EP Structured Variational Auto-Encoders

Jonathan So*

MSc Computational Statistics and Machine Learning
Supervisor: David Barber

September 2018

*This report is submitted as part requirement for the MSc Degree in Computational Statistics and Machine Learning at University College London. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

Acknowledgements

I would like to thank James Townsend, Benoit Gaujac and David Barber for taking the time to supervise my thesis. I would also like to thank Chris, my lifelong maths tutor.

Finally, thank you Antonia and Jessica for being so supportive of my studies this past year. This work is dedicated to you.

Abstract

Recent work combining neural network recognition models with probabilistic graphical models [1] has demonstrated the ability to perform fast, scalable (approximate) inference in rich, structured latent variable models that include non-linear, non-conjugate observations.

The approach described in [1] employs the use of mean-field variational inference for computing approximate posteriors over local latents. It is often the case that when performing approximate inference in latent variable models, expectation propagation (EP) [2][3] is able to yield more accurate posteriors than those found by mean-field, also resulting in more accurate parameters when used in a learning setting. This work investigates the applicability of EP for local-inference in the SVAE method, with further focus on whether such a scheme is able to improve upon the learned models found by the mean-field SVAE approach.

We demonstrate that using EP for local inference in an SVAE-like scheme is effective for the models we consider, including one for which the mean-field approach performs poorly. We also note that a further benefit of using EP in this setting is the greater flexibility it permits in our choice of recognition network outputs, which may allow us to significantly improve upon the approximations employed by the mean-field approach.

Contents

Chapter 1

Introduction

In this chapter we present an overview of the problem that we aim to address in this thesis, followed by a summary of related work in this area. We provide a brief overview of the subsequent chapters, followed by a list of notational conventions followed throughout the text.

The code used for the experiments presented in this thesis is available in an online repository accessible at the following URL

`https://github.com/jonny-so/svae/tree/epsvae`

Please note that this code is largely based on the code accompanying the original SVAE paper* [1]. A summary of the new contributions is available at

`https://github.com/jonny-so/svae/compare/master...epsvae`

1.1 Problem Description

Probabilistic graphical models provide us with a framework for building rich structured representations in latent variable models, as well as allowing access to a number of efficient exact and approximate inference routines based on exponential family message passing. Such models are often restrictive however, in that in order to make use of efficient message passing routines, we are typically

*We are grateful to the authors for making this code publicly available.

constrained to the class of conditionally conjugate models defined in Section 2.1. For many problems of interest this class of models can prove overly restrictive.

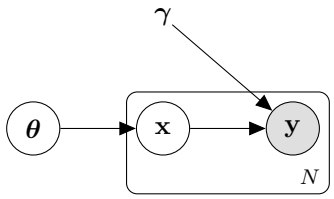

Figure 1.1: Directed graph for the general class of models we consider

The class of problems we consider here are those that consist of conditionally conjugate exponential family latent variables, combined with complex non-linear mappings from latent variables to observations. Formally, we consider the class of models of the form

$$p(\boldsymbol{\theta}) \prod_{n=1}^{N} p(\mathbf{x}^{(n)}|\boldsymbol{\theta}) p(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, \boldsymbol{\gamma}) \tag{1.1}$$

where $\boldsymbol{\theta}$ are the parameters corresponding to our latent variables $\mathbf{x}$, and $\boldsymbol{\gamma}$ are the parameters of our observation likelihoods, for observed variables $\mathbf{y}$. Each conditional density in (1.1) is individually in the exponential family, with the further restriction that $p(\mathbf{x}^{(n)}|\boldsymbol{\theta})$ is conjugate to $p(\boldsymbol{\theta})$. Note that while the conditional density $p(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, \boldsymbol{\gamma})$ is within the exponential family, we allow for the natural parameters to depend in arbitrary non-linear ways on the sufficient statistics of $p(\mathbf{x}^{(n)}|\boldsymbol{\theta})$, and so we lack conjugacy in the observation likelihoods.

For our purposes we consider $p(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, \boldsymbol{\gamma})$ to be parameterised by a neural network, with network weights $\boldsymbol{\gamma}$. It is a straightforward extension, although we have not done so here, to also permit an exponential family prior on the observation parameters $p(\boldsymbol{\gamma})$, see [1] for details.

We make a distinction between local latents $\{\mathbf{x}^{(n)}\}$, which are conditionally independent of each other given $\boldsymbol{\theta}$, and global latents / parameters $\boldsymbol{\theta}$, which influence all $\{\mathbf{x}^{(n)}\}$.

While (1.1) defines the general class of models considered here, we shall focus on the case of $N = 1$, allowing us to drop the superscript throughout this work in order to ease notation. The extension to multiple observations is

straightforward. Our model then becomes

$$p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})p(\mathbf{y}|\mathbf{x},\boldsymbol{\gamma}) \tag{1.2}$$

Note that the model class (1.2) permits $p(\mathbf{x}|\boldsymbol{\theta})$, $p(\mathbf{y}|\mathbf{x},\boldsymbol{\gamma})$ to have additional structure, as will typically be the case

The lack of conjugacy between latents $\mathbf{x}$ and observations $\mathbf{y}$ makes inference challenging, as the exact posterior over latents does not in general lie in any tractable parametric family. We therefore look to perform approximate inference in such a model.

Variational inference in models such as this where we lack convenient conjugacy structure throughout typically amounts to a computationally demanding optimisation problem which must be solved for each observation. Furthermore, in a learning scheme this expensive optimisation must be performed in an inner loop for each step of the learning process. We would therefore like the ability to perform approximate inference in this class of models efficiently, to reduce the cost of inference both during learning and for future unseen data. Furthermore, we would also like our inference scheme to be scalable to extremely large datasets.

The problem described here is exactly that addressed in [1], using an approach which the authors call the *structured variational auto-encoder* (SVAE). In this work, we consider a variant of the SVAE in which we utilise the expectation propagation algorithm [3] to perform approximate inference over local latent variables, rather than the mean-field variational inference method used in the original work. We shall refer to the existing and new approaches as SVAE and EP-SVAE respectively throughout.

The *expectation propagation* algorithm (EP) can itself be viewed as an approach to tackling the class of models we consider. In particular, EP provides a method for projecting intractable distributions into tractable exponential families. However, using EP directly on the models considered here requires computing the expected sufficient statistics of tractable families, but under *analytically* intractable tilted distributions (see Section 2.6). Running EP on such models would then typically require the use of numerical integration or sampling techniques in order to estimate the required expectations. Furthermore, these expectations would need to be estimated for each inference computation, both in the inner loop of our learning scheme and on future unseen data. We instead employ the use of recognition networks to produce tractable potentials to

replace the intractable observation likelihoods, so that inference computations can be performed efficiently. We shall define what it means for potentials to be tractable in this setting in later chapters.

1.2 Related Work

This work is primarily based on the approach of [1], in which the authors describe an approach called the structured variational auto-encoder (SVAE). We consider a variant of this approach which utilises the expectation propagation algorithm of [2][3] for local inference.

The SVAE can be seen as both a generalisation of the variational auto-encoder model of [4] and an extension of the stochastic variational inference method of [5] to models that lack conjugacy in the observation likelihoods.

We note that our approach is similar in aim to the stochastic-EP (SEP) method of [6]. Stochastic-EP can be seen as an extension of EP that can scale to extremely large datasets, with memory overhead that does not grow with the number of data points N. This is achieved by maintaining a single approximate local factor, capturing the average effect of all local factors, rather than storing an approximation for each individual factor. However, there is an implicit assumption in the use of SEP (and EP more generally) that we are able to efficiently compute moments under the tilted distributions (see section 2.6). Efficient closed form updates are only available in a handful of special cases and for the more general likelihood factors we consider here, computing the required expectations would typically involve the use of numerical integration or sampling techniques.

An extension of stochastic-EP which aims to mitigate this cost is the approach of [7], in which the author combines neural network recognition models with SEP in order to amortise the cost of the projection step so as to mitigate the limitation previously discussed. Our approach is similar in both aim and methods, but there are several significant differences which we highlight here.

The approach of [7] uses a recognition network to parameterise the full local approximate factor directly, whereas our approach, as with the SVAE, uses recognition networks to parameterise observation potentials within a larger structured latent graphical model, over which we then perform local inference using message passing routines. Optimisation of our objective then requires differentiating through the fixed point of the message passing iterations, which

is made possible using automatic differentiation techniques.

[7] employs the use of a single stochastic objective corresponding to the SEP algorithm of [6], whereas we employ a hybrid approach, targeting a global variational inference objective, whilst employing the use of a surrogate objective function for performing local inference, corresponding to the EP energy function of [8]. Such a hybrid approach can be thought of as similar in spirit to the approximate EM algorithms of [9], in which the exact E step of EM is replaced by an approximate E step using belief propagation or EP for computing approximate posteriors.

We also note that in [7] the author found gradient-based optimisation of the objective to be challenging due to the need to sample a normalising factor, resulting in both biased and noisy gradient estimates. The need to sample the normalising constant effectively stems from the use of the inclusive KL divergence as the optimisation objective. The approach we present here is able to avoid this issue by instead targeting the exclusive KL as our global objective.

1.3 Chapter Overview

Chapter 2 contains a number of technical preliminaries providing the foundations for the techniques presented in later chapters. In particular, we provide a brief introduction to exponential families of distributions, including a number of properties and definitions that we shall make use of throughout. We then discuss a number of techniques for approximate inference in probabilistic models which will provide the necessary building blocks for the SVAE and EP-SVAE.

Chapter 3 describes the SVAE approach of [1]. First presenting an overview of the method, followed by a discussion of the objective functions employed and corresponding optimisation schemes.

Chapter 4 introduces our main contribution, the EP-SVAE. We begin by discussing the motivation for our approach, including potential challenges, followed by a more detailed discussion of the surrogate objective function targeted by the technique.

Chapter 5 presents some experimental results, demonstrating the efficacy and characteristics of the EP-SVAE method on synthetic datasets. Chapter 6 discusses potential avenues for future research, finally followed by a summary of the findings in Chapter 7.

A number of derivations that were sufficiently long to distract from the flow

of the main text have been relegated to the appendix. We shall refer the reader to the relevant appendices where appropriate.

1.4 Notation

Throughout the text we shall make use of the following notational conventions

x	Scalar x
X	Matrix X
$\mathbf{x}$	Vector $\mathbf{x}$
x_i	i-th component of vector $\mathbf{x}$
$\mathbf{x}^\top$	Transpose of $\mathbf{x}$
$\mathbf{x}^\top\mathbf{y}$	Vector dot product of $\mathbf{x}$, $\mathbf{y}$
$. \otimes .$	Tensor product
$\delta(. = .)$	Kronecker delta function
$\nabla f(.)$	Gradient of f
$\nabla^2 f(.)$	Hessian of f
$\mathcal{S} = \{...\}$	Set $\mathcal{S}$
$\mathcal{S}^\circ$	Interior of set $\mathcal{S}$
$p(.)$	Probability distribution
$q(.)$	Approximate probability distribution
$\mathbf{t}(.)$	Exponential family sufficient statistic (vector-valued) function
$A(.)$	Exponential family cumulant / log-partition function
$Z(.)$	Exponential family normaliser ($A(.) = \ln Z(.)$)
$\boldsymbol{\eta}$	Prior natural parameters
$\boldsymbol{\tau}$	Variational / approximation natural parameters
$\langle . \rangle_p$	Expectation under p
$F[.]$	Functional F
$. \mapsto .$	Anonymous function
$\mathrm{vec}(.)$	Vectorisation operator

The convention of using upper case, lower case and bold to represent matrix, scalar and vector quantities respectively extends also to greek letters and functions. There are a couple of exceptions to this however, in particular $A(.)$, $Z(.)$ as noted above and the conventional $\mathcal{L}$ which we use to denote a variational lower-bound.

We shall often introduce subscripts when there are multiple variables involved. Ocassionally, when we need to index over multiple vector or matrix

quantities, we will use e.g. $\boldsymbol{\mu}(i,j)$ to denote the i,j-th vector $\boldsymbol{\mu}$. In cases where we have multiple i.i.d. variables, we shall use a superscript with brackets, e.g. $\mathbf{x}^{(n)}$ to denote the n-th such variable.

When there is no ambiguity, we shall occasionally find it convenient to use natural parameters η as a shorthand for $p(.|\eta)$, for example we may use $\langle . \rangle_\eta$ to denote expectation under an exponential family with natural parameters η when the family in question is clear from context.

Finally, we sometimes mix scalar, vector or matrix valued quantities in the definition of a vector, e.g. $\theta = (A, \mathbf{b}, c)$. In such cases, this should be read as vectorisation of each argument with the results concatenated together into a single vector.

Chapter 2

Theoretical Preliminaries

In this chapter we provide an overview of several topics which form the foundations for the techniques presented in later chapters. We do not aim to provide a comprehensive treatment of the subjects presented in this chapter, rather we focus on those aspects that are particularly relevant to the problem at hand. For an excellent introduction to many of the topics covered here, in particular exponential families and variational inference, we refer the reader to [10]. For further details on stochastic variational inference, see [5]. For amortized variational inference and the variational auto-encoder, see [4], [11]. For a detailed treatment of expectation propagation, see [2], [3], [8].

2.1 Exponential Families

The following section is largely based on [10]. We aim to summarise the most relevant topics for our purposes and refer the reader to [10] for a more complete treatment.

An exponential family of distributions on the random variable $\mathbf{x}$ with respect to an underlying base measure ν, is defined by a vector-valued function $\mathbf{t}_x(\mathbf{x})$ known as the sufficient statistics of the distribution. The family of distributions defined by $\mathbf{t}_x(\mathbf{x})$ with respect to ν then takes the following form

$$p(\mathbf{x} \mid \boldsymbol{\eta}) = \exp\{\boldsymbol{\eta}^\top \mathbf{t}_x(\mathbf{x}) - A(\boldsymbol{\eta})\} \tag{2.1}$$

where $A(\boldsymbol{\eta}) = \log \int \exp\{\boldsymbol{\eta}^\top \mathbf{t}_x(\mathbf{x})\} \mathrm{d}\nu(\mathbf{x})$ is the log-partition, or cumulant func-

tion, that ensures the distribution is correctly normalised. We will ocassionally refer instead to the normaliser $Z(\boldsymbol{\eta}) = e^{A(\boldsymbol{\eta})}$ when convenient.

The base measure ν is typically the counting measure for discrete variables, or the Lebesgue measure for continuous variables. It is sometimes convenient to multiply (2.1) by a base factor $h(\mathbf{x})$, however this can be incorporated into the base measure ν and so we shall not consider $h(\mathbf{x})$ further here. The base measure may also be used to impose so called *hard-core* constraints on $\mathbf{x}$ whilst remaining within the exponential family. For example, in error-control coding problems in coding theory, certain configurations of $\mathbf{x}$ for which parity-checks fail are disallowed. This can be handled in the exponential family framework by incorporating parity-check indicators into $\nu(\mathbf{x})$ [10]. Note that we shall drop explicit reference to the base measure going forwards except where it is helpful for the discussion at hand.

The components of $\boldsymbol{\eta}$ are known as the natural or canonical parameters of the distribution, and a defining characteristic of an exponential family distribution is the linear interaction between the natural parameters and the sufficient statistics in the exponent. From (2.1) the reason for calling $\mathbf{t}_x(\mathbf{x})$ the sufficient statistic vector becomes clear. By the Fisher-Neyman factorisation theorem [12] [13], we know that any inferences about $\boldsymbol{\eta}$ in the data must be a function of $\mathbf{t}_x(\mathbf{x})$ only, and so $\mathbf{t}_x(\mathbf{x})$ are sufficient for $\boldsymbol{\eta}$.

The set of permissible natural parameters is defined as

$$\Omega := \{\boldsymbol{\eta} \in \mathbb{R}^d : A(\boldsymbol{\eta}) < \infty\} \tag{2.2}$$

We say that an exponential family is *regular* if Ω is an open set. For the remainder of this thesis we shall consider only regular famiels. An exponential family is said to be *minimal* if there is no $\boldsymbol{\eta} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ such that $\boldsymbol{\eta}^\top \mathbf{t}_x(\mathbf{x}) = 0$ $\forall \mathbf{x}$ [1]. For minimal families, any distribution in the family is associated with a unique natural parameter vector $\boldsymbol{\eta}$. An exponential family that is not minimal is said to be *overcomplete*, in which case each distribution is associated with an entire affine subspace of Ω.

For any given distribution p, not necessarily in the exponential family, we can compute the expected sufficient statistics vector under this distribution

$$\boldsymbol{\mu} := \langle \mathbf{t}_x(\mathbf{x}) \rangle_p \tag{2.3}$$

For a given statistic function $\mathbf{t}_x(\mathbf{x})$, let us denote by $\mathcal{M}$ the set of all vectors

$\boldsymbol{\mu} \in \mathbb{R}^d$ that can be realised by *any* distribution

$$\mathcal{M} := \{\boldsymbol{\mu} \in \mathbb{R}^d \mid \exists p : \langle \mathbf{t}_x(\mathbf{x}) \rangle_p = \boldsymbol{\mu}\} \tag{2.4}$$

For discrete $\mathbf{x}$, $\mathcal{M}$ is known as the marginal polytope. The set $\mathcal{M}$ is a convex subset of $\mathbb{R}^d$, as can be seen by noting that for any two vectors $\boldsymbol{\mu}$, $\boldsymbol{\mu}'$ realised by distributions p and p' respectively, any convex combination of $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$ is realised by the same convex combination of p and p'. It turns out that although $\mathcal{M}$ was defined by allowing expectations under any disitribution, any vector $\boldsymbol{\mu}$ within the interior $\mathcal{M}^\circ$ is realisable by a member of the exponential family defined by the sufficient statistics $\mathbf{t}_x(\mathbf{x})$. That is, $\forall \boldsymbol{\mu} \in \mathcal{M}^\circ$, $\exists \boldsymbol{\eta} \in \{\Omega : \langle \mathbf{t}_x(\mathbf{x}) \rangle_\eta = \boldsymbol{\mu}\}$ (see [10] for a proof). Furthermore, this distribution is the maximum entropy distribution among all distributions with $\langle \mathbf{t}_x(\mathbf{x}) \rangle = \boldsymbol{\mu}$.

The vector $\boldsymbol{\mu}$ therefore provide an alternative parameterisation of the exponential family with sufficient statistic vector $\mathbf{t}_x(\mathbf{x})$, and for this reason we refer to $\boldsymbol{\mu}$ as the *mean parameters* of the distribution. Furthermore, for minimal families, there is a one-to-one mapping from natural parameters to mean parameters. For overcomplete families, this mapping is in general many-to-one.

Many fundamental inference problems in exponential family models are equivalent to performing either the mapping from natural parameters to mean parameters (the *forward mapping*) or vice-versa (the *reverse mapping*). For example, the expected sufficient statistics for a given natural parameter vector may themselves be directly of interest, in particular in marginalisation problems. Maximum likelihood estimation of the natural parameters is equivalent to applying the reverse mapping to the empirical average suffucient statistics, $\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{t}_x(\mathbf{x}^{(i)})$.

The cumulant function $A(\boldsymbol{\eta})$ plays a central role in many problems of inference in exponential families. Two particularly important identies are given below:

Theorem 2.1.1.

$$\nabla A(\boldsymbol{\eta}) = \langle \mathbf{t}_x(\mathbf{x}) \rangle_\eta \tag{2.5}$$

$$\nabla^2 A(\boldsymbol{\eta}) = \langle \mathbf{t}_x(\mathbf{x}) \mathbf{t}_x(\mathbf{x})^\top \rangle_\eta - \langle \mathbf{t}_x(\mathbf{x}) \rangle_\eta \langle \mathbf{t}_x(\mathbf{x})^\top \rangle_\eta \tag{2.6}$$

Proof.

$$\begin{aligned}
\nabla A(\boldsymbol{\eta}) &= \int \nabla_\eta \exp\{\boldsymbol{\eta}^\top \mathbf{t}_x(\mathbf{x}) - A(\boldsymbol{\eta})\}\mathrm{d}\mathbf{x} \\
&= \int \mathbf{t}_x(\mathbf{x}) \exp\{\boldsymbol{\eta}^\top \mathbf{t}_x(\mathbf{x}) - A(\boldsymbol{\eta})\}\mathrm{d}\mathbf{x} \\
&= \langle \mathbf{t}_x(\mathbf{x}) \rangle_\eta \\
\nabla^2 A(\boldsymbol{\eta}) &= \int \nabla_\eta \mathbf{t}_x(\mathbf{x}) \exp\{\boldsymbol{\eta}^\top \mathbf{t}_x(\mathbf{x}) - A(\boldsymbol{\eta})\}\mathrm{d}\mathbf{x} \\
&= \int \mathbf{t}_x(\mathbf{x}) \big(\mathbf{t}_x(\mathbf{x}) - \langle \mathbf{t}_x(\mathbf{x})^\top \rangle_\eta\big) \exp\{\boldsymbol{\eta}^\top \mathbf{t}_x(\mathbf{x}) - A(\boldsymbol{\eta})\}\mathrm{d}\mathbf{x} \\
&= \langle \mathbf{t}_x(\mathbf{x}) \mathbf{t}_x(\mathbf{x})^\top \rangle_\eta - \langle \mathbf{t}_x(\mathbf{x}) \rangle_\eta \langle \mathbf{t}_x(\mathbf{x}) \rangle_\eta^\top
\end{aligned}$$

□

In fact, it is true that the cumulant function $A(\boldsymbol{\eta})$ of any regular exponential family has derivatives of all orders on its domain Ω, and the n-th derivative of $A(\boldsymbol{\eta})$ yields the n-th cumulant of $\mathbf{t}_x(\mathbf{x})$. An important corrolary of (2.6) is that for regular families, $A(\boldsymbol{\eta})$ is a convex function of $\boldsymbol{\eta} \in \Omega$ (and strictly convex for minimal families).

Theorem 2.1.1 tells us that $\nabla A(\boldsymbol{\eta})$ provides the forward mapping from natural to mean parameters. We say that a pair $(\boldsymbol{\mu}, \boldsymbol{\eta})$ are *dually coupled* if $\boldsymbol{\mu} = \nabla A(\boldsymbol{\eta})$. Note that this definition does not require the mapping to be one-to-one. The relationship between dually coupled parameters motivates the following notation

$$\boldsymbol{\mu}(\boldsymbol{\eta}) := \langle \mathbf{t}_x(\mathbf{x}) \rangle_\eta \tag{2.7}$$

$$\boldsymbol{\eta}(\boldsymbol{\mu}) := \{\boldsymbol{\eta} : \langle \mathbf{t}_x(\mathbf{x}) \rangle_\eta = \boldsymbol{\mu}\} \tag{2.8}$$

For our purposes we consider a *tractable* exponential family to be one for which we can compute $A(\boldsymbol{\eta})$ (and hence its derivatives). While the reverse mapping may still be potentially intractable, what we require for our objectives are the ability to evaluate the density of a distribution and its expected sufficient statistics, both of which become easy provided that we can compute $A(\boldsymbol{\eta})$.

We say that a prior distribution $p(\mathbf{x})$ is conjugate to a conditional $p(\mathbf{y}|\mathbf{x})$ if the posterior of $\mathbf{x}$ having observed $\mathbf{y}$ remains within the same parametric family of distributions. In the context of exponential family distributions, conjugacy in

this simple case is equivalent to the terms of $\log p(\mathbf{y}|\mathbf{x})$ being an affine function of the sufficient statistics $\mathbf{t}_x(\mathbf{x})$. For simple conjugacy relationships such as this, the sufficient statistics of $\mathbf{x}$ are typically given by

$$\mathbf{t}_x(\mathbf{x}) = (\boldsymbol{\eta}_y(\mathbf{x}), -A(\boldsymbol{\eta}_y(\mathbf{x})))$$

where $\dim(\mathbf{t}_x(\mathbf{x})) = \dim(\boldsymbol{\eta}_x(\mathbf{x})) + 1$. For this reason we occasionally find it convenient to define the augmented sufficient statistic vector $\mathbf{t}'_y(\mathbf{y}) := (\mathbf{t}_y(\mathbf{y}), 1)$, so that our joint distribution becomes

$$\begin{aligned} p(\mathbf{x}, \mathbf{y}) &= \exp\{\boldsymbol{\eta}_x^\top \mathbf{t}_x(\mathbf{x}) - A_x(\boldsymbol{\eta}_x)\} \exp\{\boldsymbol{\eta}_y(\mathbf{x})^\top \mathbf{t}_y(\mathbf{y}) - A_y(\boldsymbol{\eta}_y(\mathbf{x}))\} \\ &= \exp\{(\boldsymbol{\eta}_x + \mathbf{t}'_y(\mathbf{y}))^\top \mathbf{t}_x(\mathbf{x}) - A_x(\boldsymbol{\eta}_x)\} \end{aligned}$$

The above representation becomes particularly convenient when implementing message passing schemes that exploit conjugacy structure.

In order to generalise the notion of conjugacy we first introduce the definition of a *multi-affine* function, as defined by [14].

Definition 2.1.1. *We say that a function $f(\mathbf{x}_1, ..., \mathbf{x}_M)$ is multi-affine in the vectors $\mathbf{x}_1, ..., \mathbf{x}_M$, if*

$$f(\mathbf{x}_1, ..., \mathbf{x}_M) = \sum_{\boldsymbol{\beta} \in \mathcal{B}} \boldsymbol{\eta}_\beta^\top vec(\mathbf{x}_1^{\beta_1} \otimes ... \otimes \mathbf{x}_M^{\beta_M})$$

where $\mathcal{B} \subseteq \{0, 1\}^M$ is an index set and we take $\mathbf{x}_i^0 = 1$.

In effect a multi-affine function $f(\mathbf{x}_1, ..., \mathbf{x}_M)$ is one which is an affine function when viewed as a function of any one of its M vector arguments individually. We can now introduce the class of *conditionally conjugate* exponential family models, again following the definition of [14].

Definition 2.1.2. *An exponential family $\mathcal{F}$ is conditionally conjugate if the negative energy function of the family*

$$\log p(\mathbf{x}) = f(\mathbf{t}_{x_1}(\mathbf{x}_1), ..., \mathbf{t}_{x_M}(\mathbf{x}_M))$$

is multi-affine in tractable statistic functions $\mathbf{t}_{x_1}(\mathbf{x}_1), ..., \mathbf{t}_{x_M}(\mathbf{x}_M)$, where $\{\mathbf{x}_1, ..., \mathbf{x}_M\}$ is a partition of the co-ordinates of $\mathbf{x}$.

This class of models permits a number of efficient exact and approximate inference schemes that exploit the conjugacy structure based on exponential

family message passing. We shall make use of a number of these schemes in subsequent chapters.

Theorem 2.1.2. *The KL divergence between two distributions within the same exponential family with natural parameters $\boldsymbol{\eta}_1$, $\boldsymbol{\eta}_2$ and mean parameters $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ respectively, is given by:*

$$KL[\boldsymbol{\eta}_1 \,||\, \boldsymbol{\eta}_2] = (\boldsymbol{\eta}_1 - \boldsymbol{\eta}_2)^\top \boldsymbol{\mu}_1 - A(\boldsymbol{\eta}_1) + A(\boldsymbol{\eta}_2) \tag{2.9}$$

Proof.

$$\begin{aligned}\mathrm{KL}[\boldsymbol{\eta}_1 \,||\, \boldsymbol{\eta}_2] &= \int \exp\{\boldsymbol{\eta}_1^\top t(\mathbf{x}) - A(\boldsymbol{\eta}_1)\}(\boldsymbol{\eta}_1^\top \mathbf{t}_x(\mathbf{x}) - A(\boldsymbol{\eta}_1) - \boldsymbol{\eta}_2^\top \mathbf{t}_x(\mathbf{x}) + A(\boldsymbol{\eta}_2))\mathrm{d}\mathbf{x} \\ &= (\boldsymbol{\eta}_1 - \boldsymbol{\eta}_2)^\top \boldsymbol{\mu}_1 - A(\boldsymbol{\eta}_1) + A(\boldsymbol{\eta}_2)\end{aligned}$$

□

An interesting interpretation of (2.9), is that the KL divergence between members of the same exponential family is the difference between $A(\boldsymbol{\eta}_2)$ and its linear approximation centered around $\boldsymbol{\eta}_1$

$$\begin{aligned}\mathrm{KL}[\boldsymbol{\eta}_1 \,||\, \boldsymbol{\eta}_2] &= (\boldsymbol{\eta}_1 - \boldsymbol{\eta}_2)^\top \boldsymbol{\mu}_1 - A(\boldsymbol{\eta}_1) + A(\boldsymbol{\eta}_2) \\ &= A(\boldsymbol{\eta}_2) - \big(A(\boldsymbol{\eta}_1) + \boldsymbol{\mu}_1(\boldsymbol{\eta}_2 - \boldsymbol{\eta}_1)\big) \\ &= A(\boldsymbol{\eta}_2) - \big(A(\boldsymbol{\eta}_1) + \nabla A(\boldsymbol{\eta}_1)(\boldsymbol{\eta}_2 - \boldsymbol{\eta}_1)\big)\end{aligned}$$

We restate here a useful result from [14].

Theorem 2.1.3. *For a minimal exponential family, let f and g be real-valued functions on $\mathbb{R}^d$ such that $f(\boldsymbol{\mu}) = g(\boldsymbol{\eta}(\boldsymbol{\mu}))$ for all $\boldsymbol{\mu} \in \mathcal{M}^\circ$, then for a dually-coupled pair $(\boldsymbol{\eta}, \boldsymbol{\mu})$, we have*

$$\nabla f(\boldsymbol{\mu}) = [\nabla^2 A(\boldsymbol{\eta})]^{-1} \nabla g(\boldsymbol{\eta})$$

Theorem 2.1.3 tells us that for a minimal exponential family, the natural gradient (described in Section 3.3) of a function of the natural parameters $\boldsymbol{\eta}$ is equal in coefficients to the standard gradient under a mean-parameterisation evaulated at $\boldsymbol{\mu}(\boldsymbol{\eta})$. Furthermore, this gives us a way to compute the natural gradient without explicitly inverting the Hessian of $A(.)$. For a proof of Theorem 2.1.3 see [14].

The theorem stated above is valid only for minimal exponential families. However, we show here that an analogous result holds true for overcomplete families.

Theorem 2.1.4. *For an overcomplete exponential family, let f and g be real-valued functions on $\mathbb{R}^d$ such that $f(\boldsymbol{\mu}(\boldsymbol{\eta})) = g(\boldsymbol{\eta})$ for all $\boldsymbol{\eta} \in \Omega$. Then,*

$$g(\boldsymbol{\eta}) = g^*(F\boldsymbol{\eta})$$

for some function g^* where F maps the natural parameters to those of an equivalent minimal family, and for a dually-coupled pair $(\boldsymbol{\eta}, \boldsymbol{\mu})$, we have

$$F\nabla f(\boldsymbol{\mu}) = [\nabla^2 A^*(F\boldsymbol{\eta})]^{-1} \nabla g^*(F\boldsymbol{\eta})$$

where A^* is the cumulant function of the minimal family.

Theorem 2.1.4 is useful because it tells us that taking a step in the natural parameters $\boldsymbol{\eta}$ in the direction of $\nabla f(\boldsymbol{\mu})$ is equivalent to taking a natural gradient step in a minimal representation of the parameters $\boldsymbol{\eta}^* = F\boldsymbol{\eta}$, i.e.

$$F(\boldsymbol{\eta} + \nabla f(\boldsymbol{\mu})) = \boldsymbol{\eta}^* + [\nabla^2 A^*(\boldsymbol{\eta}^*)]^{-1} \nabla g^*(\boldsymbol{\eta}^*)$$

This is a new result and a proof is given in appendix A.1.

2.2 Variational Inference

In latent variable models, a key quantity of interest is the marginal likelihood of the data, or (log) model evidence

$$\log p(\mathbf{y}) = \log \int p(\mathbf{x}, \mathbf{y}) \mathrm{d}\mathbf{x} \tag{2.10}$$

where $\mathbf{y}$ are observed and $\mathbf{x}$ are latent variables including parameters. Variational inference refers to a class of approximations in which we attempt to maximise a lower bound on the quantity (2.10) by making use of an approxi-

mating distribution over latents $q(\mathbf{x})$. From (2.10) we have

$$\begin{aligned}\log p(\mathbf{y}) &= \log \int q(\mathbf{x}) \frac{p(\mathbf{x}, \mathbf{y})}{q(\mathbf{x})} \mathrm{d}\mathbf{x} \\ &\geq \int q(\mathbf{x}) \log \frac{p(\mathbf{x}, \mathbf{y})}{q(\mathbf{x})} \mathrm{d}\mathbf{x}\end{aligned} \tag{2.11}$$

where the second relation follows from Jensen's inequality. Let

$$\mathcal{L}[q(\mathbf{x})] := \int q(\mathbf{x}) \log \frac{p(\mathbf{x}, \mathbf{y})}{q(\mathbf{x})} \mathrm{d}\mathbf{x} \tag{2.12}$$

so then we have

$$\mathcal{L}[q(\mathbf{x})] \leq \log p(\mathbf{y}) \tag{2.13}$$

for all $q(\mathbf{x})$. Note that in the lower bound $\mathcal{L}[q(\mathbf{x})]$ is a functional of the distribution $q(\mathbf{x})$. The bound (2.12) can be made tight if we choose $q(\mathbf{x}) = p(\mathbf{x}|\mathbf{y})$, as can be seen by rewriting as follows

$$\begin{aligned}\mathcal{L}[q(\mathbf{x})] &= \int q(\mathbf{x}) \log p(\mathbf{y}) \mathrm{d}\mathbf{x} - \int q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x} \mid \mathbf{y})} \mathrm{d}\mathbf{x} \\ &= \log p(\mathbf{y}) - \mathrm{KL}[q(\mathbf{x}) \mid\mid p(\mathbf{x} \mid \mathbf{y})]\end{aligned} \tag{2.14}$$

The divergence $\mathrm{KL}[q(\mathbf{x}) \mid\mid p(\mathbf{x} \mid \mathbf{y})]$ takes its minimum value of 0 when $q(\mathbf{x}) = p(\mathbf{x} \mid \mathbf{y})$, at which point the bound becomes tight.

It is worth observing that (2.13) holds for any distribution $q(\mathbf{x})$. For many problems of interest the true posterior $p(\mathbf{x}|\mathbf{y})$ may be intractable. Variational inference encompasses a range of methods that choose an approximating distribution $q(\mathbf{x})$ within some constrained, tractable family of distributions which we denote $\mathcal{Q}$, so as to maximise the lower bound (2.12).

$$q^*(\mathbf{x}) := \underset{q(\mathbf{x}) \in \mathcal{Q}}{\arg\max}\, \mathcal{L}[q(\mathbf{x})] \tag{2.15}$$

Having performed the maximisation, we find a distribution $q(\mathbf{x})$ that is hopefully close (in KL sense) to the true posterior and so as a by-product of maximising the lower bound we obtain a way of performing approximate but tractable inference queries over latents $\mathbf{x}$.

If $q(\mathbf{x})$ is constrained to lie in some parametric family, performing variational inference amounts to performing optimisation of the objective (2.12) with respect to the parameters of the chosen family $\boldsymbol{\tau}$, known as the variational parameters.

$$\boldsymbol{\tau}^* = \arg\max_{\boldsymbol{\tau}} \mathcal{L}[q(\mathbf{x}|\boldsymbol{\tau})] \tag{2.16}$$

Provided that we can evaluate (2.12), this can be performed using standard optimisation techniques.

2.3 Mean Field Variational Inference

Mean field variational inference (MFVI) refers to a class of variational approximations in which the approximating family $\mathcal{Q}$ is one that factors over disjoint cliques (subsets of co-ordinates of $\mathbf{x}$)*. For a large class of models, through judicious choice of factorisation we can perform constrained maximisation of (2.14) using an efficient scheme based on exponential family message passing.

If we consider the class of conditionally conjugate models as described in section 2.1 that is multi-affine (see definition 2.1.1) in the tractable statistic functions $\mathbf{t}_{x_1}(\mathbf{x}_1), ..., \mathbf{t}_{x_M}(\mathbf{x}_M)$, where $\{\mathbf{x}_1, ..., \mathbf{x}_M\}$ forms a partition of the latents $\mathbf{x}$, a natural approximating family $\mathcal{Q}$ is then one that factorises over the partition $\{\mathbf{x}_1, ..., \mathbf{x}_M\}$.

$$\mathcal{Q} = \left\{ q(\mathbf{x}) : q(\mathbf{x}) = \prod_i^M q_i(\mathbf{x}_i) \right\} \tag{2.17}$$

Our constrained variational objective is then defined as

$$\mathcal{L}_{\mathrm{MF}}[\{q_i(\mathbf{x}_i)\}] = \int \left(\prod_i q_i(\mathbf{x}_i) \right) \log \frac{p(\mathbf{x}, \mathbf{y})}{\prod_i q_i(\mathbf{x}_i)} \mathrm{d}\mathbf{x} \tag{2.18}$$

Given this imposed factorisation, the optimal (in KL sense) factor $q_i(\mathbf{x}_i)$ given

*Some authors use the term *structured* mean-field to refer to this general class of approximations and reserve use of the term mean-field to refer exclusively to the class of fully factorised approximations. We make no such distinction here.

all other factors $q_j(\mathbf{x}_j) : j \neq i$ is given by

$$q_i(\mathbf{x}_i) \propto \exp\langle \log p(\mathbf{x}, \mathbf{y}) \rangle_{q_{\setminus i}(\mathbf{x}_{\setminus i})} \tag{2.19}$$

where we have introduced the notation $q_{\setminus i}(\mathbf{x}_{\setminus i}) = \prod_{j \neq i} q_j(\mathbf{x}_j)$. See [15] for a proof.

For the class of conditionally conjugate models (see Definition 2.1.2), the update (2.19) takes on a particularly convenient form. In particular, due to the multi-affine form of the negative energy, we have

$$\log p(\mathbf{x}, \mathbf{y}) = f\big(\mathbf{t}_{x_1}(\mathbf{x}_1), ..., \mathbf{t}_{x_M}(\mathbf{x}_M)\big) \tag{2.20}$$

where f is a multi-affine function of the statistic functions $\{\mathbf{t}_{x_1}(\mathbf{x}_1), ..., \mathbf{t}_{x_M}(\mathbf{x}_M\})$. The update (2.19) therefore has the form

$$q_i(\mathbf{x}_i) \propto \exp\{\boldsymbol{\tau}_{x_i}(\{\boldsymbol{\mu}_j : j \neq i\})^\top \mathbf{t}_{x_i}(\mathbf{x}_i)\} \tag{2.21}$$

where

$$\boldsymbol{\tau}_{x_i}(\{\boldsymbol{\mu}_j : j \neq i\})^\top \mathbf{t}_{x_i}(\mathbf{x}_i) = f\big(\boldsymbol{\mu}_1, ..., \mathbf{t}_{x_i}(\mathbf{x}_i), ..., \boldsymbol{\mu}_M\big) + c \tag{2.22}$$

and so $\boldsymbol{\tau}_{x_i}(\{\boldsymbol{\mu}_j : j \neq i\})$ is a multi-affine function of the expected sufficient statistics $\boldsymbol{\mu}_j := \langle \mathbf{t}_{x_j}(\mathbf{x}_j) \rangle_{q_j} : j \neq i$, which are by assumption easily computable. We therefore see that in the class of conditionally conjugate models, with approximating family (2.17), applying the updates (2.21) to each factor in turn will result in a negative energy that is an *affine* function of the tractable statistic functions (c.f. the true posterior which was multi-affine in the same statistics) and is therefore a product of tractable distributions

$$q(\mathbf{x}) \propto \prod_i^M \exp\{\boldsymbol{\tau}_{x_i}(\{\boldsymbol{\mu}_j : j \neq i\})^\top \mathbf{t}_{x_i}(\mathbf{x}_i)\} \tag{2.23}$$

To optimise the variational objective (2.18), we iteratively apply the updates (2.19) to each factor in turn until we converge to a fixed point. Because each update performs a local partial maximisation of (2.18) with respect to the variational parameters of $q_i(\mathbf{x}_i)$, this iterative scheme is equivalent to performing block co-ordinate ascent of the objective. Given that (2.12), and therefore (2.18), is bounded above by the true log marginal likelihood, we are guaranteed

to converge to a local maximum of the objective.

2.4 Stochastic Variational Inference

Stochastic Variational Inference is a technique for performing mean-field variational inference (MFVI) in a way that scales to extremely large datasets. We make a distinction between parameters (global latents) which we denote $\boldsymbol{\theta}$, and local latents which we denote by $\mathcal{X} = \{\mathbf{x}^{(i)} : i = 1, ..., N\}$. We also have local observations $\mathcal{Y} = \{\mathbf{y}^{(i)} : i = 1, ..., N\}$. Our joint distribution over all variables is then given by

$$p(\boldsymbol{\theta}, \mathcal{X}, \mathcal{Y}) = p(\boldsymbol{\theta}) \prod_i^N p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \boldsymbol{\theta}) \tag{2.24}$$

It is a straightforward extension to permit each $\mathbf{x}_i$ to have further structure, but we omit details here to keep notation simple (for further details see [5]). Furthermore, we shall consider the class of exponential family models where the complete conditionals of each of $\boldsymbol{\theta}$, $\mathbf{x}_i$ remain in the same exponential family as in the prior. In particular,

$$p(\boldsymbol{\theta}) = \exp\{\boldsymbol{\eta}_\theta^\top \mathbf{t}_\theta(\boldsymbol{\theta}) - A_\theta(\boldsymbol{\eta}_\theta)\} \tag{2.25}$$

$$p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) = \exp\{\boldsymbol{\eta}_x(\boldsymbol{\theta})^\top \mathbf{t}_x(\mathbf{x}^{(i)}) - A_x(\boldsymbol{\eta}_x(\boldsymbol{\theta}))\} \tag{2.26}$$

$$p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \boldsymbol{\theta}) = \exp\{\boldsymbol{\eta}_y(\boldsymbol{\theta}, \mathbf{x}_i))^\top \mathbf{t}_y(\mathbf{y}^{(i)}) - A_y(\boldsymbol{\eta}_y(\boldsymbol{\theta}, \mathbf{x}^{(i)}))\} \tag{2.27}$$

where $\boldsymbol{\eta}_x(\boldsymbol{\theta})$ is affine in $\mathbf{t}_\theta(\boldsymbol{\theta})$ and $\boldsymbol{\eta}_y(\boldsymbol{\theta}, \mathbf{x}^{(i)})$ is multi-affine in $\mathbf{t}_\theta(\boldsymbol{\theta})$, $\mathbf{t}_x(\mathbf{x}^{(i)})$.

The model described above is a special case of the conditionally conjugate class of models considered in the previous section. The statistic functions $\mathbf{t}_\theta(\boldsymbol{\theta})$ and $\mathbf{t}_x(\mathbf{x}^{(i)})$ individually define tractable exponential families and we therefore introduce an approximate posterior that factors over $\boldsymbol{\theta}$ and $\{\mathbf{x}^{(i)}\}$. Note that we only enforce a factorsation between $\boldsymbol{\theta}$ and $\mathcal{X}$, but due to the conditional independence of observations given $\boldsymbol{\theta}$, this induces a further factorisation over each $\mathbf{x}^{(i)}$. The method of applying MFVI with a factorisation over parameters $\boldsymbol{\theta}$ and latents $\mathcal{X}$ is often referred to as variational bayes (VB). From the results of Section 2.3, our approximating family is then given by

$$\mathcal{Q}_{\text{SVI}} := \left\{ q(\boldsymbol{\theta}, \{\mathbf{x}^{(i)}\}) : q(\boldsymbol{\theta}, \{\mathbf{x}^{(i)}\}) = q(\boldsymbol{\theta} \mid \boldsymbol{\tau}_\theta) \prod_i q(\mathbf{x}^{(i)}|\boldsymbol{\tau}_x^{(i)}) \right\} \tag{2.28}$$

where

$$q(\boldsymbol{\theta} \mid \boldsymbol{\tau}_\theta) = \exp\{\boldsymbol{\tau}_\theta^\top \mathbf{t}_\theta(\boldsymbol{\theta}) - A_\theta(\boldsymbol{\theta})\}$$
$$q(\mathbf{x}^{(i)}|\boldsymbol{\tau}_{x^{(i)}}) = \exp\{\boldsymbol{\tau}_{x^{(i)}}^\top \mathbf{t}_x(\mathbf{x}^{(i)}) - A_x(\boldsymbol{\tau}_{x^{(i)}})\}$$

There is a potential drawback to applying the MFVI updates for this model in the case where the number of conditionally independent observations N is extremely large. In particular, updating the variational parameters for the factor $q(\boldsymbol{\theta})$ requires evaluating a multi-affine function of the expected sufficient statistics of *all* local approximations $q(\mathbf{x}_i) : i = 1, ..., N$. We are also therefore required to store variational parameters for all N local approximations. In order to handle very large datasets, we would like a scheme that allows us to gradually update our variational parameters for $\boldsymbol{\theta}$, incorporating updated estimates from the local approximations as they become available, without having to store the local variational parameters for each observation and with computational cost (per update of $\boldsymbol{\tau}_\theta$) that does not grow with N. This is precisely what motivates the SVI algorithm.

In the discussion to follow we refer to the concept of an *unconstrained local partial optimiser* of a function. Following the definition of [14]

Definition 2.4.1. $\mathbf{y}^*$ *is an unconstrained local partial optimiser of a function* $f(\mathbf{x}, \mathbf{y})$ *given* $\mathbf{x}$ *if there exists an* $\varepsilon > 0$ *such that*

$$f(\mathbf{x}, \mathbf{y}) \leq f(\mathbf{x}, \mathbf{y}^*) \ \forall \mathbf{y} \ \textit{with} \ ||\mathbf{y} - \mathbf{y}^*|| < \varepsilon$$

Note that when there is no ambiguity we shall use *local partial optimiser* as a synonym for unconstrained local partial optimiser in our discussion.

Performing mean-field variational inference on the model (2.24) given observations $\mathcal{Y}$ and approximating family (2.28), is equivalent to maximising the following objective

$$\mathcal{L}(\boldsymbol{\tau}_\theta, \{\boldsymbol{\tau}_{x^{(i)}}\}) := \left\langle \log \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta}|\boldsymbol{\tau}_\theta)} + \sum_i^N \log \frac{p(\mathbf{x}^{(\mathbf{i})}|\boldsymbol{\theta}) p(\mathbf{y}_i|\mathbf{x}^{(\mathbf{i})}, \boldsymbol{\theta})}{q(\mathbf{x}^{(i)}|\boldsymbol{\tau}_x^{(i)})} \right\rangle_{q(\boldsymbol{\theta}|\boldsymbol{\tau}_\theta) q(\mathbf{x}^{(\mathbf{i})}|\boldsymbol{\tau}_{x^{(i)}})} \tag{2.29}$$

For given $\boldsymbol{\tau}_\theta$, finding a local partial optimiser of the objective with respect to a single $\boldsymbol{\tau}_{x^{(i)}}$ is relatively inexpensive. In fact, for the simple case we consider

here where $q(\mathbf{x}^{(i)})$ requires no further factorisation, we have a simple closed form for the update. From (2.19) and dropping terms that do not involve $\mathbf{x}$, our locally optimal factor is given by

$$\begin{aligned} q_i(\mathbf{x}^{(i)}|\boldsymbol{\tau}_{x^{(i)}}) &\propto \exp\left\langle \log p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) + \log p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \boldsymbol{\theta})\right\rangle_{q(\boldsymbol{\theta})} \\ &= \exp\{\boldsymbol{\tau}_x^*\big(\langle \mathbf{t}_\theta(\boldsymbol{\theta})\rangle_{\boldsymbol{\tau}_\theta}, \mathbf{t}_y(\mathbf{y}^{(i)})\big)^\top \mathbf{t}_x(\mathbf{x}^{(i)})\} \end{aligned} \tag{2.30}$$

where $\boldsymbol{\tau}_x(\langle \mathbf{t}_\theta(\boldsymbol{\theta})\rangle_{\boldsymbol{\tau}_\theta}, \mathbf{t}_y(\mathbf{y}^{(i)}))$ is a multi-affine fuction of the mean parameters of $q(\boldsymbol{\theta}|\boldsymbol{\tau}_\theta)$ and the observed statistics vector $\mathbf{t}_y(\mathbf{y}^{(i)})$, which follows from the conjugacy assumptions of our model. For the general case where $q_i(\mathbf{x}_i|\boldsymbol{\tau}_{x_i})$ may require further factorisation, we iteratively apply the mean-field updates within local factors until convergence.

A key insight of [5] is that $\boldsymbol{\tau}_x^*$ is a function of $\boldsymbol{\tau}_\theta$, and so we can rephrase the entire objective as a function of the global variational parameters $\boldsymbol{\tau}_\theta$ only. Let

$$\boldsymbol{\tau}_{x^{(i)}}^*(\boldsymbol{\tau}_\theta) := \boldsymbol{\tau}_x^*\big(\langle \mathbf{t}_\theta(\boldsymbol{\theta})\rangle_{\boldsymbol{\tau}_\theta}, \mathbf{t}_y(\mathbf{y}^{(i)})\big) \tag{2.31}$$

then we have

$$\mathcal{L}_{\text{SVI}}(\boldsymbol{\tau}_\theta) := \mathcal{L}(\boldsymbol{\tau}_\theta, \{\boldsymbol{\tau}_{x^{(i)}}^*(\boldsymbol{\tau}_\theta)\}) \tag{2.32}$$

The SVI approach of [5] is to repeat the following steps

1. Sample i at random from $\{1, ..., N\}$ and compute the local partial optimiser for the datapoint $\boldsymbol{\tau}_{x^{(i)}} = \boldsymbol{\tau}_{x^{(i)}}^*(\boldsymbol{\tau}_\theta)$
2. Compute an intermediate update for $\boldsymbol{\tau}_\theta$, denoted $\hat{\boldsymbol{\tau}}_\theta$, using the MFVI update (2.19), but assuming N repeated observations of datapoint i

$$q(\boldsymbol{\theta}|\hat{\boldsymbol{\tau}}_\theta) \propto \exp\left\langle \log p(\boldsymbol{\theta}) + N \log p(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}|\boldsymbol{\theta})\right\rangle_{\boldsymbol{\tau}_{x^{(i)}}}$$

3. Update $\boldsymbol{\tau}_\theta$ using a weighted average of its previous value and $\hat{\boldsymbol{\tau}}_\theta$

$$\boldsymbol{\tau}_\theta^* = \lambda \boldsymbol{\tau}_\theta + (1 - \lambda)\hat{\boldsymbol{\tau}}_\theta \tag{2.33}$$

Remarkably, [5] demonstrate that this simple procedure is equivalent to taking stochastic, natural gradient steps in the objective (2.32), with step size controlled by the weighting factor λ. Therefore, by following an appropriate

step-size schedule that fulfills the Robbins-Monro conditions [16], we are guaranteed to converge to an optimum of (2.32).

2.5 Amortised Variational Inference

One of the drawbacks of variational inference in the general, non-conjugate case is the requirement to perform a potentially costly optimisation for each datapoint. One avenue for mitigating this cost is to learn a function $\mathbf{r}(\mathbf{y}; \boldsymbol{\phi})$ that maps directly from observation $\mathbf{y}$ to the variational parameters of $q(\mathbf{x})$. This function often takes the form of a neural network and in such cases is commonly referred to as a recognition network. Our variational objective then becomes a function of the parameters of the recognition network, $\boldsymbol{\phi}$

$$\mathcal{L}(\boldsymbol{\phi}) := \left\langle \log \frac{p(\mathbf{x})p(\mathbf{y}|\mathbf{x})}{q(\mathbf{x}|r(\mathbf{y};\boldsymbol{\phi}))} \right\rangle_{q(\mathbf{x}|r(\mathbf{y};\boldsymbol{\phi}))} \tag{2.34}$$

While it may appear that we have not gained much in exchanging one set of parameters $\boldsymbol{\tau}_{\mathbf{x}}$ for another, $\boldsymbol{\phi}$, the crucial difference is that in the case of multiple i.i.d. observations, the parameters $\boldsymbol{\phi}$ are common to *all* observations, whereas the number of variational parameters in the general problem grows with the data size N.

The cost of optimising the parameters $\boldsymbol{\phi}$ is shared by all datapoints, and furthermore, inference on future unseen data has the computational cost of a single function evaluation $r(\mathbf{y}; \boldsymbol{\phi})$. For these reasons this approach is often referred to as *amortised inference*.

A popular approach to amortised inference is the *variational auto-encoder* (VAE) of [4]. In the VAE we use both a neural network recognition model $r(\mathbf{y}; \boldsymbol{\phi})$ as well as a neural network likelihood $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\gamma})$. The recognition and likelihood parameters $\boldsymbol{\phi}$, $\boldsymbol{\gamma}$ are trained jointly using a single stochastic variational objective. In VAE parlance the recognition and likelihood networks are often referred to as the *encoder* and *decoder* respectively.

The target objective for the VAE, for a single data point, is the variational objective $\mathcal{L}_{\text{VAE}}$ below

$$\mathcal{L}_{\text{VAE}}(\boldsymbol{\phi}, \boldsymbol{\gamma}) := \left\langle \log \frac{p(\mathbf{x})p(\mathbf{y}|\mathbf{x}, \boldsymbol{\gamma})}{q(\mathbf{x}|r(\mathbf{y};\boldsymbol{\phi}))} \right\rangle_{q(\mathbf{x}|\mathbf{r}(\mathbf{y};\boldsymbol{\phi}))} \tag{2.35}$$

Typically $p(\mathbf{x})$ is chosen to be a unit variance isotropic Gaussian, with the recognition density being a neural-network parameterised Gaussian density. Here we also choose the observation likelihood to be a neural-network parameterised Gaussian, but we note that it need not mirror the family or structure of the recognition network in general.

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\gamma}) = \mathcal{N}(\mathbf{y} \mid \mu_p(\mathbf{x}, \boldsymbol{\gamma}), \Sigma_p(\mathbf{x}, \boldsymbol{\gamma})) \tag{2.36}$$

$$q(\mathbf{x}|\mathbf{y}, \boldsymbol{\phi}) = \mathcal{N}(\mathbf{x} \mid \mu_q(\mathbf{y}, \boldsymbol{\phi}), \Sigma_q(\mathbf{x}, \boldsymbol{\phi})) \tag{2.37}$$

We can rewrite the objective (2.35) in the following form

$$\mathcal{L}_{\text{VAE}}(\boldsymbol{\phi}, \boldsymbol{\gamma}) = \big\langle \log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\gamma}) \big\rangle_{q(\mathbf{x}|\mathbf{r}(\mathbf{y};\boldsymbol{\phi}))} - \text{KL}[q(\mathbf{x}|r(\mathbf{y};\boldsymbol{\phi}) \mid\mid p(\mathbf{x})] \tag{2.38}$$

In this form we can gain some intuition into what maximising the objective aims to achieve. The first term, often called the reconstruction error, aims to maximise the expected observation density value (or minimise the reconstruction cost) with the expectation taken under our approximate posterior density. The second term, the KL divergence, acts as a regularising term working to encourage our approximate posterior to be, on average, similar to the prior.

If we aim to maximise (2.35) using gradient-based optimisation, we would like to to compute the gradient

$$\nabla\mathcal{L}_{\text{VAE}}(\boldsymbol{\phi}, \boldsymbol{\gamma}) = \nabla\big\langle \log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\gamma}) \big\rangle_{q(\mathbf{x}|r(\mathbf{y};\boldsymbol{\phi}))} - \nabla\text{KL}[q(\mathbf{x}|\mathbf{r}(\mathbf{y};\boldsymbol{\phi}) \mid\mid p(\mathbf{x})] \tag{2.39}$$

Provided $p(\mathbf{x})$ is a tractable exponential family and we choose our approximation $q(\mathbf{x}|r(\mathbf{y};\boldsymbol{\phi}))$ to be within the same family, the KL term (and therefore its gradients) can easily be computed (see Theorem 2.1.2). The first term however involves computing an integral that is typically intractable.

We may be tempted to sample the integrand and compute the gradient at the sampled value, but unfortunately this results in biased estimates as the density from which we need to sample itself depends on the parameters we wish to take gradients with respect to. The VAE approach uses the so called *reparameterisation trick*, that is, let $\mathbf{x} = \mathbf{f}(\boldsymbol{\varepsilon}; r(\mathbf{y};\boldsymbol{\phi}))$ be a differentiable transformation of an auxilliary random variable $\boldsymbol{\varepsilon} \sim p(\boldsymbol{\varepsilon})$, so that

$$\nabla\big\langle \log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\gamma}) \big\rangle_{q(\mathbf{x}|r(\mathbf{y};\boldsymbol{\phi}))} = \big\langle \nabla \log p(\mathbf{y}|\mathbf{f}(\boldsymbol{\varepsilon}; r(\mathbf{y};\boldsymbol{\phi}))), \boldsymbol{\gamma}) \big\rangle_{p(\boldsymbol{\varepsilon})} \tag{2.40}$$

We can now sample the expectation under $p(\boldsymbol{\varepsilon})$ to get unbiased estimates of

(2.40) and therefore unbiased estimates of the gradient of (2.38).

We also note that, while the exact KL term has a convenient analytic expression in the case mentioned above, it has been shown that a lower variance gradient estimator can often be obtained through a different decomposition [17]. The gradient of the terms inside the expectation of (2.35) can be decomposed into *path derivative* and *score function* terms, where the latter has zero expectation. By omitting the second term in our monte-carlo gradient estimate, we can often obtain an estimate with lower variance than that obtained by computing the KL exactly, with the additional desirable property that the gradient variance goes to zero as $q\big(\mathbf{x}|\mathbf{r}(\mathbf{y};\boldsymbol{\phi})\big)$ approaches $p(\mathbf{x}|\mathbf{y},\boldsymbol{\gamma})$.

An intuition behind the VAE is that it learns to model the data as having a number of underlying latent causes, represented by continuous random variables. The assumption of a chosen, simple form for the prior (isotropic Gaussian in our example) is not as restrictive as it may seem, as given a sufficiently powerful network for the observation likelihood, the VAE can warp the latent prior mass so that the observed data density takes on almost arbitrary form.

However, there are several potential shortcomings of the VAE. First, the reparameterisation trick restricts us to using latents $\mathbf{x}$ that can be expressed as a differentiable transformation of an auxilliary variable $\boldsymbol{\varepsilon}$ which precludes the use of discrete latents. Second, while the VAE may be able to approximate marginal data densities arbitrarily well given a sufficiently powerful generative network, we may be interested in the underlying causes of the data. With the standard VAE it may be unlikely that we will recover interpretable representations in the latent variables. Further work has attempted to address this issue however by encouraging disentangled representations, aiming to recover independent latent factors of variation in the generative process [18].

Finally, it may often be the case that we have prior assumptions about the structure of our latent model. With the VAE, the recognition network is unable to take advantage of these explicit assumptions during training. In particular, the VAE is required to produce the full posterior as the output of its recognition network, and while it may be possible for the network to take advantage of any special structure in the latent variables, it must learn to do so exclusively from data, in addition to learning the nonlinear mapping from latents to observations.

2.6 Expectation Propagation

An alternative approach to performing approximate inference in graphical models is the *expectation propagation* algorithm of Minka [2] [3]. Expectation propagation (EP) targets the *inclusive* KL divergence, $\mathrm{KL}[p \,||\, q]$, so called because it favours approximations q that *include* regions of low/zero density in p, as opposed to the zero-avoiding behaviour of the *exclusive* KL divergence targeted by variational approximations, $\mathrm{KL}[q \,||\, p]$.

Given a (typically un-normalised) distribution $p(\mathbf{x})$, we wish to find an approximate distribution $q(\mathbf{x}) \in \mathcal{Q}$, where $\mathcal{Q}$ is some tractable exponential family, that minimises

$$\mathrm{KL}[p(\mathbf{x}) \,||\, q(\mathbf{x})] := \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x} \tag{2.41}$$

However, even having restricted $q(\mathbf{x})$ to lie in a tractable family, performing this minimisation is typically intractable. Indeed, performing this minimisation would require computing expectations under $p(\mathbf{x})$, which we assume to be intractable.

Expectation propagation instead performs a series of local KL minimisations. Let $p(\mathbf{x})$ be a product of K factors

$$p(\mathbf{x}) = \prod_{i=1}^{K} f_i(\mathbf{x}) \tag{2.42}$$

We use an approximating distribution $q(\mathbf{x})$, in a tractable exponential family $\mathcal{Q}$ with sufficient statistic vector $\mathbf{t}_x(\mathbf{x})$, which we also decompose into a product of K factors

$$q(\mathbf{x}) = \prod_{i=1}^{K} \tilde{f}_i(\mathbf{x}|\boldsymbol{\tau}_i) \tag{2.43}$$

For each iteration we choose $i \in \{1, ..., K\}$ and update our approximation to the i-th factor, $\tilde{f}_i(\mathbf{x})$, by minimising the inclusive KL in the context of the rest of our approximation. That is, we find

$$\boldsymbol{\tau}_i = \arg\min_{\boldsymbol{\tau}_i} \mathrm{KL}[f_i(\mathbf{x}) q_{\setminus i}(\mathbf{x}) \,||\, \tilde{f}_i(\mathbf{x}|\boldsymbol{\tau}_i) q_{\setminus i}(\mathbf{x})] \tag{2.44}$$

where

$$q_{\setminus i}(\mathbf{x}) = \prod_{j \neq i} \tilde{f}_j(\mathbf{x}|\boldsymbol{\tau}_j) \tag{2.45}$$

Note that up to this point the product $f_i(\mathbf{x})q_{\setminus i}(\mathbf{x})$ need not be normalised. We now introduce the (normalised) *tilted distribution*

$$\begin{aligned} \hat{p}_i(\mathbf{x}) &\propto f_i(\mathbf{x})q_{\setminus i}(\mathbf{x}) \\ &\propto f_i(\mathbf{x}) \exp\{\boldsymbol{\tau}_{\setminus i}^\top \mathbf{t}_x(\mathbf{x})\} \end{aligned} \tag{2.46}$$

The solution to (2.44) is found by moment matching (See appendix A.2 for a proof). First, we find the natural parameters $\hat{\boldsymbol{\tau}}_i$ of distribution $\hat{q}_i(\mathbf{x}|\hat{\boldsymbol{\tau}}_i) \in \mathcal{Q}$, such that

$$\langle \mathbf{t}_x(\mathbf{x}) \rangle_{\hat{q}_i} = \langle \mathbf{t}_x(\mathbf{x}) \rangle_{\hat{p}_i} \tag{2.47}$$

This is known as the projection step of EP, because we are projecting from the tilted distribution family into our approximating family $\mathcal{Q}$. Our updated factor $\tilde{f}_i(\mathbf{x}|\boldsymbol{\tau}_i)$ is then given by

$$\begin{aligned} \tilde{f}_i(\mathbf{x}|\boldsymbol{\tau}_i) &= \frac{\hat{q}_i(\mathbf{x} \mid \hat{\boldsymbol{\tau}}_i)}{q_{\setminus i}(\mathbf{x})} \\ &\propto \exp\{\mathbf{t}_x(\mathbf{x})^\top (\hat{\boldsymbol{\tau}}_i - \boldsymbol{\tau}_{\setminus i})\} \\ \therefore \boldsymbol{\tau}_i &= \hat{\boldsymbol{\tau}}_i - \boldsymbol{\tau}_{\setminus i} \end{aligned} \tag{2.48}$$

where $\boldsymbol{\tau}_{\setminus i}$ are the natural parameters of $q_{\setminus i}(\mathbf{x})$.

An interesting observation [19] is that the distributions $\hat{q}_i(\mathbf{x}|\hat{\boldsymbol{\tau}}_i)$ and $\hat{p}_i(\mathbf{x})$ each lie in an exponential family with statistic vector $\mathbf{t}_x(\mathbf{x})$, but with base measures that differ by a factor $f_i(\mathbf{x})$. The projection step then amounts to performing the forward mapping (computing moments) under the tilted base measure $f_i(\mathbf{x})\nu_{\mathcal{Q}}(\mathbf{x})$, followed by the reverse mapping under our approximating family base measure, $\nu_{\mathcal{Q}}(\mathbf{x})$.

We iterate the moment matching updates (2.47), (2.48) for each factor in turn until convergence. However, unlike the mean-field updates of section 2.3, the EP updates are only guaranteed to converge in a handful of special cases [20] [21].

We state here a useful result that permits simpler, local updates in many cases. See Appendix A.3 for a proof.

Theorem 2.6.1. *Let $p(\mathbf{x}) = \prod_i f_i(\mathbf{x}_i)$ be a product over K cliques $\{\mathbf{x}_1, ..., \mathbf{x}_K\}$. Furthermore, let the exponential family $\mathcal{Q}$ be closed under $\mathbf{x}_i$ marginalisation, and let $\mathcal{Q}_i \subset \mathcal{Q}$ be the subset of $\mathcal{Q}$ containing distributions over $\mathbf{x}_i$ only. Then,*

$$\underset{\tilde{f}_i(\mathbf{x})\in\mathcal{Q}}{\arg\min}\, KL[f_i(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}) \,||\, \tilde{f}_i(\mathbf{x})q_{\setminus i}(\mathbf{x})] = \underset{\tilde{f}_i(\mathbf{x}_i)\in\mathcal{Q}_i}{\arg\min}\, KL[f_i(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}_i) \,||\, \tilde{f}_i(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}_i)]$$

where

$$q_{\setminus i}(\mathbf{x}_i) := \int q_{\setminus i}(\mathbf{x}) \mathrm{d}\mathbf{x}_{\setminus i} \tag{2.49}$$

We refer to (2.49) as the *cavity distribution* for factor f_i. Theorem 2.6.1 is useful as it allows is to perform localised updates over smaller sets of variables. Note that this is a special case as in general exponential families are not closed under marginalisation, but notable exceptions are Gaussian or fully-factorised families.

Another useful simplification occurs when a factor $f_i(\mathbf{x})$, and also therefore $\hat{p}_i(\mathbf{x})$, already lies within our approximating family. In this case, if we initialise the approximation to the true factor so that $\hat{f}_i(\mathbf{x}|\boldsymbol{\tau}_i) = f_i(\mathbf{x})$, further updates to the factor have no effect.

While we have described the application of EP for the purpose of finding a tractable, normalised approximation to $p(\mathbf{x})$, often the normaliser is itself of interest. EP can also be used to compute an approximate normaliser $\tilde{Z}$ by performing local minimisations of the *unnormalised KL divergence*

$$\overline{\mathrm{KL}}[p||q] := \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x} + \int \big(q(\mathbf{x}) - p(\mathbf{x})\big) \mathrm{d}\mathbf{x} \tag{2.50}$$

Minimising (2.50) amounts to moment matching as before, but additionally multiplying our approximation by a scale factor so that zeroth moments match. Following [8] we let $\tilde{f}_i(\mathbf{x})$ be any approximation to the factor $f_i(\mathbf{x})$ so that our full (unscaled) approximating distribution is $q(\mathbf{x}) = \prod_i \tilde{f}_i(\mathbf{x})$. We can minimise the local un-normalised KL divergence (2.50) for factor $\tilde{f}_i$ with respect to a scale

parameter s_i so that

$$s_i = \arg\min_s \overline{\mathrm{KL}}[f_i(\mathbf{x})q_{\setminus i}(\mathbf{x})||s\tilde{f}_i(\mathbf{x})q_{\setminus i}(\mathbf{x})] \tag{2.51}$$

Taking derivatives and equating to zero, we find

$$\begin{aligned}\frac{\partial}{\partial s}\overline{\mathrm{KL}}[f_i(\mathbf{x})q_{\setminus i}(\mathbf{x})||s\tilde{f}_i(\mathbf{x})q_{\setminus i}(\mathbf{x})] &= -\int \frac{f_i(\mathbf{x})q_{\setminus i}(\mathbf{x})}{s}\mathrm{d}\mathbf{x} + \int \tilde{f}_i(\mathbf{x})q_{\setminus i}(\mathbf{x})\mathrm{d}\mathbf{x} \\ \therefore s_i &= \frac{\int f_i(\mathbf{x})q_{\setminus i}(\mathbf{x})\mathrm{d}\mathbf{x}}{\int \tilde{f}_i(\mathbf{x})q_{\setminus i}(\mathbf{x})\mathrm{d}\mathbf{x}}\end{aligned} \tag{2.52}$$

Our approximate normaliser $\tilde{Z}$ is then given by

$$\begin{aligned}\tilde{Z}(\{\boldsymbol{\tau}_i\}) &:= \left(\int q(\mathbf{x})\mathrm{d}\mathbf{x}\right)\prod_{i=1}^{K} s_i \\ &= \left(\int q(\mathbf{x})\mathrm{d}\mathbf{x}\right)\prod_{i=1}^{K}\frac{\int f_i(\mathbf{x})q_{\setminus i}(\mathbf{x})\mathrm{d}\mathbf{x}}{\int \tilde{f}_i(\mathbf{x}_i|\boldsymbol{\tau}_i)q_{\setminus i}(\mathbf{x})\mathrm{d}\mathbf{x}} \\ &= \left(\int q(\mathbf{x})\mathrm{d}\mathbf{x}\right)\prod_{i=1}^{K}\frac{\int f_i(\mathbf{x})q_{\setminus i}(\mathbf{x})\mathrm{d}\mathbf{x}}{\int q(\mathbf{x})\mathrm{d}\mathbf{x}} \\ &= \left(\int q(\mathbf{x})\mathrm{d}\mathbf{x}\right)^{1-K}\prod_i\left(\int f_i(\mathbf{x})q_{\setminus i}(\mathbf{x})\mathrm{d}\mathbf{x}\right)\end{aligned} \tag{2.53}$$

where we have made explicit the dependence of $\tilde{Z}$ on the natural parameters of each site approximation $\tilde{f}_i(\mathbf{x}|\boldsymbol{\tau}_i)$.

It can be shown (see Appendix A.4) that the stationary points of $\tilde{Z}$ are in one-to-one correspondence with the fixed points of EP, and so (2.53) can therefore be viewed as an energy function for EP [8].

Chapter 3

Structured Variational Auto-Encoders

In this chapter we describe the structured variational auto-encoder (SVAE) of [1]. We begin with an overview, aiming to provide some high level intuition and motivation for the approach. We then explore the objective functions employed by the SVAE, followed by a discussion of the optimisation schemes used to target those objectives.

3.1 Overview

The SVAE approach of [1] can be seen as an generalisation of SVI (Section 2.4) to cases where we have general, non-conjugate observation likelihoods. In particular, we are interested in the class of models with factorisation

$$p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})p(\mathbf{y}|\mathbf{x},\boldsymbol{\gamma}) \tag{3.1}$$

where the prior over global latents $p(\boldsymbol{\theta})$ is conjugate to $p(\mathbf{x}|\boldsymbol{\theta})$, but $p(\mathbf{y}|\mathbf{x},\boldsymbol{\gamma})$ can take any form and in general does not remain conjugate to the prior on $\mathbf{x}$. For our purposes we shall assume that the observation likelihoods are parameterised by neural networks.

The SVI approach of [5] has several desirable properties that we would like to replicate for this more general class of models

1. A single variational objective that maximises a lower bound on the marginal

likelihood / log model evidence

2. Guaranteed convergence to an optimum of the variational objective

3. Stochastic natural gradient steps in objective with computational and memory cost that does not grow with the number of data points N

4. Avoiding comparatively slow gradient-based optimisation of local variational parameters by exploiting assumed graphical model structure

It turns out that we can recover all of these properties in the non-conjugate case, albeit by introducing further approximations. The SVI approach chooses the local variational parameters to be a local partial optimiser (see Definition 2.4.1) of the true objective given $\boldsymbol{\tau}_\theta$, and in doing so 'optimises away' these parameters, leaving the objective as a function of the global variational parameters only. The additional approximation introduced by the SVAE is to instead choose the local variational parameters to be a local partial optimiser of a *surrogate* objective function which is easier to optimise. The surrogate objective is parameterised by the output of a recognition network, which is trained jointly with the global variational and likelihood parameters using a single stochastic objective.

A complimentary view of the SVAE is that of a generalisation of the variational auto-encoder of Section 2.5. Whereas the VAE utilises a recognition network to parameterise the approximate posterior $q(\mathbf{x})$ directly, the SVAE produces recognition network potentials that are a subset of the factors in a larger structured graphical model. As discussed previously, a drawback of the VAE approach is that for models in which we assume some structure in the latent variables, the recognition network is effectively required to learn to do inference in a structured model while also learning the non-linear mapping from latents to observations. The SVAE instead separates out these two effects, utilising fast exponential family inference routines that exploit our prior assumptions about latent variable structure, while using the flexibility of neural networks solely for learning the nonlinear mapping between latents and observations.

3.2 Objective Function(s)

Given the model (3.1), our aim is to maximise a variational lower bound on the model evidence. Using the notation introduced here, this objective is given by

$$\mathcal{L}[q(\boldsymbol{\theta}, \mathbf{x})] := \left\langle \log \frac{p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})p(\mathbf{y}|\mathbf{x}, \boldsymbol{\gamma})}{q(\boldsymbol{\theta}, \mathbf{x})} \right\rangle_{q(\boldsymbol{\theta}, \mathbf{x})} \tag{3.2}$$

The optimum value of this objective is found with $q(\boldsymbol{\theta}, \mathbf{x}) = p(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$, however, by assumption this posterior is intractable. Given the conjugacy between $p(\boldsymbol{\theta})$ and $p(\mathbf{x}|\boldsymbol{\theta})$, with assumed tractable statistic functions $\mathbf{t}_\theta(\boldsymbol{\theta})$, $\mathbf{t}_x(\mathbf{x})$, a natural approach would be to use a factorised approximate posterior $q(\boldsymbol{\theta}, \mathbf{x}) = q(\boldsymbol{\theta})q(\mathbf{x})$, as was the case with conditionally conjugate MFVI and SVI. Our constrained optimisation problem then becomes

$$\mathcal{L}[q(\boldsymbol{\theta}), q(\mathbf{x})] := \left\langle \log \frac{p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})p(\mathbf{y}|\mathbf{x}, \boldsymbol{\gamma})}{q(\boldsymbol{\theta})q(\mathbf{x})} \right\rangle_{q(\boldsymbol{\theta})q(\mathbf{x})} \tag{3.3}$$

In the case of SVI, or conjugate MFVI more generally, any optimum of the factorised objective (3.3) but with $q(\boldsymbol{\theta})$, $q(\mathbf{x})$ otherwise unconstrained, results in an approximation that is a product of the exponential families with statistic functions $\mathbf{t}_x(\mathbf{x})$ and $\mathbf{t}_\theta(\boldsymbol{\theta})$.

In our more general case however, the lack of conjugate observation likelihoods means that the optimal factorised approximation will not in general be in any tractable exponential family. We therefore impose the additional constraint that $q(\boldsymbol{\theta})$, $q(\mathbf{x})$ are in the exponential families with statistic functions $\mathbf{t}_\theta(\boldsymbol{\theta})$, $\mathbf{t}_x(\mathbf{x})$ respectively, and so our objective becomes

$$\mathcal{L}(\boldsymbol{\tau}_\theta, \boldsymbol{\tau}_x) := \left\langle \log \frac{p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})p(\mathbf{y}|\mathbf{x}, \boldsymbol{\gamma})}{q(\boldsymbol{\theta} \mid \boldsymbol{\tau}_\theta)q(\mathbf{x} \mid \boldsymbol{\tau}_x)} \right\rangle_{q(\boldsymbol{\theta}|\boldsymbol{\tau}_\theta)q(\mathbf{x}|\boldsymbol{\tau}_x)} \tag{3.4}$$

where

$$q(\boldsymbol{\theta} \mid \boldsymbol{\tau}_\theta) = \exp\{\boldsymbol{\tau}_\theta^\top \mathbf{t}_\theta(\boldsymbol{\theta}) - A_\theta(\boldsymbol{\tau}_\theta)\}$$
$$q(\mathbf{x} \mid \boldsymbol{\tau}_x) = \exp\{\boldsymbol{\tau}_x^\top \mathbf{t}_x(\mathbf{x}) - A_x(\boldsymbol{\tau}_x)\}$$

In the SVI approach of [5], the objective is rephrased as a function of the global variational parameters $\boldsymbol{\tau}_\theta$ only, by choosing the local variational parameters $\boldsymbol{\tau}_x$ to be a function of the global variational parameters, i.e. $\boldsymbol{\tau}_x = \boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta)$

$$\mathcal{L}_{\text{SVI}}(\boldsymbol{\tau}_\theta, \boldsymbol{\tau}_x) := \mathcal{L}(\boldsymbol{\tau}_\theta, \boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta)) \tag{3.5}$$

In conjugate SVI, the function $\boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta)$ returns local variational parameters $\boldsymbol{\tau}_x$ that are a local partial optimiser of the objective (3.4), given $\boldsymbol{\tau}_\theta$. However, in the non-conjugate setting we consider here, finding a local partial optimiser of this true objective is not something we can do efficiently and would typically require resorting to relatively slow gradient-based optimisation techniques.

Instead, the SVAE approach chooses $\boldsymbol{\tau}_x$ to be a local partial optimiser of a *surrogate* objective function $\hat{\mathcal{L}}(\boldsymbol{\tau}_\theta, \boldsymbol{\tau}_x, \boldsymbol{\phi})$, which serves as an approximation to the true objective, but one which by construction can be optimised efficiently. The surrogate objective depends not only on the variational parameters $\boldsymbol{\tau}_\theta$ and $\boldsymbol{\tau}_x$, but also on $\boldsymbol{\phi}$, the parameters of a recognition network.

The surrogate objective $\hat{\mathcal{L}}$ is the mean-field variational objective of a surrogate graph, where non-conjugate likelihood terms $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\gamma})$ are replaced by conjugate potentials $\exp\{\mathbf{t}_x(\mathbf{x})^\top \mathbf{r}(\mathbf{y}; \boldsymbol{\phi})\}$, where $\mathbf{r}(\mathbf{y}; \boldsymbol{\phi})$ is the output of a recognition network.

$$\hat{\mathcal{L}}(\boldsymbol{\tau}_\theta, \boldsymbol{\tau}_x, \boldsymbol{\phi}) := \left\langle \log \frac{p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta}) \exp\{\mathbf{t}_x(\mathbf{x})^\top \mathbf{r}(\mathbf{y}; \boldsymbol{\phi})\}}{q(\boldsymbol{\theta} \mid \boldsymbol{\tau}_\theta) q(\mathbf{x} \mid \boldsymbol{\tau}_x)} \right\rangle_{q(\boldsymbol{\theta}|\boldsymbol{\tau}_\theta) q(\mathbf{x}|\boldsymbol{\tau}_x)} \tag{3.6}$$

Our local partial optimiser $\boldsymbol{\tau}_x^*$ is then a function of $\boldsymbol{\tau}_\theta, \boldsymbol{\phi}$

$$\boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi}) := \arg\max_{\boldsymbol{\tau}_x} \hat{\mathcal{L}}(\boldsymbol{\tau}_\theta, \boldsymbol{\tau}_x, \boldsymbol{\phi}) \tag{3.7}$$

where here we take arg max to be the argument of *any* maximum, not necessarily the global maximum. As the surrogate objective (3.6) by construction has conjugacy structure throughout, we can maximise (3.7) efficiently using conjugate MFVI block co-ordinate ascent updates (2.21). In the simple case that $q(\mathbf{x})$ is a single factor this can be performed in a single step, but typically $q(\mathbf{x})$ will have further structure, in which case we iterate updates of local factors until convergence.

Given this local partial optimiser $\boldsymbol{\tau}_x^*$ of our surrogate objective, we can now rephrase our objective as a function of the global variational parameters $\boldsymbol{\tau}_\theta$, the likelihood parameters $\boldsymbol{\gamma}$, and additionally now our recognition network

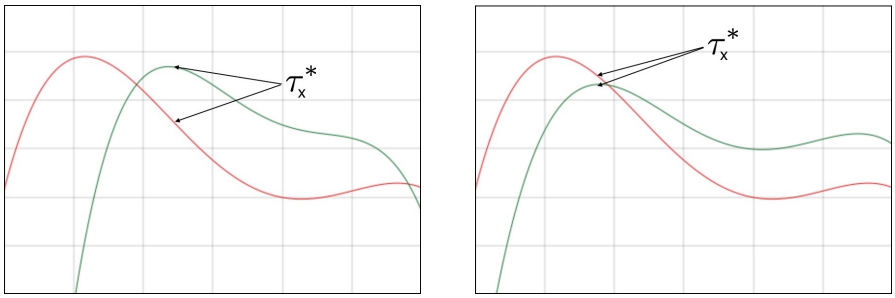

Figure 3.1: Local partial optimiser $\boldsymbol{\tau}_x^*$ of surrogate objective $\hat{\mathcal{L}}$ (green) before (left) and after (right) gradient step in true objective $\mathcal{L}_{\text{SVAE}}$ (red). x-axis: variational parameter value, y-axis: objective function value(s).

parameters $\boldsymbol{\phi}$.

$$\mathcal{L}_{\text{SVAE}}(\boldsymbol{\tau}_\theta, \boldsymbol{\gamma}, \boldsymbol{\phi}) := \mathcal{L}(\boldsymbol{\tau}_\theta, \boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi}), \boldsymbol{\gamma}) \tag{3.8}$$

Note that here we have made explicit the dependence on $\boldsymbol{\gamma}$ as we would like to learn this jointly with the variational and recognition network parameters. (3.8) is the global objective targeted by the SVAE approach.

It is worth taking a moment to consider the role played by the surrogate objective (3.6) and recognition network parameters $\boldsymbol{\phi}$. Figure 3.1 presents an idealised illustration of the role of the surrogate objective. We find a local partial optimiser $\boldsymbol{\tau}_x^*$ of the surrogate objective $\hat{\mathcal{L}}$ (green), which in general does not correspond to an optimum of the true objective $\mathcal{L}_{\text{SVAE}}$ (red). However, when we take a gradient step in $\mathcal{L}_{\text{SVAE}}$, which is a function of $\boldsymbol{\phi}$ due to the indirect dependence through $\boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$, this should act to pull the local partial optimiser of $\hat{\mathcal{L}}$ to correspond to a higher value of $\mathcal{L}_{\text{SVAE}}$. Training the recognition network then serves to produce a surrogate objective that is an approximation to the true objective, at least to the extent that their optima coincide. The left and right images of Figure 3.1 present an illustration of the two objectives before and after a gradient step in $\boldsymbol{\phi}$ respectively.

The SVAE objective (3.8) lower-bounds the partially optimised mean-field objective in the following sense (Proposition D.2 of [1])

$$\mathcal{L}_{\text{SVAE}}(\boldsymbol{\tau}_\theta, \boldsymbol{\gamma}, \boldsymbol{\phi}) \leq \max_{\boldsymbol{\tau}_x} \mathcal{L}(\boldsymbol{\tau}_\theta, \boldsymbol{\tau}_x) \leq \max_{q(\mathbf{x})} \mathcal{L}[q(\boldsymbol{\theta})q(\mathbf{x})] \tag{3.9}$$

Furthermore, because we know that the mean-field objective is bounded above by the log model evidence, we know that the SVAE objective is bounded above. As noted by [1], if there is some $\boldsymbol{\phi}^*$ such that $\mathbf{t}_x(\mathbf{x})^\top \mathbf{r}(\mathbf{y}; \boldsymbol{\phi}^*) = \log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\gamma})$, then the bound can be made tight in the sense that

$$\mathcal{L}_{\text{SVAE}}(\boldsymbol{\tau}_\theta, \boldsymbol{\gamma}, \boldsymbol{\phi}) = \max_{\boldsymbol{\tau}_x} \mathcal{L}(\boldsymbol{\tau}_\theta, \boldsymbol{\tau}_x) = \max_{q(\mathbf{x})} \mathcal{L}[q(\boldsymbol{\theta}), q(\mathbf{x})] \tag{3.10}$$

See Proposition D.2 of [1] for a proof. In general this will not be attainable, regardless of the flexibility of the recognition network. Indeed if this were possible, it would imply that $\log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\gamma})$ is affine in the sufficient statistics $\mathbf{t}_x(\mathbf{x})$, which is contrary to our initial assumption of non-conjugacy. It does however provide some intuition as to what the recognition network is trying to approximate.

Whereas conjugate SVI is guaranteed to converge to an optimum of the mean-field variational objective (2.29), the SVAE is only guaranteed convergence to an optimum of $\mathcal{L}_{\text{SVAE}}$.

3.3 Optimisation

We now turn our attention to optimisation of the objective (3.8). First, we note that the objective depends on the parameters $\boldsymbol{\tau}_\theta$, $\boldsymbol{\phi}$ through the function $\boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$ (as well as directly in the case of $\boldsymbol{\tau}_\theta$). To use gradient based optimisation techniques, we therefore require that the function $\boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$ be differentiable. This function returns the result of a sequence of optimisation iterations (conjugate MFVI updates), which may not intuitively seem like something we can differentiate. However, the mean-field optimisation can be viewed as an infinite recursion of a vector-valued function Φ that converges to an attractive fixed point

$$\begin{aligned} \boldsymbol{\tau}_x^{(n)} &= \Phi(\boldsymbol{\tau}_x^{(n-1)}, \boldsymbol{\tau}_\theta, \boldsymbol{\phi}) \\ \boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi}) &= \lim_{n \to \infty} \boldsymbol{\tau}_x^{(n)} \end{aligned} \tag{3.11}$$

Where $\boldsymbol{\tau}_x^{(0)}$ are our our initial variational parameters. Under certain differentiability conditions on Φ (see Theorem 2.2 of [22]), the fixed point $\boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$ is continuously differentiable with respect to its arguments and, in particular, its gradient is a function of the derivatives of Φ evaluated at the fixed point $\boldsymbol{\tau}_x^*$. This gradient can typically be evaluated far more efficiently than the forward

pass required to compute the fixed point. See [1] [14] [22] for further details.

Assuming that the differentiability conditions on Φ are met (we make this assumption without offering a proof) we can maximise objective 3.8 using standard gradient-based optimisation techniques. However, for the global variational parameters $\boldsymbol{\tau}_\theta$ it turns out that we are able to compute a stochastic estimate of the natural gradient as or more efficiently than the standard gradient, depending on the factorisation of $q(\mathbf{x})$.

The *natural gradient* [23] modifies the standard gradient to adjust for the information geometry of its parameter space. While the standard gradient gives the direction of steepest ascent within an infinitesimaly small radius, with the radius defined using euclidian distance, this distance metric is typically a poor measure of the dis-similarity between a pair of distributions. The natural gradient is also defined as the direction of steepest ascent within an infinitesimally small radius, but using the symmetrised KL divergence ($\mathrm{KL}[p, p'] + \mathrm{KL}[p', p]$) as the distance metric [5]. It can be shown [23] that the natural gradient of a function $f(\boldsymbol{\theta})$ with respect to a distribution $p(.|\boldsymbol{\theta})$ is given by

$$\tilde{\nabla} f(\boldsymbol{\theta}) := G^{-1}(\boldsymbol{\theta}) \nabla f(\boldsymbol{\theta}) \tag{3.12}$$

where G is the fisher information matrix of $p(.|\boldsymbol{\theta})$. When $p(.|\boldsymbol{\eta})$ is an exponential family with natural parameters $\boldsymbol{\eta}$ we have the following identity (see [5] for a proof)

$$\tilde{\nabla} f(\boldsymbol{\eta}) = [\nabla^2 A(\boldsymbol{\eta})]^{-1}(\boldsymbol{\eta}) \nabla f(\boldsymbol{\eta}) \tag{3.13}$$

In the case where the local latent factor $q(\mathbf{x})$ has no additional factorisation structure, we have

$$\tilde{\nabla}_{\boldsymbol{\tau}_\theta} \mathcal{L}_{\mathrm{SVAE}}(\boldsymbol{\tau}_\theta, \boldsymbol{\gamma}, \boldsymbol{\phi}) = (\boldsymbol{\eta}_\theta + \langle \mathbf{t}'_x(\mathbf{x}) \rangle_{q(\mathbf{x}|\boldsymbol{\tau}^*_x)} - \boldsymbol{\tau}_\theta) + (\nabla_{\boldsymbol{\tau}_x} \mathcal{L}(\boldsymbol{\tau}_\theta, \boldsymbol{\gamma}, \boldsymbol{\tau}^*_x(\boldsymbol{\tau}_\theta, \boldsymbol{\phi}), 0) \tag{3.14}$$

where we use $\tilde{\nabla}_{\boldsymbol{\tau}_\theta}$ to denote the natural gradient with respect to $\boldsymbol{\tau}_\theta$ and $\boldsymbol{\eta}_\theta$ are the natural parameters for the prior $p(\boldsymbol{\theta}|\boldsymbol{\eta}_\theta)$ (see Proposition D.3 of [1] for a proof). Note also that $\mathbf{t}'_x(\mathbf{x})$ is the augmented sufficient statistic vector as defined in Section 2.1. A stochastic estimate (see below) of the second term is computed as part of the backward pass for the gradient with respect to $\boldsymbol{\phi}$, and so if we store the expected statistics $\langle \mathbf{t}'_x(\mathbf{x}) \rangle_{q(\mathbf{x}|\boldsymbol{\tau}^*_x)}$ during the computation of $\boldsymbol{\tau}^*_x(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$, we get the natural gradient with respect to $\boldsymbol{\tau}_\theta$ for the additional cost

of just a handful of vector addition / subtractions.

In the more general case, where $q(\mathbf{x})$ contains additional factorisation structure, we have [1]

$$\tilde{\nabla}_{\boldsymbol{\tau}_\theta}\mathcal{L}_{\text{SVAE}}(\boldsymbol{\tau}_\theta, \boldsymbol{\gamma}, \boldsymbol{\phi}) = (\boldsymbol{\eta}_\theta + \langle t_x^*(\mathbf{x})\rangle_{q(\mathbf{x}|\boldsymbol{\tau}_x^*)} - \boldsymbol{\tau}_\theta) + (\nabla^2 A_\theta(\boldsymbol{\tau}_\theta))^{-1}\nabla[\boldsymbol{\tau}'_\theta \mapsto \mathcal{L}(\boldsymbol{\tau}_\theta, \boldsymbol{\gamma}, \boldsymbol{\tau}_x^*(\boldsymbol{\tau}'_\theta, \boldsymbol{\phi}))] \tag{3.15}$$

(see [1] for a proof). If we reparameterise $\boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$ to be a function of the mean parameters, so that $\tilde{\boldsymbol{\tau}}_x^*(\mu, \boldsymbol{\phi}) := \boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta(\mu), \boldsymbol{\phi})$, then by applying Theorem 2.1.3 we can compute the second term of (3.15) without explicitly inverting the Fisher information $\nabla^2 A_\theta(\boldsymbol{\tau}_\theta)$.

$$\tilde{\nabla}_{\boldsymbol{\tau}_\theta}\mathcal{L}_{\text{SVAE}}(\boldsymbol{\tau}_\theta, \boldsymbol{\gamma}, \boldsymbol{\phi}) = (\boldsymbol{\eta}_\theta + \langle t_x^*(\mathbf{x})\rangle_{q(\mathbf{x}|\boldsymbol{\tau}_x^*)} - \boldsymbol{\tau}_\theta) + \nabla_{\mu_{\boldsymbol{\tau}_\theta}}\mathcal{L}(\boldsymbol{\tau}_\theta, \boldsymbol{\gamma}, \tilde{\boldsymbol{\tau}}_x^*(\mu_{\boldsymbol{\tau}_\theta}, \boldsymbol{\phi})) \tag{3.16}$$

While the definition above strictly only applies to minimal families, an analogous result for overcomplete families follows from Theorem 2.1.4.

The gradient of (3.4) with respect to τ_x involves taking the gradient of an intractable expectation. We therefore employ the use of the reparameterisation trick as described in Section 2.5 in order to obtain a stochastic gradient estimate. This gradient estimate is required as part of the backward pass for τ_θ and ϕ.

By also sampling a batch of observations on which to compute the desired gradients, we can obtain stochastic gradient estimates with cost that does not grow with N. We can therefore optimise the objective (3.8) using stochastic natural gradient steps in $\boldsymbol{\tau}_\theta$ and standard gradient steps in $\boldsymbol{\gamma}$, $\boldsymbol{\theta}$. Taken jointly this amounts to a preconditioned stochastic gradient step in *all* parameters (gradient pre-multiplied by a symmetric positive-definite matrix) and, given a decay schedule that satisfies the Robbins-Monro step-size conditions, we are guaranteed to converge to a maximum of (3.8).

Chapter 4

EP Structured Variational Auto-Encoders

In this chapter we present our main contribution, a variant of the SVAE that employs the use of expectation propagation rather than mean-field to perform local inference. We refer to this approach as the EP-SVAE.

4.1 Motivation

The function $\boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$ of the SVAE as we have seen performs an optimisation over local variational parameters using the surrogate mean-field objective (3.6). This appears a natural choice for the problem at hand, as the surrogate objective is similar in form to the true mean-field objective $\mathcal{L}(\boldsymbol{\tau}_\theta, \boldsymbol{\tau}_x)$ (3.4). However, $\boldsymbol{\tau}_x^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$ is completely decoupled from $\mathcal{L}(\boldsymbol{\tau}_\theta, \boldsymbol{\tau}_x)$ and in principle could be any function that provides a parametric mapping $(\boldsymbol{\tau}_\theta, \boldsymbol{\phi}) \to \boldsymbol{\tau}_x$. In particular, we can choose $\boldsymbol{\tau}_x$ to be the result of optimising an arbitrary surrogate objective. A desirable property of such a surrogate objective however is that it should favour solutions $\boldsymbol{\tau}_x$ that result in *accurate* local posterior approximations $q(\mathbf{x}; \boldsymbol{\tau}_x)$.

It is not immediately apparent what is meant by accurate in this context. For the purposes of maximising a variational lower bound, as we showed in section 2.2, our measure of success is simply the (negative) exclusive KL divergence between our approximate posterior and the true posterior. However, maximising a variational lower bound is not our true objective, rather it is a proxy for maximising the marginal likelihood directly, made necessary due to intractabil-

ity of the original problem. Even maximising the marginal likelihood is itself an indirect way of tackling the somewhat harder to define goal of jointly learning a structured generative model with a method for performing fast, scalable and *accurate* inference in said model. Our primary considerations therefore should be that our inference scheme facilitates learning of an accurate generative model, while also providing accurate inference in the learned model (where the meaning of accurate is necessarily problem-specific). The exclusive KL divergence does not provide a direct measure of success for either of these objectives.

We consider here the use of EP to infer an approximate posterior for local latents $\mathbf{x}$ in place of MFVI, using the inferred natural parameters $\boldsymbol{\tau}_x$ as the local variational parameters in our global objective. EP and its variants are popular alternatives to the variational mean-field approach to approximate inference, often yielding somewhat better approximations [9].

As previously discussed, EP targets minimisation of the *inclusive* KL divergence $\mathrm{KL}[p(\mathbf{x}) \mid\mid q(\mathbf{x})]$, whereas variational approaches target the *exclusive* KL divergence $\mathrm{KL}[q(\mathbf{x}) \mid\mid p(\mathbf{x})]$. It may seem counter-intuitive therefore that we would target a divergence for local inference that favours somewhat different characteristics than the divergence we are targeting in our global objective [15]. However, this kind of approach is not without precedent. It is somewhat analogous to schemes that use EP and its variants as an approximate E step in the EM algorithm for learning in latent variable models. Furthermore, such schemes are the approach of choice for many models [24] [25]. Similarly, the Wake-Sleep algorithm [26] applied to the Helmholtz Machine effectively alternates between optimisation of the two KL divergences. While both of these examples have been applied successfully, we note that they lack convergence guarantees in general.

Another motivation for using EP for local inference in this class of models is that we have some additional flexibility in our choice of recognition network outputs. We can, as with the SVAE, use recognition potentials that are conjugate to the prior directly, in which case the recognition network is effectively learning to perform the projection step of EP directly (albeit without making use of contextual information). However, in order to be able to run EP efficiently on this model our only requirement is that we are able to compute the expectations of the statistics vector $\mathbf{t}_x(\mathbf{x})$ under the tilted distributions efficiently. As an example, if $p(\mathbf{x}|\boldsymbol{\theta})$, and therefore $q(\mathbf{x})$, is Gaussian, our recognition network may produce Mixture-of-Gaussian potentials, in which case the tilted distributions are also Mixture-of-Gaussian. Computing the expectations of $\mathbf{t}_x(\mathbf{x})$ under the

tilted distributions therefore has computational cost comparable to that of the conjugate potential approach, but the recognition network can now in principle learn to approximate the true potential arbitrarily well as we increase the number of components in the mixture.

4.2 Replacing the Surrogate Objective

We consider a new objective $\mathcal{L}_{\text{EPSVAE}}$ that is analogous to the SVAE objective (3.8), but we choose $\boldsymbol{\tau}_x$ to instead be a local partial optimiser of a different objective, in particular, the EP energy function of a surrogate model.

$$\mathcal{L}_{\text{EPSVAE}}(\boldsymbol{\tau}_\theta, \boldsymbol{\gamma}, \boldsymbol{\phi}) := \mathcal{L}(\boldsymbol{\tau}_\theta, \hat{\boldsymbol{\tau}}_x^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi}), \boldsymbol{\gamma}) \tag{4.1}$$

where

$$\begin{aligned} \hat{\boldsymbol{\tau}}_x^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi}) &= \sum_{i=1}^{K} \boldsymbol{\tau}_{x,i} \\ \{\boldsymbol{\tau}_{x,i}\} &= \underset{\{\boldsymbol{\tau}_{x,i}\}}{\arg\max}\, \tilde{Z}(\{\boldsymbol{\tau}_{x,i}\}; \boldsymbol{\tau}_\theta, \boldsymbol{\phi}) \end{aligned} \tag{4.2}$$

and $\tilde{Z}$ is the EP energy function (2.53) of Section 2.6 for a surrogate model (defined below), but we have made explicit the dependence of this energy on $\boldsymbol{\tau}_\theta$, $\boldsymbol{\phi}$, which parameterise the target distribution. The summation above is required because the EP energy is a function of the individual site parameters, whereas we are interested in the combined approximation. We again here use arg max to refer to any maximum, not necessarily a global maximum.

We do not evaluate the objective $\tilde{Z}$ explicitly during local inference, rather we optimise it implicitly by running EP to convergence on our surrogate model. Our surrogate model replaces the non-conjugate observation likelihoods with recognition network potentials

$$\hat{p}(\mathbf{x}; \mathbf{y}, \boldsymbol{\tau}_\theta, \boldsymbol{\phi}) :\propto p(\mathbf{x} \mid \hat{\boldsymbol{\eta}}_x)\psi(\mathbf{x}; \mathbf{y}, \boldsymbol{\phi}) \tag{4.3}$$

where $\psi(\mathbf{x}; \mathbf{y}, \boldsymbol{\phi})$ is the output of our recognition network. In general, the factors $p(\mathbf{x} \mid \hat{\boldsymbol{\eta}}_x)$ and $\psi(\mathbf{x}; \mathbf{y}, \boldsymbol{\phi})$ may have additional structure. Note that for this local optimisation, we use a point estimate of the natural parameters of $p(\mathbf{x}|\boldsymbol{\theta})$,

specifically their expectation under our approximate posterior $q(\theta|\boldsymbol{\tau}_\theta)$

$$\hat{\boldsymbol{\eta}}_x := \left\langle \boldsymbol{\eta}_x(\boldsymbol{\theta}) \right\rangle_{\tau_\theta} \tag{4.4}$$

This is similar to the approach taken in mean-field / variational bayes, in which the update for the approximation $q(\mathbf{x})$ does not depend on $p(\boldsymbol{\theta})$ directly, but instead replaces the global natural parameters with their expectations under $q(\boldsymbol{\theta})$.

In order to optimise objective (4.1), we require the fixed point obtained by EP, $\hat{\boldsymbol{\tau}}^*_x(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$, to be differentiable. We note that it cannot be the case that this fixed point is differentiable everywhere, as in general convergence is not guaranteed and so $\hat{\boldsymbol{\tau}}^*_x(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$ may not exist for all $\boldsymbol{\tau}_\theta$, $\boldsymbol{\phi}$. However, we proceed in the hope that encountering such situations is sufficiently rare in practice that the EP-SVAE remains a useful approach. The conditions under which $\hat{\boldsymbol{\tau}}^*_x(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$ remains differentiable is a question for future research.

The parametric mapping $\hat{\boldsymbol{\tau}}^*_x(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$, which is the result of optimising the EP energy of our surrogate model (4.3), then gives us the local variational parameters for use in our global objective (4.1). Optimisation of the remaining parameters is then performed as with the SVAE. In particular we optimise $\boldsymbol{\tau}_\theta$ using the stochastic natural gradient estimates of section 3.3, with the recognition and likelihood network parameters $\boldsymbol{\phi}$, $\boldsymbol{\gamma}$ optimised using standard gradient-based techniques.

Chapter 5

Experiments

In this chapter we present proof-of-concept experiments to demonstrate the EP-SVAE approach. We begin with a latent Gaussian mixture model (GMM), in particular using the 2D toy problem of [1]. However, for this model, using the same factorisation as that of the SVAE, we find that EP local inference converges in a single step. We therefore introduce a second model which we call the latent cycle GMM, which in general requires several iterations of EP in order to reach convergence.

5.1 Latent Gaussian Mixture Model

5.1.1 Model Setup

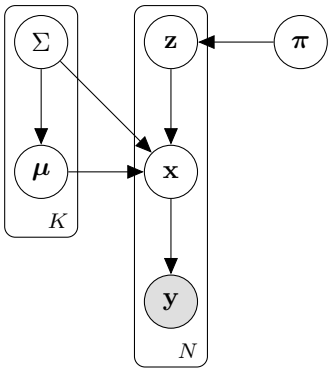

Figure 5.1: Directed graph for latent GMM with N i.i.d observations

We consider the latent Gaussian mixture model (GMM) of [1] with conjugate prior over latent generative parameters

$$\begin{aligned} p(\boldsymbol{\pi}) &= \mathrm{Dir}(\boldsymbol{\pi}) \\ p(\Sigma_i, \boldsymbol{\mu}_i) &= \mathrm{NIW}(\Sigma_i, \boldsymbol{\mu}_i) \\ p(z^{(n)}|\boldsymbol{\pi}) &= \mathrm{Categorical}(\boldsymbol{\pi}) \\ p(\mathbf{x}^{(n)}|z^{(n)}, \{\Sigma_i, \boldsymbol{\mu}_i\}) &= \mathcal{N}\big(\mathbf{x}^{(n)} \mid \boldsymbol{\mu}_{z^{(n)}}, \Sigma_{z^{(n)}}\big) \\ p(\mathbf{y}^{(n)}|\boldsymbol{\gamma}) &= \mathcal{N}\big(\mathbf{y}^{(n)} \mid \boldsymbol{\mu}(\mathbf{x}^{(n)}; \boldsymbol{\gamma}), \Sigma(\mathbf{x}^{(n)}; \boldsymbol{\gamma})\big) \end{aligned}$$

where $\boldsymbol{\mu}(\mathbf{x}^{(n)}; \boldsymbol{\gamma})$ and $\Sigma(\mathbf{x}^{(n)}; \boldsymbol{\gamma})$ provide a smooth, nonlinear parametric mapping from latent $\mathbf{x}$ to the parameters of Gaussian $\mathbf{y}$, which we assume to be given by neural networks. Going forwards we shall focus on the case of a single observation ($N = 1$) in order to keep notation uncluttered. The extension to multiple observations is straightforward.

In [1] the authors consider a variational mean-field approximation of the following form

$$q(\{\Sigma_i, \boldsymbol{\mu}_i\}, \boldsymbol{\pi}, z, \mathbf{x}) = \left(\prod_{i=1}^{K} q(\Sigma_i, \boldsymbol{\mu}_i)\right) q(\boldsymbol{\pi}) q(z) q(\mathbf{x}) \tag{5.1}$$

where each approximation remains in the same exponential family as the prior. In particular,

$$\begin{aligned} q(\boldsymbol{\pi}) &\propto \exp\{\boldsymbol{\tau}_\pi^\top \mathbf{t}_\pi(\boldsymbol{\pi})\} \\ q(\Sigma_i, \boldsymbol{\mu}_i) &\propto \exp\{\boldsymbol{\tau}_{\Sigma,\mu}(i)^\top \mathbf{t}_{\Sigma,\mu}(\Sigma_i, \boldsymbol{\mu}_i)\} \\ q(z) &\propto \exp\{\boldsymbol{\tau}_z^\top \mathbf{t}_z(z)\} \\ q(\mathbf{x}) &\propto \exp\{\boldsymbol{\tau}_x^\top \mathbf{t}_x(\mathbf{x})\} \end{aligned} \tag{5.2}$$

where $\mathbf{t}_\pi(\boldsymbol{\pi})$, $\mathbf{t}_{\Sigma,\mu}(\Sigma, \boldsymbol{\mu})$, $\mathbf{t}_z(z)$ and $\mathbf{t}_x(\mathbf{x})$ are the sufficient statistics of the Dirichlet, Normal-Inverse-Wishart, Categorical and Gaussian families respectively. Note that while EP is not strictly a variational approach, we shall still refer to $\boldsymbol{\tau}_z$, $\boldsymbol{\tau}_x$ as the local variational parameters given that our overall objective in these parameters is a variational lower bound.

In prior discussion we exclusively used $\mathbf{x}$ to refer to local latent variables,

however, as we now have additional factorisation structure our local latent variables consist of $\mathbf{x}$ and z. We shall use $\boldsymbol{\theta} := (\Sigma, \boldsymbol{\mu}, \boldsymbol{\pi})$ to refer collectively to the global generative parameters over latents.

It is worth noting that for this problem the local factorisation assumption $q(\mathbf{x}, z) = q(\mathbf{x})q(z)$ is not strictly necessary as local inference on our surrogate model, for given global parameters, already lies within a tractable exponential family. However, by enforcing the same factorisation constraints as [1] we are able to isolate the effect of replacing the MF local inference procedure with EP without confounding the results by also allowing for greater structure in the posterior.

As outlined in chapter 4, in order to compute our local variational parameters $\boldsymbol{\tau}_z^*$, $\boldsymbol{\tau}_x^*$, given global variational parameters $\boldsymbol{\tau}_\theta$, we run EP to convergence on a surrogate model to find a posterior approximation $q(\mathbf{z}|\boldsymbol{\tau}_z^*)q(\mathbf{x}|\boldsymbol{\tau}_x^*)$ within our chosen family. This surrogate model consists of the prior on local latents, as well recognition network potentials which act as approximations to the observation likelihoods. The target distribution then, which we denote $\tilde{p}(\mathbf{x}, z)$, is defined as follows

$$\begin{aligned}\tilde{p}(z, \mathbf{x}) &\propto p(z \mid \hat{\boldsymbol{\eta}}_z)p(\mathbf{x} \mid z, \{\hat{\boldsymbol{\eta}}_x(i)\})\psi(\mathbf{x}|\mathbf{y}, \boldsymbol{\phi}) \\ &= \text{Categorical}(z \mid \hat{\boldsymbol{\eta}}_z)\mathcal{N}(\mathbf{x}|\hat{\boldsymbol{\eta}}_x(z)) \exp\{\mathbf{t}_x(\mathbf{x})^\top \mathbf{r}(\mathbf{y}; \boldsymbol{\phi})\} \qquad (5.3)\end{aligned}$$

where we take our point estimates of the global natural parameters to be their expectations under the current approximate posterior $q(\boldsymbol{\theta})$. Note that due to the conjugacy structure of our model these are given by the mean parameters of $q(\boldsymbol{\theta}) = q(\boldsymbol{\pi}) \prod_i q(\Sigma_i, \boldsymbol{\mu}_i)$.

$$\begin{aligned}\hat{\boldsymbol{\eta}}_z &:= \langle \log \boldsymbol{\pi} \rangle_{q(\boldsymbol{\pi})} \\ \hat{\boldsymbol{\eta}}_x(i) &:= \left\langle \left(\Sigma_i^{-1}\boldsymbol{\mu}_i, -\frac{1}{2}\Sigma_i^{-1}\right) \right\rangle_{q(\boldsymbol{\mu}_i, \Sigma_i)}\end{aligned}$$

The surrogate model contains three factors, two of which, $p(z \mid \hat{\boldsymbol{\eta}}_z)$ and $\psi(\mathbf{x}|\mathbf{y}, \boldsymbol{\phi})$, already lie within our approximating family. We therefore only need to project the remaining factor $p(\mathbf{x} \mid \{\hat{\eta}_{x_i}\})$ into our family. Furthermore, as we only have one factor to approximate, no iteration is required and the unique optimum is found in a single step. See Appendix A.5 for further details.

Let $\hat{\boldsymbol{\tau}}_{x,z}^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$ return the local parameters resulting from running EP until convergence on our surrogate model. The resulting parameters $\boldsymbol{\tau}_z^*$, $\boldsymbol{\tau}_x^*$ are then used as the local variational parameters in the EP-SVAE objective (4.1), which

we restate here with our additional local factorisation

$$\mathcal{L}_{\text{EPSVAE}}(\boldsymbol{\tau}_\theta, \boldsymbol{\gamma}, \boldsymbol{\phi}) := \mathcal{L}(\boldsymbol{\tau}_\theta, \boldsymbol{\tau}_z^*, \boldsymbol{\tau}_x^*, \boldsymbol{\gamma}) \tag{5.4}$$

where

$$\{\boldsymbol{\tau}_x^*, \boldsymbol{\tau}_z^*\} = \hat{\boldsymbol{\tau}}_{x,z}^*(\boldsymbol{\tau}_\theta, \boldsymbol{\phi}) \tag{5.5}$$

The remaining parameters $\boldsymbol{\tau}_\theta$, $\boldsymbol{\phi}$, $\boldsymbol{\gamma}$ are then optimised as described in section 4.2.

5.1.2 Results

In order to demonstrate the EP-SVAE on this model we extended the experimental code of [1] by introducing our EP inference routine. We trained the EP-SVAE latent GMM on a synthetic 2-dimensional dataset ($\dim(\mathbf{y}) = 2$) where points are arranged in a swirling pinwheel formation. The generative model was also taken to have 2-dimensional Gaussian latents ($\dim(\mathbf{x}) = 2$), with $K = 15$ latent clusters. Note that while the dataset clearly exhibits 5 distinct clusters (see figure 5.2), we allow for a greater number of latent components in the generative model in the expectation that many of the mixture probabilities will be found to be insignificant.

Figure 5.2 displays the synthetic data as black points, along with the density learned by the model. Each colour represents the conditional density for a single value of the latent z. The translucency of each cluster depends on the corresponding mixture probability, so that clusters with low probability are less visible. The two plots were generated with identital initial conditions and hyperparameters, with the only difference being the choice of local inference routine.

The results of the two approaches were qualitatively very similar. We typically observed that the probability of most clusters approaches zero, with 5 clusters retaining significant probability, corresponding to the 5 distinctly visible clusters in the observations.

In some runs however, in particular if we allowed training to run for a long time, we observed more or less than 5 mixture components identified as having significant probabilities, for both the MF and EP approaches. In such cases a single mixture component may be responsible for multiple visible clusters in the learned model, or conversely, multiple mixture components may be responsible

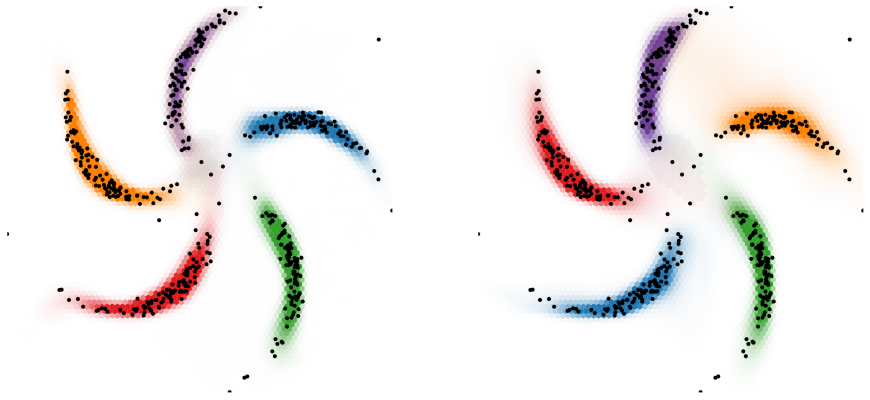

Figure 5.2: Synthetic observations (black) and generative density (multi) for the latent GMM pinwheel example. EP-SVAE (left), SVAE (right).

for a single visible cluster. We noticed that the former was more common with MF, whereas the latter was more common with EP. This may be due to the characteristics of the different KL divergences targeted by local inference in the two approaches. In particular, due to the zero-avoiding nature of the inclusive KL divergence [15], EP will tend to share responsibility among multiple latent components if the membership of an observation is amibguous. Conversely, the exclusive KL, and therefore MF, will tend to prefer hard assignment to a single cluster in such situations. If we have multiple mixture components covering similar sets of points with varying prior probabilities, it is likely MF will allow one component to dominate in the sense of being assigned most of the responsibility for those points. This in turn will cause the inferred prior probabilities of the other components to shrink. EP however has a tendency to hedge its bets, likely resulting in multiple clusters retaining significant prior probability in such cases, as we observed.

It is worth adding that we used a sparsifying Dirichlet prior for the latent cluster responsibilities. If the learned model is able to fit the data with low reconstruction error using only a single latent mixture component, then this solution may result in a higher value of the ELBO objective than a solution with 5 latent components and similar reconstruction error. This solution is clearly not preferable however from the viewpoint of recovering a meaningful structured representation of the data. While we did not observe complete collapse to a single cluster with either approach, this may be more problematic as we increase

the flexibility of the decoder network, as a sufficiently powerful network can reshape the mass of a single latent cluster into arbitrary observation density shapes. This is analogous to the posterior collapse observed in autoregressive VAE models [27].

5.2 Latent Cycle Gaussian Mixture Model

While the latent GMM of the previous section demonstrated that, in principle, EP may be used for local inference in an SVAE-like scheme, the unique EP fixed point was reached in a single step. In order to demonstrate the applicability of the EP-SVAE in slightly more general settings, we now introduce a model in which EP typically requires several iterations before converging.

5.2.1 Model Setup

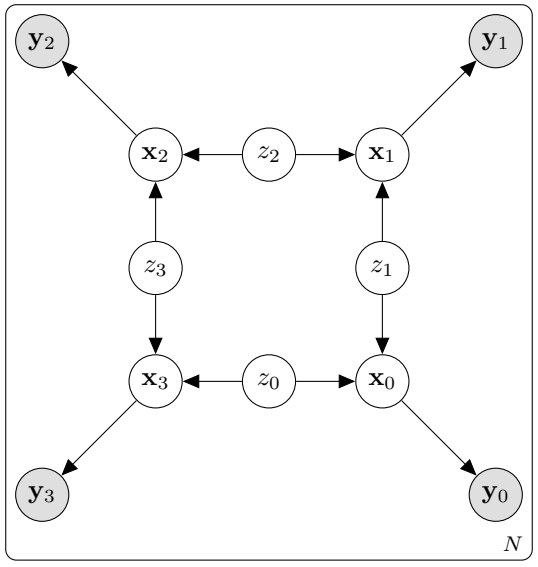

Figure 5.3: Directed graph for latent cycle GMM with $K = 4$. Global latents are omitted.

Note that we shall find it convenient to use 0-based indexing for latents throughout this section. While we allow for N conditionally independent latent observations in our model, we consider the case of a single observation in the remainder of this subsection in order to keep notation manageable. The extension

to multiple observations is straightforward.

Our global parameters consist of 4 Gaussian means and variances $\boldsymbol{\mu}_{l,r}$, $\Sigma_{l,r}$ for $(l,r) \in \{0,1\}^2$ with Normal-Inverse-Wishart priors, as well as K probabilities $\{\pi_i\}$ for $i \in \{0,...,K-1\}$ with independent Beta priors. We refer to the global parameters collectively as $\boldsymbol{\theta}$, and their prior is given by

$$p(\boldsymbol{\theta}) = \left(\prod_{(l,r)\in\{0,1\}^2} \text{NIW}(\boldsymbol{\mu}_{l,r}, \Sigma_{l,r}) \right) \prod_{i=0}^{K-1} \text{Beta}(\pi_i) \tag{5.6}$$

where we have omitted prior hyperparameters to help simplify notation. Note that going forwards any index over (l,r) should be viewed implicitly as an index over $\{0,1\}^2$.

Our local variables consist of K discrete binary latents $\{z_i : i = 0,...,K-1\}$ which are independent in the prior. We then further have K Gaussian latents $\{\mathbf{x}_i : i = 0,...,K-1\}$, each with mean and variance that depend on the state of two adjacent binary latents (see Figure 5.3). That is

$$\begin{aligned} p(z_i|\boldsymbol{\theta}) &= \text{Bernoulli}(z_i|\pi_i) \\ p(\mathbf{x}_i|z_i, z_{j(i)}, \boldsymbol{\theta}) &= \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_{z_i,z_{j(i)}}, \Sigma_{z_i,z_{j(i)}}) \end{aligned} \tag{5.7}$$

for $i \in \{0,...,K-1\}$ where

$$j(i) := (i+1) \bmod K \tag{5.8}$$

The marginal of each latent $\mathbf{x}_i$ is then a mixture of 4 Gaussians, where we have dependence between the mixture components of each $\mathbf{x}_i$, due to overlapping dependence on the binary latents $\{z_i\}$. Note that our prior $p(\boldsymbol{\theta})$ is then conjugate to $p(\{\mathbf{x}_i, z_i\}|\boldsymbol{\theta})$

Finally we have K observed values $\{\mathbf{y}_i : i = 0,...,K-1\}$ which are conditionally Gaussian, but with mean and (diagonal) variance given by a non-linear function of the corresponding Gaussian latent.

$$p(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\gamma}) := \mathcal{N}(\mathbf{y}_i \mid \boldsymbol{\mu}(\mathbf{x}_i;\boldsymbol{\gamma}), \sigma(\mathbf{x}_i;\boldsymbol{\gamma})) \tag{5.9}$$

Our full model is then given by

$$p(\boldsymbol{\theta}, \{z_i, \mathbf{x}_i, \mathbf{y}_i\}|\boldsymbol{\gamma}) := p(\boldsymbol{\theta}) \prod_i p(z_i|\boldsymbol{\theta}) p(\mathbf{x}_i|z_i, z_{j(i)}, \boldsymbol{\theta}) p(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\gamma}) \tag{5.10}$$

Figure 5.3 shows a directed graph for local latents with $K = 4$, where we have omitted global parameters in order to keep the diagram relatively simple.

Our approximating family $\mathcal{Q}$ consists of distributions of the form

$$q(\{\Sigma_{l,r}, \boldsymbol{\mu}_{l,r}\}, \{\pi_i, z_i, \mathbf{x}_i\}) = \Big(\prod_{(l,r)} q(\Sigma_{l,r}, \boldsymbol{\mu}_{l,r}) \Big) \prod_i q(\pi_i) q(z_i) q(\mathbf{x}_i) \tag{5.11}$$

where, as before, each approximate distribution remains in the same exponential family as the prior. Specifically

$$\begin{aligned} q(\pi_i) &\propto \exp\{\boldsymbol{\tau}_{\pi_i}^\top \mathbf{t}_\pi(\pi_i)\} \\ q(\Sigma_{l,r}, \boldsymbol{\mu}_{l,r}) &\propto \exp\{\boldsymbol{\tau}_{\Sigma,\mu}(l,r)^\top \mathbf{t}_{\Sigma,\boldsymbol{\mu}}(\Sigma_{l,r}, \boldsymbol{\mu}_{l,r})\} \\ q(z_i) &\propto \exp\{\boldsymbol{\tau}_{z_i}^\top \mathbf{t}_z(z_i)\} \\ q(\mathbf{x}_i) &\propto \exp\{\boldsymbol{\tau}_{x_i}^\top \mathbf{t}_x(\mathbf{x}_i)\} \end{aligned} \tag{5.12}$$

where $\mathbf{t}_\pi(\pi)$, $\mathbf{t}_{\Sigma,\boldsymbol{\mu}}(\Sigma, \boldsymbol{\mu})$, $\mathbf{t}_z(z)$ and $\mathbf{t}_x(\mathbf{x})$ are the sufficient statistic functions of the Beta, Normal-Inverse-Wishart, Bernoulli and Gaussian families respectively. Note that we are using the overcomplete Bernoulli representation for z_i so that $\mathbf{t}_z(z) = (\delta(z = 0), \delta(z = 1))$.

In order to perform efficient approximate inference in this model, we again employ the use of a surrogate graph where we replace the non-linear likelihood terms $p(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\gamma})$ with recognition network potentials $\psi(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\phi})$. We again take point-estimates of the global natural parameters to be their expectations under our current approximation $q(\boldsymbol{\theta})$

$$\begin{aligned} \hat{\boldsymbol{\eta}}_{z_i} &:= \langle \big(\log(\pi_i), \log(1 - \pi_i) \big) \rangle_{q(\pi_i)} \\ \hat{\boldsymbol{\eta}}_x(l,r) &:= \Big\langle \big(\Sigma_{l,r}^{-1} \boldsymbol{\mu}_{l,r}, -\frac{1}{2} \Sigma_{l,r}^{-1} \big) \Big\rangle_{q(\boldsymbol{\mu}_{l,r}, \Sigma_{l,r})} \end{aligned} \tag{5.13}$$

Our (normalised) target distribution for EP is then

$$\begin{aligned} \tilde{p}(\{z_i, \mathbf{x}_i\} \mid \hat{\boldsymbol{\eta}}_z, \{\hat{\boldsymbol{\eta}}_x\}) &:\propto \prod_i p(z_i \mid \hat{\boldsymbol{\eta}}_{z_i}) p(\mathbf{x}_i \mid z_i, z_{j(i)}, \{\hat{\boldsymbol{\eta}}_x(l,r)\}) \psi(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\phi}) \\ &= \prod_i \text{Bernoulli}(z_i|\hat{\boldsymbol{\eta}}_{z_i}) \mathcal{N}(\mathbf{x}_i|\hat{\boldsymbol{\eta}}_x(z_i, z_{j(i)})) \psi(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\phi}) \end{aligned} \tag{5.14}$$

Let $\hat{\boldsymbol{\tau}}^*_{x,z}(\boldsymbol{\tau}_\theta, \boldsymbol{\phi})$ return the local parameters $\{\hat{\boldsymbol{\tau}}^*_{z_i}, \hat{\boldsymbol{\tau}}^*_{z_i}\}$ resulting from running EP upon convergence on our surrogate model. These local parameters are then used as the variational parameters in the EP-SVAE objective (4.1), which we restate here with our additional local factorisation

$$\mathcal{L}_{\text{EPSVAE}}(\boldsymbol{\tau}_\theta, \boldsymbol{\gamma}, \boldsymbol{\phi}) := \mathcal{L}(\boldsymbol{\tau}_\theta, \{\hat{\boldsymbol{\tau}}^*_{z_i}, \hat{\boldsymbol{\tau}}^*_{x_i}\}, \boldsymbol{\gamma}) \tag{5.15}$$

where

$$\{\hat{\boldsymbol{\tau}}^*_{z_i}, \hat{\boldsymbol{\tau}}^*_{z_i}\} = \hat{\boldsymbol{\tau}}_{x,z}(\boldsymbol{\tau}_\theta, \boldsymbol{\phi}) \tag{5.16}$$

Finally, as before, the remaining variational paramters $\boldsymbol{\eta}_\theta$ and network parameters $\boldsymbol{\phi}$, $\boldsymbol{\gamma}$ are optimised as outlined in section 4.2.

5.2.2 EP-SVAE Results

To demonstrate the EP-SVAE on this model we generated a synthetic dataset with 2-dimensional observations ($\dim(\mathbf{y}_i^{(n)}) = 2$). The observations were synthesised using the generative model over local latents $p(\{\mathbf{x}_i^{(n)}, z_i^{(n)}\}|\boldsymbol{\theta})$ as detailed above, with $K = 3$, and 2-dimensional Gaussian latents ($\dim(\mathbf{x}_i^{(n)}) = 2$). The mean and variance of the latent clusters were chosen to create a crosshair formation in the latent space. The observations $\{\mathbf{y}_i^{(n)}\}$ were then generated by applying a swirling non-linear transformation to the Gaussian latents so that the resulting density of each observation again had the appearance of a pinwheel (see Figure 5.4). We generated $N = 800$ independent observations from this model, each resulting in a triplet of black dots, one for each of the 3 plots in Figure 5.4.

Each of the 3 sub-plots in Figure 5.4 corresponds to a value of $i \in \{0, 1, 2\}$, with each black dot corresponding to a single observation $\mathbf{y}_i^{(n)}$. The coloured clouds show the conditional density for each of the 4 possible configurations of adjacent binary latents $(z_i^{(n)}, z_{j(i)}^{(n)}) \in \{0, 1\}^2$.

It is apparent from Figure 5.7 that the EP-SVAE is able to learn the structure and marginal density of the data well. It is interesting however that the generative density appears to fit the data less sharply than that observed for the latent GMM of the previous section.

It is insightful also to observe the data represention in the latent space. Figure 5.5 shows the mean corresponding to the Gaussian recognition network potential of each data point (in black). We focus on just $i = 0$ here, the

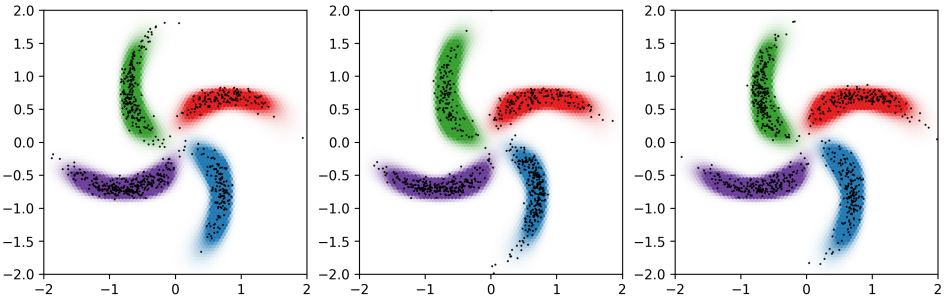

Figure 5.4: Synthetic observations and overlaid EP-SVAE density for latent cycle GMM pinwheel example, with a ring of $K = 3$ observations. Co-ordinates are in data space.

plots for $i \in \{1, 2\}$ are qualitatively similar. In red, we have also plotted the means $\langle \mathbf{x}_i^{(n)} \rangle_{q(\mathbf{x}_i^{(n)})}$ obtained by the EP local inference routine, which is itself a function of the recognition potentials and the prior. Finally, the 4 coloured ellipses correspond to the 2 standard-deviation contours of each latent prior Gaussian component.

We see here that the learned representation largely captures just a single dimension of variation within each cluster. The model could in principle achieve lower reconstruction error by capturing two directions of variation within the clusters and so the reasons for the EP-SVAE settling on this solution are not clear. This may be contributing to the lack of sharpness relative to the latent GMM of the previous section.

It is useful also to consider whether the recognition network produces potentials that are in some sense reasonable approximations to the true observation likelihoods. This is not necessarily desirable in itself, however it may give us some indication as to the robustness of the learned representations.

It is conceivable that the recognition network could learn to produce potentials that bear little resemblance to the true observation likelihoods, provided that the resulting local parameters found by EP result in relatively high values of the global objective. In order to perform approximate EP inference on this model, we require that the encoder produces Gaussian potentials that result in moments that are similar to the observation likelihoods when taken in the context of the cavity distribution. This is a rather different requirement than to say that the moments of the Gaussian approximation should be similar to those of the standalone likelihood terms. However, as our recognition networks are a function of $\mathbf{y}_i$ and $\boldsymbol{\phi}$ only, they have no direct access to contextual information

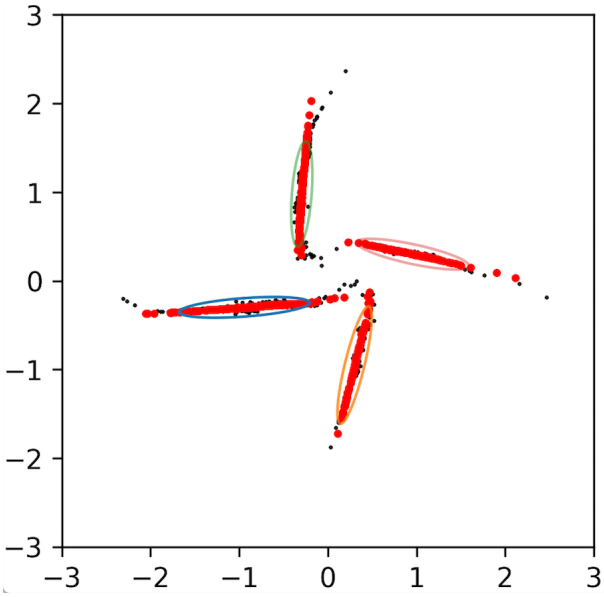

Figure 5.5: EP-SVAE latent space plot for latent cycle GMM pinwheel example, with a ring of $K = 3$ observations, displaying $i = 0$ only. Encoder potential means (black), EP inference means (red) and 2-SD Gaussian prior contours (multi). Co-ordinates are in latent space.

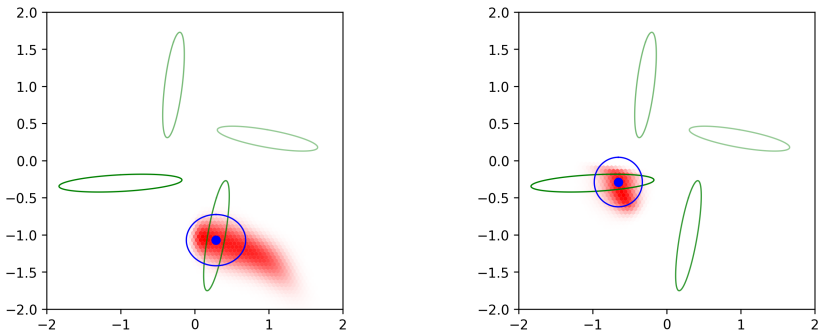

Figure 5.6: EP-SVAE Gaussian encoder potential mean and 2-SD contour (blue), true likelihood potential (red) and Gaussian prior 2-SD contours (green). Displaying 2 independent observations for $i = 0$ only. Co-ordinates are in latent space.

and so are restricted to learning to produce Gaussian potentials that produce similar moments *on average*, over all observed cavity distributions.

Figure 5.6 shows the Gaussian encoder potentials $\psi(\mathbf{x}; \mathbf{y}, \boldsymbol{\phi})$ (blue), along with the true likelihood potential $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\gamma})$ when viewed as a function of $\mathbf{x}$ (red) and the latent prior Gaussian components (green). We display the plot for 2 independent datapoints from the latent cycle GMM pinwheel problem (observing $i = 0$ only).

In the right plot of Figure 5.6 we see that the encoder potential appears to be a reasonable Gaussian approximation to the factor overall. The left plot however shows that the encoder potential provides a Gaussian approximation to the factor within a limited region only. In particular, it produces a Gaussian approximation to the likelihood within the region of high values of the Gaussian prior (and therefore of the average cavity distribution). The encoder therefore seems to be learning to make a sensible approximation to the true likelihood, albeit within regions of high prior density only.

5.2.3 SVAE Results

When attempting the SVAE approach on this dataset, we obtained consistently poor results. The scheme would eventually settle on a trivial solution where the entire density is modelled with a single cluster (see Figure 5.7). While this may be a reasonable approximation to the marginal density of any single $i \in \{0, 1, 2\}$, it captures no useful latent structure and therefore none of the

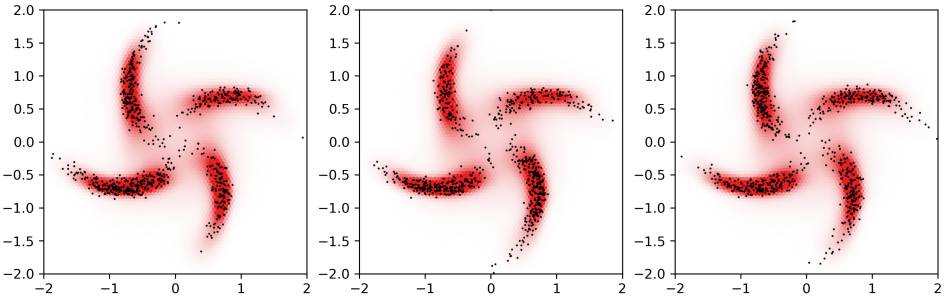

Figure 5.7: Synthetic observations and overlaid SVAE density for latent cycle GMM pinwheel example, with a ring of $K = 3$ observations. Co-ordinates are in data space.

dependence between observations. Details of the local MF updates for this model can be found in Appendix A.7.

In order to isolate the cause of these issues, we experimented with mean-field inference on a similar model but with linear dependence between the latents $\mathbf{x}_i$ and observations $\mathbf{y}_i$. We observed that if the observations have little noise, relative to the distance between clusters, then mean-field is able to provide accurate inference. However, in the presence of significant noise in the observations, so that cluster membership of each point is relatively ambiguous, the mean-field approximation quickly degrades to the point where it is little better than random guessing. This is in contrast to EP / loopy BP inference which in the same problem was able to provide relatively accurate inference even in the presence of significant noise. The observations in this linear model played the role of the encoder potentials. While the encoder potentials may have high precision eventually, they have relatively low precision early on during training, which is equivalent to the high observation noise scenario just described.

We believe the inability of mean-field to provide accurate inference on this problem is likely related to the strong, near-deterministic interactions between latent variables, combined with the tendency of mean-field approximations to over-commit to particular representations. The cluster membership of any particular $\mathbf{x}_i$ is fully determined by the membership of its imediate neighbours. If the observations are relatively noisy but mean-field is overly confident in assigning observations to clusters regardless, the relatively weak encoder potentials may be dominated by the strong, yet largely random messages arriving from adjacent nodes. This is mostly conjecture however and fully investigating the cause of these problems is beyond the scope of this work.

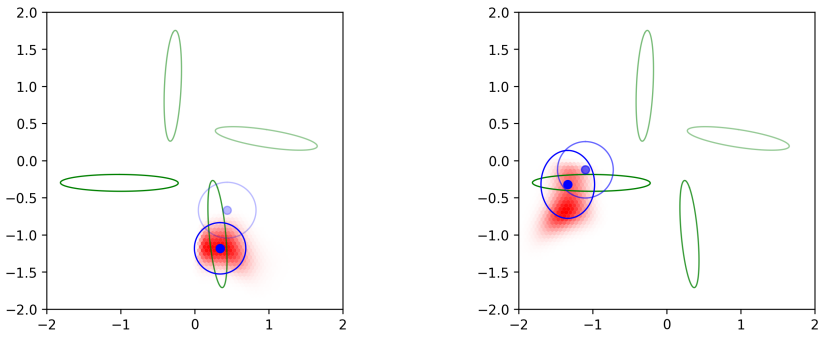

Figure 5.8: MoG encoder potential means and 2-SD contours (blue), true likelihood potential (red) and Gaussian prior 2-SD contours (green). Displaying 2 independent observations for $i = 0$ only.

5.2.4 Extensions

With the EP-SVAE we have additional flexibility in our choice of encoder potential that was not available to us in the SVAE. In particular, whereas the SVAE is restricted to using conjugate recognition network potentials, our only constraint with the EP-SVAE is that we must be able to compute the expected sufficient statistics of the approximating family under the tilted distributions efficiently.

As a concrete example, if our prior distribution $p(\mathbf{x}|\boldsymbol{\theta})$, and therefore $q(\mathbf{x})$ is Gaussian, a natural choice would be to choose our encoder potentials $\psi(\mathbf{x}|\mathbf{y}, \boldsymbol{\phi})$ to have Gaussian form, as we have done in the preceding sections. However, we can also perform approximate inference on this graph efficiently using EP if our encoder potentials have the form of a Mixture-of-Gaussians (MoG). In this case, the tilted distributions for the encoder potentials are also MoG, and so we can compute expected sufficient statistics with computational cost that is comparable to the conjugate case.

To demonstrate this approach, we repeated the experiment of the preceding subsection using MoG encoder potentials with 2 mixture components. For this problem the results were were virtually indistinguishable from the conjugate potential case, both qualitatively and in the observed ELBO sequences during training. We have omitted these plots as they provide no further insight.

Figure 5.8 shows a selection of encoder potential plots for the MoG model with 2 components, which is analogous to Figure 5.6. The translucency of each Gaussian in the plots is determined by its mixture probability. We can see that

the encoder is utilising both mixture components, however it is clear that the additional flexibility results in no meaningful improvement for this problem. In most cases the two components are largely overlapping and not significantly different than a single Gaussian.

However, the current problem seems particularly well suited to the conjugate potential approximation because the latent cluster membership is well determined by the observation $\mathbf{y}$. The encoder is therefore effectively able to predict the cavity distribution for a given datapoint, as it can determine with near certainty which cluster the point belongs to solely based on the observation. This would not be the case for example if the observations were noisier so that the cluster boundaries are less clear. In such cases, and for more general problems, the encoder may need to produce potentials that are good approximations in the context of a wider range of possible cavity distributions for a given observation. This becomes easier if the encoder has more flexibility in the form of its potentials. In fact, if we can increase the number of mixture components without limit, the encoder can in principle learn to approximate the standalone likelihood potential arbitrarily well, resulting in good approximations in the context of *any* cavity distribution.

5.2.5 Relation to Error-Correcting Coding

The problem described in this section bears close resemblance to a probabilistic error-correcting coding (ECC) scheme. We can consider the binary latents $z_i : i = 0, ..., K-1$ to be a latent message of K bits. Each observation $\mathbf{y}_i : i = 0, ..., K-1$ is then a noisy (continuous) observation of the state of the 2 adjacent binary latents z_i, $z_{j(i)}$. In order to recover the values of the latent message we must combine the noisy observations subject to the constraint that the inferred latents for all observations must agree. This viewpoint may provide some insight into the relative success of the EP-SVAE on this model when compared to the SVAE.

In particular, for many graphical codes the most successful decoder is based on applying loopy belief propagation to the underlying (necessarily loopy) graphical model [10] [28] [29]. Our EP scheme on the model of this section is equivalent to performing loopy BP over the binary latents, and so it is perhaps unsurprising that the EP-SVAE performs particularly well on this problem.

Chapter 6

Outlook

The experimental results show that in principle we may use EP for local inference in an SVAE like scheme for certain models. However, several significant questions remain. Perhaps most pressing is the issue of convergence. For the problems we considered here, convergence of EP was guaranteed. In the latent GMM problem only one factor required projecting to the approximating family, and so EP was guaranteed to converge in a single step. For the latent cycle GMM, the message passing equations for the discrete latents were exactly those of loopy belief propagation, and furthermore, these updates did not depend on the Gaussian messages. We could therefore iterate the discrete message passing updates until convergence and then compute Gaussian messages as a final step. As our discrete sub-graph consisted of a single loop, convergence was guaranteed [28].

In general graphs, convergence of EP is not guaranteed. An important avenue for future research is to assess the convergence properties of the EP-SVAE in more general settings. Particular questions to be addressed are whether convergence problems of EP may be exacerbated by encoder potentials that can be largely random during early training iterations. Optimisation of our objective also requires differentiability of the EP fixed point (with respect to $\boldsymbol{\eta}_\theta$, $\boldsymbol{\phi}$). While it cannot be true that the fixed point remains differentiable everywhere, it is an open question to what extent this will cause issues in practice with the EP-SVAE.

One view of the role of the recognition network of the EP-SVAE in the case of conjugate potentials is that it is performing the projection step of EP for the observation likelihoods. A key contributor to the success of EP in providing ac-

curate approximations is that approximate factors are refined within the context of the global approximation. However, as our recognition network is a function of only the observations $\mathbf{y}$ and parameters $\boldsymbol{\phi}$, it is restricted to performing the projection step without making use of contextual information from the cavity distributions.

One way to relax the severity of this approximation is to use more flexible encoder potential forms, such as Mixture-of-Gaussians, followed by a real projection step on these non-conjugate potentials, as demonstrated in Section 5.2.4. An alternative approach however may be to provide the cavity information as an input to the recognition network. Given a sufficiently powerful network, the encoder could then in theory learn to approximate an exact EP projection step arbitrarily well. Unfortunately it seems likely that such a scheme would encounter convergence problems, as the encoder potentials are no longer fixed during the EP iterations, and during training the potentials may bear little resemblance to those that would result from an exact EP projection step. However, we do not discount the possibility that training such a scheme may be feasible through careful pre-training or other methods.

Finally, it remains to be seen whether the EP-SVAE has favourable results when compared to the SVAE in general settings. The results for the latent GMM were qualitatively very similar for both approaches. While the results for the EP-SVAE on the latent cycle GMM problem were significantly better than those of the SVAE, we believe the nature of local inference in that problem was particularly well suited to EP and ill-suited to MF, for reasons described in the previous section. Future work therefore may compare performance of the two approaches on a wider class of problems.

Chapter 7

Summary

In this work we introduced a new approach for performing fast, scalable inference in probabilistic graphical models with non-conjugate observation likelihoods. We built on the SVAE approach of [1] and employed the use of expectation propagation for local inference as opposed to the Mean-Field approach originally employed. Proof-of-concept experimental results were given, including one model for which existing methods gave poor results. We believe the EP-SVAE approach to be a promising addition to the set of tools available for tackling the class of problems considered here and we have proposed a number of potential avenues for future research which we believe will offer greater insight into the wider potential of the method.

Appendix A

Derivations

A.1 Overcomplete Natural Gradients

Theorem. For an overcomplete exponential family, let f and g be real-valued functions on $\mathbb{R}^d$ such that $f(\boldsymbol{\mu}(\boldsymbol{\eta})) = g(\boldsymbol{\eta})$ for all $\boldsymbol{\eta} \in \Omega$. Then,

$$g(\boldsymbol{\eta}) = g^*(F\boldsymbol{\eta})$$

for some function g^* where F maps the natural parameters to those of an equivalent minimal family, and for a dually-coupled pair $(\boldsymbol{\eta}, \boldsymbol{\mu})$, we have

$$F\nabla f(\boldsymbol{\mu}) = [\nabla^2 A^*(F\boldsymbol{\eta})]^{-1}\nabla g^*(F\boldsymbol{\eta})$$

where A^* is the cumulant function of the minimal family.

Proof. Let $\mathbf{t}_x(\mathbf{x})$ be the statistic function of our overcomplete family with components $\{t_x(\mathbf{x})_i\}$, and let $\{t^*(x)_i\}$ be any minimal basis for $\text{span}(\{t_x(\mathbf{x})_i\})$. There then must exist a matrix F such that $\mathbf{t}_x(\mathbf{x}) = F^\top \mathbf{t}^*_x(\mathbf{x})\ \forall \mathbf{x} \in \mathbb{R}^n$.

We therefore also have that $\boldsymbol{\eta}^\top \mathbf{t}_x(\mathbf{x}) = (F\boldsymbol{\eta})^\top \mathbf{t}^*_x(\mathbf{x})$, and so $\boldsymbol{\eta}^* = F\boldsymbol{\eta}$ gives us the equivalent natural parameters in this minimal representation.

Denote the equivalent overcomplete and minimal densities as $p(\mathbf{x}|\boldsymbol{\eta})$ and $p^*(\mathbf{x}|\boldsymbol{\eta}^*)$ respectively. Similarly, let $\boldsymbol{\mu} = \langle \mathbf{t}_x(\mathbf{x})\rangle_{p(\mathbf{x}|\boldsymbol{\eta})}$ and $\boldsymbol{\mu}^* = \langle \mathbf{t}^*_x(\mathbf{x})\rangle_{p^*(\mathbf{x}|\boldsymbol{\eta}^*)}$ denote the corresponding mean parameters under each density.

From the assumption that $f(\boldsymbol{\mu}(\boldsymbol{\eta})) = g(\boldsymbol{\eta})$ for all $\boldsymbol{\eta} \in \Omega$, we have

$$\begin{aligned} g(\boldsymbol{\eta}) &= f(\boldsymbol{\mu}(\boldsymbol{\eta})) \\ &= f(\langle \mathbf{t}_x(\mathbf{x}) \rangle_{p(\mathbf{x}|\boldsymbol{\eta})}) \\ &= f(\langle F^\top \mathbf{t}^*_x(\mathbf{x}) \rangle_{p(\mathbf{x}|\boldsymbol{\eta})}) \\ &= f(\langle F^\top \mathbf{t}^*_x(\mathbf{x}) \rangle_{p^*(\mathbf{x}|\boldsymbol{\eta}^*)}) \\ &= f(F^\top \boldsymbol{\mu}^*) \end{aligned} \tag{A.1}$$

We now define

$$\begin{aligned} g^*(\boldsymbol{\eta}^*) &:= f(F^\top \langle \mathbf{t}^*_x(\mathbf{x}) \rangle_{p^*(\mathbf{x}|\boldsymbol{\eta}^*)}) \\ &= f(F^\top \boldsymbol{\mu}^*) \end{aligned} \tag{A.2}$$

$$f^*(\boldsymbol{\mu}^*) := f(F^\top \boldsymbol{\mu}^*) \tag{A.3}$$

Clearly then $g(\boldsymbol{\eta}) = g^*(F\boldsymbol{\eta})$, and from Theorem 2.1.4 it follows that

$$\nabla f^*(\boldsymbol{\mu}^*) = [\nabla^2 A^*(\boldsymbol{\eta}^*)]^{-1} \nabla g^*(\boldsymbol{\eta}^*) \tag{A.4}$$

Also, applying the chain rule to (A.3) we have

$$\begin{aligned} \nabla f^*(\boldsymbol{\mu}^*) &= F \nabla f(F^\top \boldsymbol{\mu}^*) \\ &= F \nabla f(\boldsymbol{\mu}) \end{aligned}$$

and therefore

$$F \nabla f(\boldsymbol{\mu}) = [\nabla^2 A^*(F\boldsymbol{\eta})]^{-1} \nabla g^*(F\boldsymbol{\eta})$$

□

A.2 EP Moment Matching

We wish to find natural parameters $\boldsymbol{\tau}$ of our approximation $q(\mathbf{x}|\boldsymbol{\tau}) \in \mathcal{Q}$ that minimises the inclusive KL divergence between our approximation and $p(\mathbf{x})$. Assume $\mathcal{Q}$ is some tractable exponential family with sufficient statistic function $\mathbf{t}_x(\mathbf{x})$ and let us define

$$f(\boldsymbol{\tau}) := \mathrm{KL}[p(\mathbf{x}) \mid\mid q(\mathbf{x}|\boldsymbol{\tau})] \tag{A.5}$$

we differentiate with respect to $\boldsymbol{\tau}$

$$\begin{aligned}
\frac{\partial f}{\partial \boldsymbol{\tau}} &= \frac{\partial}{\partial \boldsymbol{\tau}} \left\{ \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x}|\boldsymbol{\tau})} \mathrm{d}\mathbf{x} \right\} \\
&= -\frac{\partial}{\partial \boldsymbol{\tau}} \left\{ \int p(\mathbf{x}) \log q(\mathbf{x}|\boldsymbol{\tau}) \mathrm{d}\mathbf{x} \right\} \\
&= -\frac{\partial}{\partial \boldsymbol{\tau}} \left\{ \int p(\mathbf{x}) \big(\boldsymbol{\tau}^\top \mathbf{t}_x(\mathbf{x}) - A(\boldsymbol{\tau})\big) \mathrm{d}\mathbf{x} \right\} \\
&= -\int p(\mathbf{x}) \big(\mathbf{t}_x(\mathbf{x}) - \langle \mathbf{t}_x(\mathbf{x}) \rangle_{\boldsymbol{\tau}}\big) \mathrm{d}\mathbf{x}
\end{aligned} \tag{A.6}$$

equating to zero, we we find that any stationary point must satisfy the moment matching conditions

$$\langle \mathbf{t}_x(\mathbf{x}) \rangle_p = \langle \mathbf{t}_x(\mathbf{x}) \rangle_{\boldsymbol{\tau}} \tag{A.7}$$

Finally, taking second derivatives of $f(\boldsymbol{\tau})$, we have

$$\begin{aligned}
\frac{\partial^2 f}{\partial \boldsymbol{\tau}^2} &= \frac{\partial}{\partial \boldsymbol{\tau}} \left\{ \int \mathbf{t}_x(\mathbf{x}) \exp\{\boldsymbol{\tau}^\top \mathbf{t}_x(\mathbf{x}) - A(\boldsymbol{\tau})\} \mathrm{d}\mathbf{x} \right\} \\
&= \int \big(\mathbf{t}_x(\mathbf{x}) \mathbf{t}_x(\mathbf{x})^\top - \mathbf{t}_x(\mathbf{x}) \langle \mathbf{t}_x(\mathbf{x}) \rangle_{\boldsymbol{\tau}}^\top\big) \exp\{\boldsymbol{\tau}^\top \mathbf{t}_x(\mathbf{x}) - A(\boldsymbol{\tau})\} \mathrm{d}\mathbf{x} \\
&= \langle \mathbf{t}_x(\mathbf{x}) \mathbf{t}_x(\mathbf{x})^\top \rangle_{\boldsymbol{\tau}} - \langle \mathbf{t}_x(\mathbf{x}) \rangle_{\boldsymbol{\tau}} \langle \mathbf{t}_x(\mathbf{x}) \rangle_{\boldsymbol{\tau}}^\top
\end{aligned} \tag{A.8}$$

We can recognise (A.8) as the variance of the statistic vector $\mathbf{t}_x(\mathbf{x})$ under $q(\mathbf{x}|\boldsymbol{\tau})$. This quantity must be positive-semidefinite over all $\boldsymbol{\tau} \in \Omega$, and so f is convex and any stationary point must correspond to a minimum.

A.3 EP Local Updates

Theorem. Let $p(\mathbf{x}) = \prod_i f_i(\mathbf{x}_i)$ be a product over K cliques $\{\mathbf{x}_1, ..., \mathbf{x}_K\}$. Furthermore, let the exponential family $\mathcal{Q}$ be closed under $\mathbf{x}_i$ marginalisation, and let $\mathcal{Q}_i \subset \mathcal{Q}$ be the subset of $\mathcal{Q}$ containing distributions over $\mathbf{x}_i$ only. Then,

$$\arg\min_{\tilde{f}_i(\mathbf{x}) \in \mathcal{Q}} \mathrm{KL}[f_i(\mathbf{x}_i) q_{\setminus i}(\mathbf{x}) \,||\, \tilde{f}_i(\mathbf{x}) q_{\setminus i}(\mathbf{x})] = \arg\min_{\tilde{f}_i(\mathbf{x}_i) \in \mathcal{Q}_i} \mathrm{KL}[f_i(\mathbf{x}_i) q_{\setminus i}(\mathbf{x}_i) \,||\, \tilde{f}_i(\mathbf{x}_i) q_{\setminus i}(\mathbf{x}_i)]$$

where

$$q_{\setminus i}(\mathbf{x}_i) := \int q_{\setminus i}(\mathbf{x}) \mathrm{d}\mathbf{x}_{\setminus i} \tag{A.9}$$

Proof. This is based on the approach of [19].

First we note that $q_{\setminus i}(\mathbf{x}) = \prod_{j \neq i} \tilde{f}_j(\mathbf{x}) \in \mathcal{Q}$. By assumption then $q_{\setminus i}(\mathbf{x}_i) \in \mathcal{Q}_i$, and furthermore

$$q_{\setminus i}(\mathbf{x}_{\setminus i} \mid \mathbf{x}_i) := \frac{q_{\setminus i}(\mathbf{x})}{q_{\setminus i}(\mathbf{x}_i)}$$

Note that $q_{\setminus i}(\mathbf{x}_{\setminus i} \mid \mathbf{x}_i) \in \mathcal{Q}$, and assume w.l.o.g. that it is a normalised distribution. Let us also define

$$\hat{q}(\mathbf{x}) := \tilde{f}_i(\mathbf{x}) q_{\setminus i}(\mathbf{x}) \tag{A.10}$$

where $\hat{q}(\mathbf{x}) \in \mathcal{Q}$. We also then have

$$\begin{aligned} \hat{q}(\mathbf{x}_i) &:= \int \hat{q}(\mathbf{x}) \mathrm{d}\mathbf{x}_{\setminus i} \\ \hat{q}(\mathbf{x}_{\setminus i} \mid \mathbf{x}_i) &:= \frac{\hat{q}(\mathbf{x})}{\hat{q}(\mathbf{x}_i)} \end{aligned} \tag{A.11}$$

so that $\hat{q}(\mathbf{x}_i) \in \mathcal{Q}_i$, $\hat{q}(\mathbf{x}_{\setminus i} \mid \mathbf{x}_i) \in \mathcal{Q}$. Now we can rewrite the left KL divergence

as

$$\begin{aligned}
\mathrm{KL}[f_i(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}) \mid\mid \hat{q}(\mathbf{x})] &= \int f_i(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}) \log \frac{f_i(\mathbf{x}_i)q_{\setminus i}(\mathbf{x})}{\hat{q}(\mathbf{x})} \mathrm{d}\mathbf{x} \\
&= \int f_i(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}_i) \log \frac{f_i(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}_i)}{\hat{q}(\mathbf{x}_i)} \mathrm{d}\mathbf{x}_i \\
&\quad + \int f_i(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}_{\setminus i} \mid \mathbf{x}_i) \log \frac{q_{\setminus i}(\mathbf{x}_{\setminus i} \mid \mathbf{x}_i)}{\hat{q}(\mathbf{x}_{\setminus i} \mid \mathbf{x}_i)} \mathrm{d}\mathbf{x} \\
&= \mathrm{KL}[f_i(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}_i) \mid\mid \hat{q}(\mathbf{x}_i)] \\
&\quad + \int f_i(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}_i)\mathrm{KL}[q_{\setminus i}(\mathbf{x}_{\setminus i} \mid \mathbf{x}_i) \mid\mid \hat{q}(\mathbf{x}_{\setminus i} \mid \mathbf{x}_i)] \mathrm{d}\mathbf{x}_i
\end{aligned} \tag{A.12}$$

Observe that the second term above is strictly non-negative, and can be minimised by choosing $\hat{q}(\mathbf{x}_{\setminus i} \mid \mathbf{x}_i) = q_{\setminus i}(\mathbf{x}_{\setminus i} \mid \mathbf{x}_i)$ (which is in $\mathcal{Q}$). In order to minimise the first term for $\hat{q}(\mathbf{x}_i) \in \mathcal{Q}$ we moment match as usual, but this must necessarily be a projection within $\mathcal{Q}_i$. Finally, we then have

$$\begin{aligned}
\tilde{f}_i(\mathbf{x}) &= \frac{\hat{q}(\mathbf{x})}{q_{\setminus i}(\mathbf{x})} \\
&= \frac{\hat{q}(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}_{\setminus i} \mid \mathbf{x}_i)}{q_{\setminus i}(\mathbf{x})} \\
&= \frac{\hat{q}(\mathbf{x}_i)}{q_{\setminus i}(\mathbf{x}_i)}
\end{aligned} \tag{A.13}$$

therefore, $\tilde{f}_i(\mathbf{x})$ is a function of $\mathbf{x}_i$ only, and is given by

$$\tilde{f}_i(\mathbf{x}_i) = \underset{\tilde{f}_i(\mathbf{x}_i) \in \mathcal{Q}_i}{\arg\min}\ \mathrm{KL}[f_i(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}_i) \mid\mid \tilde{f}_i(\mathbf{x}_i)q_{\setminus i}(\mathbf{x}_i)] \tag{A.14}$$

□

A.4 EP Energy Function

We prove here the assertion made in [8] that when viewed as a function of the site approximation natural parameters, stationary points of the approximate normaliser (2.53) have a one to one correspondence with the fixed points of EP, and as such (2.53) can be viewed as an energy function for the EP fixed point iterations.

We repeat the approximate normaliser here for convenience

$$\tilde{Z}(\{\boldsymbol{\tau}_i\}) := \left(\int q(\mathbf{x}) \mathrm{d}\mathbf{x} \right)^{1-K} \prod_{i=1}^{K} \left(\int f_i(\mathbf{x}) q_{\setminus i}(\mathbf{x}) \mathrm{d}\mathbf{x} \right) \tag{A.15}$$

where our (un-normalised) target distribution is

$$p(\mathbf{x}) = \prod_{i=1}^{K} f_i(\mathbf{x}) \tag{A.16}$$

and our candidate (unscaled) approximation is

$$q(\mathbf{x}|\{\boldsymbol{\tau}_i\}) = \prod_{i=1}^{K} \tilde{f}_i(\mathbf{x}|\boldsymbol{\tau}_i) \tag{A.17}$$

where each approximate factor is given by

$$\tilde{f}_i(\mathbf{x}|\boldsymbol{\tau}_i) \propto \exp\{\boldsymbol{\tau}_i^\top \mathbf{t}_x(\mathbf{x})\} \tag{A.18}$$

and $\mathbf{t}_x$ is the statistic function of our approximating family $\mathcal{Q}$.

We note first that stationary points of $\tilde{Z}(\{\boldsymbol{\tau}_i\}) \to \mathbb{R}_+$, are in one-to-one correspondence with stationary points of $\log \tilde{Z}(\{\boldsymbol{\tau}_i\})$. Let $t_x(\mathbf{x})_j$ be the j-th component of $\mathbf{t}_x(\mathbf{x})$, and $\tau_{i,j}$ the corresponding natural parameter for the i-th factor $\tilde{f}_i(\mathbf{x}|\boldsymbol{\tau}_i)$. Then

$$\frac{\partial \log \tilde{Z}}{\partial \tau_{i,j}} = \frac{\partial}{\partial \tau_{i,j}} \left\{ (1-K) \log \int q(\mathbf{x}) \mathrm{d}\mathbf{x} + \sum_{k=1}^{K} \log \int f_k(\mathbf{x}) q_{\setminus k}(\mathbf{x}) \mathrm{d}\mathbf{x} \right\} \tag{A.19}$$

$$= (1-K) \frac{\int t_x(\mathbf{x})_j q(\mathbf{x}) \mathrm{d}\mathbf{x}}{\int q(\mathbf{x}) \mathrm{d}\mathbf{x}} + \sum_{k \neq i} \frac{\int t_x(\mathbf{x})_j f_k(\mathbf{x}) q_{\setminus k}(\mathbf{x})}{\int f_k(\mathbf{x}) q_{\setminus k}(\mathbf{x}) \mathrm{d}\mathbf{x}} \tag{A.20}$$

equating to zero, then summing over i, we find

$$0 = -K\frac{\int t_x(\mathbf{x})_j q(\mathbf{x})\mathrm{d}\mathbf{x}}{\int q(\mathbf{x})\mathrm{d}\mathbf{x}} + \sum_k \frac{\int t_x(\mathbf{x})_j f_k(\mathbf{x}) q_{\setminus k}(\mathbf{x})}{\int f_k(\mathbf{x}) q_{\setminus k}(\mathbf{x})\mathrm{d}\mathbf{x}} \tag{A.21}$$

and finally, subtracting (A.20), we have

$$\frac{\int t_x(\mathbf{x})_j q(\mathbf{x})\mathrm{d}\mathbf{x}}{\int q(\mathbf{x})\mathrm{d}\mathbf{x}} = \frac{\int t_x(\mathbf{x})_j f_i(\mathbf{x}) q_{\setminus i}(\mathbf{x})}{\int f_i(\mathbf{x}) q_{\setminus i}(\mathbf{x})\mathrm{d}\mathbf{x}} \tag{A.22}$$

for all i, j. These are precisely the moment matching conditions of EP and so stationary points of objective (A.15) are fixed points of EP. To prove the reverse statement, we can see that whenever the moment matching conditions (A.22) hold, (A.20) will be zero for all i, j and so there is a one-to-one correspondence between fixed points of EP and stationary points of objective (A.15).

A.5 Latent GMM EP

Our target distribution is defined as

$$\begin{aligned}\tilde{p}(z, \mathbf{x}) &\propto p(z \mid \hat{\eta}_z) p(\mathbf{x} \mid z, \{\hat{\boldsymbol{\eta}}_{\boldsymbol{x}}(i)\}) \psi(\mathbf{x}|\mathbf{y}, \phi) \\ &= \text{Categorical}(z \mid \hat{\boldsymbol{\eta}}_{\boldsymbol{z}}) \mathcal{N}(\mathbf{x}|\hat{\boldsymbol{\eta}}_{\boldsymbol{x}}(z)) \exp\{t_{\mathbf{x}}(\mathbf{x})^\top \mathbf{r}(\mathbf{y}; \phi)\} \\ &= \exp\left\{ \sum_i \delta(z=i) \hat{\eta}_{z,i} + \sum_i \delta(z=i) \big(\hat{\boldsymbol{\eta}}_{\boldsymbol{x}}(i) + \mathbf{r}(\mathbf{y}; \phi)\big)^\top \mathbf{t}_x(\mathbf{x}) \right\}\end{aligned} \tag{A.23}$$

which we can recognise as having the functional form of a GMM with discrete mixture identity z. $\hat{\boldsymbol{\eta}}_{\boldsymbol{z}}$ denotes the expected natural parameters (under $q(\theta)$) of our categorical prior on z, and $\{\hat{\boldsymbol{\eta}}_{\boldsymbol{x}}(i)\}$ are the expected natural parameters of each prior Gaussian mixture component for $i \in \{1, ..., K\}$. We can therefore rewrite (A.23) as

$$\hat{p}(z, \mathbf{x}) = r_z \mathcal{N}\big(\mathbf{x} \mid \hat{\boldsymbol{\eta}}_{\boldsymbol{x}}(z) + \mathbf{r}(\mathbf{y}; \phi)\big) \tag{A.24}$$

where $\hat{\boldsymbol{\eta}}_{\boldsymbol{x}}(z) + \mathbf{r}(\mathbf{y}; \phi)$ are the natural parameters of the z-th posterior Gaussian component. The posterior responsibility for component i is defined as

$$r_i := \langle \delta(z=i) \rangle_{\tilde{p}(z, \mathbf{x})} \tag{A.25}$$

Performing EP inference on this model reduces to matching moments of our approximating distribution and the target distribution. In particular, we must find $\boldsymbol{\tau}_x$, $\boldsymbol{\tau}_z$ such that

$$\langle \delta(z=i) \rangle_{q(z|\boldsymbol{\tau}_z)} = r_i, \quad \forall\ i \in \{1, ..., K\} \tag{A.26}$$

$$\langle \mathbf{x} \rangle_{q(x|\boldsymbol{\tau}_x)} = \langle \mathbf{x} \rangle_{\hat{p}(z, \mathbf{x})} \tag{A.27}$$

$$\langle \mathbf{x}\mathbf{x}^\top \rangle_{q(x|\boldsymbol{\tau}_x)} = \langle \mathbf{x}\mathbf{x}^\top \rangle_{\hat{p}(z, \mathbf{x})} \tag{A.28}$$

or, equivalently, we must choose the mean-parameters of our approximating distributions to match the corresponding expectations under $\hat{p}(\mathbf{x}, z)$.

We can derive the mean parameters for our categorical approximation $q(z)$

as follows

$$
\begin{aligned}
r_i &= \langle \delta(z=i) \rangle_{\hat{p}(\mathbf{x},z)} \\
&= \sum_{z=1}^{K} \int \delta(z=i) \tilde{p}(\mathbf{x},z) \mathrm{d}\mathbf{x} \\
&\propto \sum_{z=1}^{K} \int \delta(z=i) p(z|\hat{\boldsymbol{\eta}}_{\boldsymbol{z}}) p(\mathbf{x}|z, \hat{\boldsymbol{\eta}}_{\boldsymbol{x}}(z)) \exp\{t_{\mathbf{x}}(\mathbf{x})^\top r(\mathbf{y};\phi)\} \mathrm{d}\mathbf{x} \\
&\propto \hat{\pi}_i \int \mathcal{N}(\mathbf{x}|\hat{\boldsymbol{\eta}}_x(i)) \exp\{t_{\mathbf{x}}(\mathbf{x})^\top r(\mathbf{y};\phi)\} \mathrm{d}\mathbf{x} \\
&\propto \hat{\pi}_i \frac{Z_x\big(\hat{\boldsymbol{\eta}}_{\boldsymbol{x}}(i) + r(\mathbf{y};\phi)\big)}{Z_x(\hat{\boldsymbol{\eta}}_{\boldsymbol{x}}(i))}
\end{aligned} \tag{A.29}
$$

where $Z_x(.) = e^{A_x(.)}$ is the normaliser of the Gaussian family on $\mathbf{x}$, and $\hat{\pi}_i := \langle \delta(z=i) \rangle_{\hat{\boldsymbol{\eta}}_{\boldsymbol{z}}}$ is the prior probability of the i-th cluster under our expected global natural parameters.

Let standard($\boldsymbol{\eta}_x$) return the mean, variance corresponding to Gaussian natural parameters $\boldsymbol{\eta}_x$, then, define the quantities

$$
\big\{\hat{\boldsymbol{\mu}}_i, \hat{\Sigma}_i\big\} := \mathrm{standard}\big(\hat{\boldsymbol{\eta}}_x(i) + \mathbf{r}(\mathbf{y};\phi)\big) \tag{A.30}
$$

Evaluating the quantities (A.27), (A.28) under the Gaussian mixture model (A.24) is then straightforward, by applying the law of iterated expectation and from the definition of (A.24), we have

$$
\langle \mathbf{x} \rangle_{\hat{p}(\mathbf{x},z)} = \sum_{i=1}^{K} r_i \hat{\boldsymbol{\mu}}_i \tag{A.31}
$$

$$
\langle \mathbf{x}\mathbf{x}^\top \rangle_{\hat{p}(\mathbf{x},z)} = \sum_{i=1}^{K} r_i (\hat{\Sigma}_i + \hat{\boldsymbol{\mu}}_i \hat{\boldsymbol{\mu}}_i^\top) \tag{A.32}
$$

which have the simple interpretation of the responsibility weighted mean parameters of each Gaussian mixture component.

Together, (A.29), (A.31) and (A.32) form the mean parameters of our approximation $q(\mathbf{x}, z) = q(\mathbf{x})q(z)$. The corresponding natural parameters $\boldsymbol{\tau}_x$, $\boldsymbol{\tau}_z$ can then found by performing the reverse mapping if desired.

A.6 Latent Cycle GMM EP

Our target distribution for EP, which we denote $\tilde{p}(\{z_i, \mathbf{x}_i\})$ is defined as

$$\begin{aligned}
\tilde{p}(\{z_i, \mathbf{x}_i\}) &\propto \prod_i p(z_i \mid \hat{\boldsymbol{\eta}}_{z_i}) p(\mathbf{x}_i \mid z_i, z_{j(i)}, \{\hat{\boldsymbol{\eta}}_x(l,r)\}) \psi(\mathbf{x}_i | \mathbf{y}_i, \phi) \\
&= \prod_i \text{Bernoulli}(z_i | \hat{\boldsymbol{\eta}}_{z_i}) \mathcal{N}(\mathbf{x}_i | \hat{\boldsymbol{\eta}}_x(z_i, z_{j(i)})) \psi(\mathbf{x}_i | \mathbf{y}_i, \phi) \\
&\propto \prod_i \exp\Bigg\{ \hat{\boldsymbol{\eta}}_{z_i}^\top \mathbf{t}_z(z_i) + \mathbf{r}(\mathbf{y}_i; \phi)^\top \mathbf{t}_x(\mathbf{x}_i) \\
&\quad + \sum_{(l,r)} \delta(z_i = l) \delta(z_{j(i)} = r) \big(\hat{\boldsymbol{\eta}}_x(l,r)^\top \mathbf{t}_x(\mathbf{x}_i) - A_x(\hat{\boldsymbol{\eta}}_x(l,r)) \big) \Bigg\}
\end{aligned} \tag{A.33}$$

where $\{\hat{\boldsymbol{\eta}}_x(l,r) : (l,r) \in \{0,1\}^2, \{\hat{\boldsymbol{\eta}}_{z_i} : i \in \{0, ...K-1\}\}$ are the expected global natural parameters under $q(\theta)$. Note that we are using the overcomplete Bernoulli representation here. Denote our joint factors as follows

$$\begin{aligned}
f_i(z_i) &= \exp\{\hat{\boldsymbol{\eta}}_{z_i}^\top \mathbf{t}_z(z_i)\} \\
g_i(\mathbf{x}_i, z_i, z_{j(i)}) &= \exp\Bigg\{ \sum_{(l,r)} \delta(z_i = l) \delta(z_{j(i)} = r) \big(\hat{\boldsymbol{\eta}}_x(l,r)^\top \mathbf{t}_x(\mathbf{x}_i) - A_x(\hat{\boldsymbol{\eta}}_x(l,r)) \big) \Bigg\} \\
h_i(\mathbf{x}_i) &= \exp\{\mathbf{t}_x(\mathbf{x}_i)^\top \mathbf{r}(\mathbf{y}_i; \phi)\}
\end{aligned}$$

so that our approximate distribution has the general form

$$q(\{z_i, \mathbf{x}_i\}) \propto \prod_i \tilde{f}_i(z_i) \tilde{g}_i(\mathbf{x}_i, z_i, z_{j(i)}) \tilde{h}_i(\mathbf{x}_i) \tag{A.34}$$

We shall choose our approximate posterior $q(\{z_i, \mathbf{x}_i\})$ to be a fully factorised approximation, where each factor remains in the same exponential family as the prior

$$q(\{z_i, \mathbf{x}_i\}) = \prod_{i=0}^{K-1} q(z_i) q(\mathbf{x}_i) \tag{A.35}$$

$$q(z_i) = \text{Bernoulli}(z_i) \tag{A.36}$$

$$q(\mathbf{x}_i) = \mathcal{N}(\mathbf{x}_i) \tag{A.37}$$

We observe that the singleton factors $f_i(z_i)$ and $h_i(\mathbf{x}_i)$ already lie within the

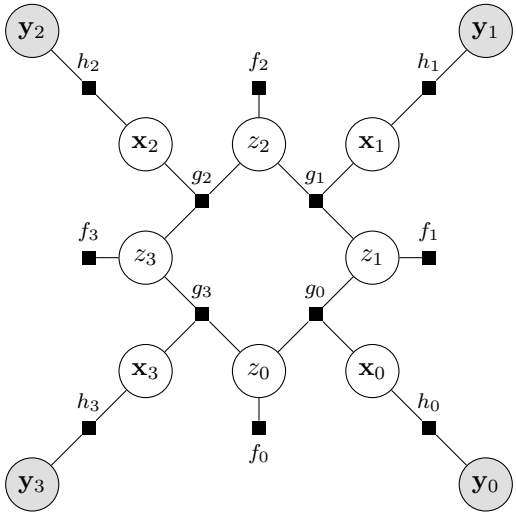

Figure A.1: Surrogate factor graph for latent cycle GMM model with $K = 4$

desired family, and so if we initialise these to the true factors, updating these in future is a no-op. The remaining factors $g_i(\mathbf{x}_i, z_i, z_{j(i)})$ require projecting to our approximating family, giving us the following forms for our approximations

$$\tilde{f}_i(z_i) = f_i(z_i) \tag{A.38}$$

$$\tilde{g}_i(\mathbf{x}_i)\tilde{g}_i(\mathbf{z}_i)\tilde{g}_i(\mathbf{z}_{j(i)}) \approx g_i(\mathbf{x}_i, z_i, z_{j(i)}) \tag{A.39}$$

$$\tilde{h}_i(\mathbf{x}_i) = h_i(\mathbf{x}_i) \tag{A.40}$$

Note that we are using overloaded notation for $\tilde{g}_i$ here. $g_i(\mathbf{x}_i, z_i, z_{j(i)})$, once projected to our approximate family, consists of a triplet of factors (one Gaussian, two Bernoulli). The intended factor should be clear from its argument.

As usual with EP, the update of $\tilde{g}_i$ is performed by moment-matching our approximating distribution with that of the tilted distribution

$$\hat{p}_i(\mathbf{x}_i, z_i, z_{j(i)}) \propto g_i(\mathbf{x}_i, z_i, z_{j(i)}) q_{\setminus i}(\mathbf{x}_i, z_i, z_{j(i)}) \tag{A.41}$$

where the cavity distribution for g_i is given by

$$q_{\setminus i}(\mathbf{x}_i, z_i, z_{j(i)}) \propto \tilde{f}_i(z_i)\tilde{f}_j(z_{j(i)}) m_{i-}(z_i) m_{i+}(z_{j(i)}) \tilde{h}_i(\mathbf{x}_i) \tag{A.42}$$

which follows from our fully factored approximation, and we have introduced

the notation

$$m_{i-}(z_i) = \tilde{g}_k(z_i) : k = (i-1) \bmod K \tag{A.43}$$

$$m_{i+}(z_{j(i)}) = \tilde{g}_k(z_{j(i)}) : k = (i+1) \bmod K \tag{A.44}$$

where the notation is intentionally suggestive of the fact that these factors are messages passed between adjacent nodes and in fact these are exactly the messages of loopy BP. The update for $\tilde{g}_i$ then involves performing the following minimisation

$$\begin{aligned}&\{\tilde{g}_i(\mathbf{x}_i), \tilde{g}_i(z_i), \tilde{g}_i(z_{j(i)})\} = \\ &\arg\min_{\tilde{g} \in \mathcal{Q}} \mathrm{KL}[g_i(\mathbf{x}_i, z_i, z_{j(i)}) q_{\setminus i}(\mathbf{x}_i, z_i, z_{j(i)}) \;||\; \tilde{g}_i(\mathbf{x}_i)\tilde{g}_i(z_i)\tilde{g}_i(z_{j(i)}) q_{\setminus i}(\mathbf{x}_i, z_i, z_{j(i)})]\end{aligned} \tag{A.45}$$

as usual, this minimisation is found by moment matching. If we use $\hat{p}_i(\mathbf{x}_i, z_i, z_{j(i)})$, $\hat{q}_i(\mathbf{x}_i, z_i, z_{j(i)})$ to denote the left and right distributions inside the KL respectively, moment matching requires

$$\langle \delta(z_i = k) \rangle_{\hat{q}_i} = \langle \delta(z_i = k) \rangle_{\hat{p}_i} \tag{A.46}$$

$$\langle \delta(z_{j(i)} = k) \rangle_{\hat{q}_i} = \langle \delta(z_{j(i)} = k) \rangle_{\hat{p}_i} \tag{A.47}$$

$$\langle \mathbf{x}_i \rangle_{\hat{q}_i} = \langle \mathbf{x}_i \rangle_{\hat{p}_i} \tag{A.48}$$

$$\langle \mathbf{x}_i \mathbf{x}_i^\top \rangle_{\hat{q}_i} = \langle \mathbf{x}_i \mathbf{x}_i^\top \rangle_{\hat{p}_i} \tag{A.49}$$

for all $k \in \{0, 1\}$, which we can interpret as choosing the mean parameters of $\hat{q}$ to match the corresponding expectations under $\hat{p}$. In order to evaluate (A.46), (A.47), we shall find it convenient to first compute the pairwise probabilities $\langle \delta(z_i = l)\delta(z_{j(i)} = r) \rangle_{\hat{p}}$ as an intermediate step and then marginalise to find the singleton statistics.

$$
\begin{aligned}
&\langle \delta(z_i = l)\delta(z_{j(i)} = r)\rangle_{\hat{p}_i} \\
&\quad \propto \sum_{z_i, z_{j(i)}} \int \delta(z_i = l)\delta(z_{j(i)} = r) q_{\setminus i}(\mathbf{x}_i, z_i, z_{j(i)}) g_i(\mathbf{x}_i, z_i, z_{j(i)}) \mathrm{d}\mathbf{x}_i \\
&\quad \propto \tilde{f}_i(l)\tilde{f}_{j(i)}(r) m_{i-}(l) m_{i+}(r) \int \mathcal{N}(\mathbf{x}_i | \hat{\boldsymbol{\eta}}_x(l, r)) \exp\{\mathbf{t}_x(\mathbf{x}_i)^\top \mathbf{r}(\mathbf{y}_i; \phi)\} \mathrm{d}\mathbf{x}_i \\
&\quad = \tilde{f}_i(l)\tilde{f}_{j(i)}(r) m_{i-}(l) m_{i+}(r) \frac{Z_x\big(\hat{\boldsymbol{\eta}}_x(l, r) + \mathbf{r}(\mathbf{y}_i; \phi)\big)}{Z_x\big(\hat{\boldsymbol{\eta}}_x(l, r)\big)} \qquad \text{(A.50)}
\end{aligned}
$$

where the constant of proportionality does not depend on z_i, $z_{j(i)}$. If we observe that (A.50) defines a distribution over z_i, $z_{j(i)}$, then the constant of proportionality can be found easily by ensuring that the statistics sum to 1. We shall find it convenient to define

$$
r_i(l, r) = \big\langle \delta(z_i = l)\delta(z_{j(i)} = r)\big\rangle_{\hat{p}_i} \qquad \text{(A.51)}
$$

Our singleton statistics are given by

$$
\begin{aligned}
\langle \delta(z_i = 0)\rangle_{\hat{p}_i} &= r_i(0, 0) + r_i(0, 1) && \text{(A.52)} \\
\langle \delta(z_i = 1)\rangle_{\hat{p}_i} &= r_i(1, 0) + r_i(1, 1) && \text{(A.53)} \\
\langle \delta(z_{j(i)} = 0)\rangle_{\hat{p}_i} &= r_i(0, 0) + r_i(1, 0) && \text{(A.54)} \\
\langle \delta(z_{j(i)} = 1)\rangle_{\hat{p}_i} &= r_i(0, 1) + r_i(1, 1) && \text{(A.55)}
\end{aligned}
$$

These statistics fully determine the distributions $\hat{q}_i(z_i)$, $\hat{q}_i(z_{j(i)})$, and so we find our approximate factors by dividing off the cavity distribution, giving

$$
\begin{aligned}
\tilde{g}_i(z_i) &= \frac{\hat{q}_i(z_i)}{\tilde{f}_i(z_i) m_{i-}(z_i)} && \text{(A.56)} \\
\tilde{g}_i(z_{j(i)}) &= \frac{\hat{q}_i(z_{j(i)})}{\tilde{f}_j(z_{j(i)}) m_{i+}(z_{j(i)})} && \text{(A.57)}
\end{aligned}
$$

With (A.56) and (A.57) then becoming messages sent to the adjacent factors.

In order to calculate the sufficient statistics (A.48), (A.49), we need to take

expectations under the marginal distribution

$$\begin{aligned}
\hat{p}_i(\mathbf{x}_i) &= \sum_{z_i, z_{j(i)}} \hat{p}_i(\mathbf{x}_i, z_i, z_{j(i)}) \\
&\propto \sum_{z_i, z_{j(i)}} \tilde{f}_i(z_i)\tilde{f}_j(z_{j(i)})m_{i-}(z_i)m_{i+}(z_{j(i)})\tilde{h}_i(\mathbf{x}_i)g_i(\mathbf{x}_i, z_i, z_{j(i)}) \\
&\propto \sum_{z_i, z_{j(i)}} \tilde{f}_i(z_i)\tilde{f}_j(z_{j(i)})m_{i-}(z_i)m_{i+}(z_{j(i)})\exp\{\mathbf{t}_x(\mathbf{x}_i)^\top \mathbf{r}(\mathbf{y}_i;\phi)\} \\
&\quad \cdot \mathcal{N}\big(\mathbf{x}_i \mid \hat{\boldsymbol{\eta}}_x(z_i, z_{j(i)})\big) \\
&\propto \sum_{z_i, z_{j(i)}} \tilde{f}_i(z_i)\tilde{f}_j(z_{j(i)})m_{i-}(z_i)m_{i+}(z_{j(i)})\frac{Z_x\big(\hat{\boldsymbol{\eta}}_x(z_i, z_{j(i)}) + \mathbf{r}(\mathbf{y}_i;\phi)\big)}{Z_x\big(\hat{\boldsymbol{\eta}}_x(z_i, z_{j(i)})\big)} \\
&\quad \cdot \mathcal{N}\big(\mathbf{x}_i \mid \hat{\boldsymbol{\eta}}_x(z_i, z_{j(i)}) + \mathbf{r}(\mathbf{y}_i;\phi)\big) \\
&\propto \sum_{z_i, z_{j(i)}} r_i(z_i, z_{j(i)})\mathcal{N}\big(\mathbf{x}_i \mid \hat{\boldsymbol{\eta}}_x(z_i, z_{j(i)}) + \mathbf{r}(\mathbf{y}_i;\phi)\big) \qquad \text{(A.58)}
\end{aligned}$$

Which we can recognise as having the form of a mixture of 4 Gaussians. Let standard($\boldsymbol{\eta}_x$) return the mean, variance corresponding to Gaussian natural parameters $\boldsymbol{\eta}_x$, then, define the quantities

$$\big\{\hat{\boldsymbol{\mu}}_i(l,r), \hat{\Sigma}_i(l,r)\big\} := \text{standard}\big(\hat{\boldsymbol{\eta}}_x(l,r) + \mathbf{r}(\mathbf{y}_i;\phi)\big) \qquad \text{(A.59)}$$

Computing the required expected statistics (A.48), (A.49) then follows the from law of iterated expectations

$$\langle \mathbf{x}_i \rangle_{\hat{p}_i} = \sum_{z_i, z_{j(i)}} r_i(z_i, z_{j(i)})\hat{\boldsymbol{\mu}}_i(z_i, z_{j(i)}) \qquad \text{(A.60)}$$

$$\langle \mathbf{x}_i \mathbf{x}_i^\top \rangle_{\hat{p}_i} = \sum_{z_i, z_{j(i)}} r_i(z_i, z_{j(i)})\Big(\hat{\Sigma}_i(z_i, z_{j(i)}) + \hat{\boldsymbol{\mu}}_i(z_i, z_{j(i)})\hat{\boldsymbol{\mu}}_i(z_i, z_{j(i)})^\top\Big) \qquad \text{(A.61)}$$

giving us the mean parameters for our approximate marginal $\hat{q}_i(\mathbf{x}_i)$, from which we can obtain our updated approximation $\tilde{g}_i(\mathbf{x}_i)$ as before

$$\tilde{g}_i(\mathbf{x}_i) = \frac{\hat{q}_i(\mathbf{x}_i)}{\tilde{h}_i(\mathbf{x}_i)} \qquad \text{(A.62)}$$

A.7 Latent Cycle GMM MF

Our surrogate model, which we denote $\tilde{p}(\theta, \{z_i, \mathbf{x}_i\})$, is given below

$$\begin{aligned}
\tilde{p}(\theta, \{z_i, \mathbf{x}_i\}) &\propto p(\theta) \prod_i p(z_i \mid \theta) p(\mathbf{x}_i \mid z_i, z_{j(i)}, \theta) \psi(\mathbf{x}_i | \mathbf{y}_i, \phi) \\
&= p(\theta) \prod_i \text{Bernoulli}(z_i | \pi_i) \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_{z_i, z_{j(i)}}, \Sigma_{z_i, z_{j(i)}}) \psi(\mathbf{x}_i | \mathbf{y}_i, \phi) \\
&\propto p(\theta) \prod_i \exp \Bigg\{ \mathbf{t}_\pi(\pi_i)^\top \mathbf{t}_z(z_i) + \mathbf{r}(\mathbf{y}_i; \phi)^\top \mathbf{t}_x(\mathbf{x}_i) \\
&\quad + \sum_{(l,r)} \delta(z_i = l) \delta(z_{j(i)} = r) \mathbf{t}_{\Sigma, \boldsymbol{\mu}}(\Sigma_{l,r}, \boldsymbol{\mu}_{l,r})^\top \mathbf{t}'_x(\mathbf{x}_i) \Bigg\}
\end{aligned}$$

where $\theta = (\{\pi_i\}, \{\Sigma_{l,r}, \boldsymbol{\mu}_{l,r}\})$ for $i \in \{0, K-1\}$, $(l, r) \in \{0, 1\}^2$ are our global parameters and $\mathbf{t}_\pi(\pi)$, $\mathbf{t}_{\Sigma, \boldsymbol{\mu}}(\Sigma, \boldsymbol{\mu})$ are the sufficient statistics functions of our conjugate priors. Note that $\mathbf{t}'_x(.)$ is the augmented sufficient statistic function introduced in section 2.1. Also note that here we are using the overcomplete Bernoulli representation so that $\mathbf{t}_z(z) = \big(\delta(z = 0), \delta(z = 1)\big)$. Let us define

$$\begin{aligned}
h(i) &:= (i - 1) \bmod K \\
j(i) &:= (i + 1) \bmod K
\end{aligned}$$

and we shall use the (somewhat relaxed) notation

$$\begin{aligned}
q_{\setminus z_i} &:= q(\theta) \Big(\prod_{j \neq i} q(z_j) \Big) \prod_k q(\mathbf{x}_k) \\
q_{\setminus x_i} &:= q(\theta) \Big(\prod_{j \neq i} q(\mathbf{x}_j) \Big) \prod_k q(z_k)
\end{aligned}$$

then from (2.19), we find the update for $q(z_i)$ as follows

$$
\begin{aligned}
q(z_i) &\propto \exp \left\langle \log \tilde{p}(\theta, \{z_i, \mathbf{x}_i\}) \right\rangle_{q_{\setminus z_i}} \\
&\propto \exp \Bigg\langle \mathbf{t}_\pi(\pi_i)^\top \mathbf{t}_z(z_i) \\
&\quad + \sum_{(l,r)} \delta(z_i = l)\delta(z_{j(i)} = r)\big(\mathbf{t}_{\Sigma,\boldsymbol{\mu}}(\Sigma_{l,r}, \boldsymbol{\mu}_{l,r})^\top \mathbf{t}'_x(\mathbf{x}_i)\big) \\
&\quad + \sum_{(l,r)} \delta(z_{h(i)} = l)\delta(z_i = r)\big(\mathbf{t}_{\Sigma,\boldsymbol{\mu}}(\Sigma_{l,r}, \boldsymbol{\mu}_{l,r})^\top \mathbf{t}'_x(\mathbf{x}_{h(i)})\big) \Bigg\rangle_{q_{\setminus z_i}} \\
&\propto \exp \Bigg\{ \langle \mathbf{t}_\pi(\pi_i) \rangle^\top_{q_{\setminus z_i}} \mathbf{t}_z(z_i) \\
&\quad + \sum_{(l,r)} \delta(z_i = l) \big\langle \delta(z_{j(i)} = r) \big\rangle_{q_{\setminus z_i}} \big\langle \mathbf{t}_{\Sigma,\boldsymbol{\mu}}(\Sigma_{l,r}, \boldsymbol{\mu}_{l,r}) \big\rangle^\top_{q_{\setminus z_i}} \big\langle \mathbf{t}'_x(\mathbf{x}_i) \big\rangle_{q_{\setminus z_i}} \\
&\quad + \sum_{(l,r)} \big\langle \delta(z_{h(i)} = l) \big\rangle_{q_{\setminus z_i}} \delta(z_i = r) \big\langle \mathbf{t}_{\Sigma,\boldsymbol{\mu}}(\Sigma_{l,r}, \boldsymbol{\mu}_{l,r}) \big\rangle^\top_{q_{\setminus z_i}} \big\langle \mathbf{t}'_x(\mathbf{x}_{h(i)}) \big\rangle_{q_{\setminus z_i}} \Bigg\} \\
&\propto \exp\{\boldsymbol{\tau}_{z_i}^\top \mathbf{t}_z(\mathbf{z}_i)\}
\end{aligned}
$$

which has the form of a Bernoulli distribution as expected, where

$$
\begin{aligned}
\tau_{z_i,0} &= \big\langle t_\pi(\pi_i)_0 \big\rangle_{q(\pi_i)} \\
&\quad + \sum_r \big\langle \delta(z_{j(i)} = r) \big\rangle_{q(z_{j(i)})} \big\langle \mathbf{t}_{\Sigma,\boldsymbol{\mu}}(\Sigma_{0,r}, \boldsymbol{\mu}_{0,r}) \big\rangle^\top_{q(\Sigma_{0,r},\boldsymbol{\mu}_{0,r})} \big\langle \mathbf{t}'_x(\mathbf{x}_i) \big\rangle_{q(\mathbf{x}_i)} \\
&\quad + \sum_l \big\langle \delta(z_{h(i)} = l) \big\rangle_{q(z_{h(i)})} \big\langle \mathbf{t}_{\Sigma,\boldsymbol{\mu}}(\Sigma_{l,0}, \boldsymbol{\mu}_{l,0}) \big\rangle^\top_{q(\Sigma_{l,0},\boldsymbol{\mu}_{l,0})} \big\langle \mathbf{t}'_x(\mathbf{x}_{h(i)}) \big\rangle_{q(\mathbf{x}_{h(i)})} \\
\tau_{z_i,1} &= \big\langle t_\pi(\pi_i)_1 \big\rangle_{q(\pi_i)} \\
&\quad + \sum_r \big\langle \delta(z_{j(i)} = r) \big\rangle_{q(z_{j(i)})} \big\langle \mathbf{t}_{\Sigma,\boldsymbol{\mu}}(\Sigma_{1,r}, \boldsymbol{\mu}_{1,r}) \big\rangle^\top_{q(\Sigma_{1,r},\boldsymbol{\mu}_{1,r})} \big\langle \mathbf{t}'_x(\mathbf{x}_i) \big\rangle_{q(\mathbf{x}_i)} \\
&\quad + \sum_l \big\langle \delta(z_{h(i)} = l) \big\rangle_{q(z_{h(i)})} \big\langle \mathbf{t}_{\Sigma,\boldsymbol{\mu}}(\Sigma_{l,1}, \boldsymbol{\mu}_{l,1}) \big\rangle^\top_{q(\Sigma_{l,1},\boldsymbol{\mu}_{l,1})} \big\langle \mathbf{t}'_x(\mathbf{x}_{h(i)}) \big\rangle_{q(\mathbf{x}_{h(i)})}
\end{aligned}
$$

Following similar reasoning for $q(\mathbf{x}_i)$,

$$
\begin{aligned}
q(\mathbf{x}_i) &\propto \exp\left\langle \log \tilde{p}(\theta, \{z_i, \mathbf{x}_i\})\right\rangle_{q_{\setminus x_i}} \\
&\propto \exp\left\langle \left(\sum_{(l,r)} \delta(z_i = l)\delta(z_{j(i)} = r)\mathbf{t}_{\Sigma,\boldsymbol{\mu}}(\Sigma_{l,r}, \boldsymbol{\mu}_{l,r}) + \mathbf{r}'(\mathbf{y}_i;\phi)\right)^\top \mathbf{t}'_x(\mathbf{x}_i)\right\rangle_{q_{\setminus x_i}} \\
&\propto \exp\{\boldsymbol{\tau}'^\top_x \mathbf{t}'_x(\mathbf{x}_i)\}
\end{aligned}
$$

where

$$
\boldsymbol{\tau}'_x = \sum_{(l,r)} \left\langle \delta(z_i = l)\right\rangle_{q(z_i)} \left\langle \delta(z_{j(i)} = r)\right\rangle_{q(z_{j(i)})} \left\langle \mathbf{t}_{\Sigma,\boldsymbol{\mu}}(\Sigma_{l,r}, \boldsymbol{\mu}_{l,r})\right\rangle_{q(\theta)} + \mathbf{r}'(\mathbf{y}_i;\phi)
$$

where we have introduced $\mathbf{r}'(\mathbf{y};\phi) = (\mathbf{r}(\mathbf{y};\phi), 0)$. We recognise $q(\mathbf{x}_i)$ as having the form of a Gaussian as expected.

Bibliography

[1] Matthew Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2946–2954. Curran Associates, Inc., 2016.

[2] Thomas Minka. *A family of algorithms for approximate Bayesian inference.* Phd thesis, MIT, 2001.

[3] Thomas P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, pages 362–369, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[4] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[5] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *J. Mach. Learn. Res.*, 14(1):1303–1347, May 2013.

[6] Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. Stochastic expectation propagation. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2323–2331. Curran Associates, Inc., 2015.

[7] Vera Gangeskar Johne. Auto-encoding with stochastic expectation propagation in latent variable models. Msc thesis, 2015.

[8] Thomas Minka. Divergence measures and message passing. Technical report, 2005.

[9] Tom Heskes, Onno Zoeter, and Wim Wiegerinck. Approximate expectation maximization. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 353–360. MIT Press, 2004.

[10] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, January 2008.

[11] Carl Doersch. Tutorial on variational autoencoders. 2016. cite arxiv:1606.05908.

[12] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222:309–368, 1922.

[13] J. Neyman. *Su un teorema concernente le cosiddette statistiche sufficienti.* Istituto Italiano degli Attuari, 1935.

[14] Matthew Johnson. Personal communication.

[15] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag, Berlin, Heidelberg, 2006.

[16] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951.

[17] Geoffrey Roeder, Yuhuai Wu, and David Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference, 2017.

[18] Irina Higgins, Loic Matthey, Xavier Glorot, Arka Pal, Benigno Uria, Charles Blundell, Shakir Mohamed, and Alexander Lerchner. Early visual concept learning with unsupervised deep learning, 2016.

[19] Matthias Seeger. Expectation propagation for exponential families. Technical report, 2007.

[20] Sekhar Tatikonda and Michael I. Jordan. Loopy belief propogation and gibbs measures, 2012.

[21] Alexander T. Ihler, John W. Fisher, and Alan S. Willsky. Message errors in belief propagation. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 609–616. MIT Press, 2005.

[22] Bruce Christianson. Reverse accumulation and attractive fixed points. *Optimization Methods and Software*, 3(4):311–326, 1994.

[23] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural Comput.*, 10(2):251–276, February 1998.

[24] T.P. Minka and John Lafferty. Expectation-Propagation for the Generative Aspect Model. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 352–359, 2002.

[25] Brendan J Frey and Anitha Kannan. Accumulator networks: Suitors of local probability propagation. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 486–492. MIT Press, 2001.

[26] Peter Dayan, Geoffrey E. Hinton, Radford M. Neal, and Richard S. Zemel. The helmholtz machine, 1995.

[27] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *CoRR*, abs/1711.00937, 2017.

[28] Yair Weiss. Belief propagation and revision in networks with loops. Technical report, Cambridge, MA, USA, 1997.

[29] Robert J. McEliece, David J. C. MacKay, and Jung-Fu Cheng. Turbo decoding as an instance of pearl's "belief propagation" algorithm. *IEEE Journal on Selected Areas in Communications*, 16(2):140–152, 1998.