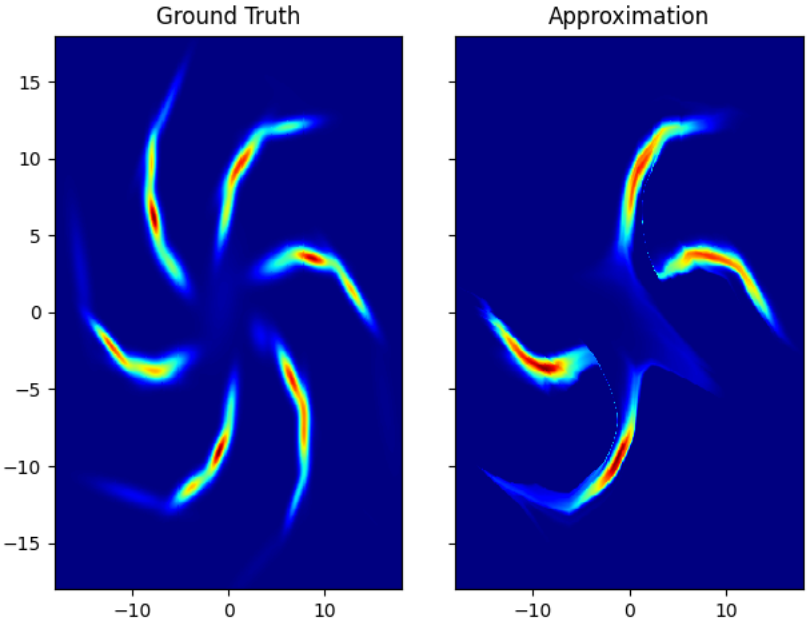


Figure A.3.: The figure summarizes the results of a single Normalizing Flow trained on a pinwheel density. The Flow is comprised of $L = 12$ affine coupling transformations. The left graphic shows the ground truth density $p(\mathbf{x})$, while right next to it, the Flow density is plotted. This figure serves as a comparison to the results for a mixture of Normalizing Flows trained on the same pinwheel density to highlight the increased versatility of adopting broader instead of deeper models.