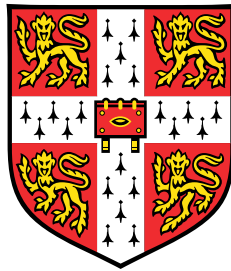


Natural gradient methods in statistics and machine learning



Jonathan Jak Sum So

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Darwin College

April 2025

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the preface and specified in the text. It is not substantially the same as any work that has already been submitted, or, is being concurrently submitted, for any degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Jonathan Jak Sum So

April 2025

Abstract

Many inference problems can be expressed as optimisations. These optimisations can often be challenging, requiring many iterations to converge using standard methods, making them computationally demanding. In this thesis, we present methods for performing such optimisations efficiently, drawing heavily on the use of natural gradient methods (Amari, 1998).

In the first main contribution chapter of this thesis, we present a novel natural-gradient-based method for optimising the parameters of probability distributions where a direct application of natural gradients would be computationally demanding. We apply this method to maximum likelihood estimation and variational inference tasks involving a number of distributions. These include: skew-elliptical distributions, which can model multivariate real-valued data with characteristics beyond those which can be captured by the Gaussian family, such as asymmetry or heavy-tailedness; elliptical copulas, which are commonly used for modelling correlation structure in high-dimensional statistics; and various mixture distributions, which represent complex probability distributions as combinations of simpler distributions, allowing for features such as multimodality, which may not be possible to represent in the component distributions alone. Our method expands the set of distribution families that can efficiently be targeted with natural gradients, and as we demonstrate, can result in significantly faster convergence than standard methods.

In the second main contribution chapter, we use a novel natural-gradient interpretation of the expectation propagation (EP) algorithm of Minka (2001) to motivate two *new* natural-gradient-based EP variants that have particularly desirable properties in *black-box* inference settings. Black-box inference methods allow practitioners to answer questions of inference without requiring expert knowledge of the underlying inference techniques, and typically place few restrictions on the model of interest. EP has several desirable computational properties in black-box settings, but existing EP variants face multiple challenges. Our new variants have several advantages over their predecessors that allow them to address these challenges. Namely, they converge faster, are easier to tune, and do not make use of debiasing estimators. By facilitating the use of EP in

such settings, our advances have the potential to reduce the computational demands of performing black-box inference.

We hope that our contributions will prove to be useful in their own right, and also that they may facilitate or inspire further advances in statistics, machine learning, or indeed any other field in which problems of inference are to be found.

Acknowledgements

I am grateful to all the members of the CBL (past and present) for helping to create such a wonderful environment for pursuing a PhD. The people are too numerous to list, but I will always cherish the time I have spent here. I would like to thank my friend and collaborator Hermann for introducing me to the world of identifiable generative models, and for accompanying me in my quest to find the perfect burger. I am also thankful to the staff at Hot Numbers on Trumpington Street for keeping me caffeinated during my PhD.

I am grateful to David and Claudia Harding for generously funding my PhD through the Harding Distinguished Postgraduate Scholars Programme. I am thankful to my parents, for giving me every opportunity in life. I am especially grateful to my supervisor, Rich, for taking me on as a student despite my unusual path into academia, for his patience as I spent several years deciding what to work on, and for his support and guidance throughout.

I would like to thank Jessica, Serena, and Miles, for keeping me on my toes during this time. I will always be thankful that the flexibility of a PhD has allowed me to spend time with you during the early years of your lives. Finally, to Antonia – thank you for everything you do and for always being there for me. This thesis would not have been possible without you.

For Antonia, Jessica, Serena and Miles.

Table of contents

Nomenclature	xv
1 Introduction	1
1.1 Thesis outline	3
1.2 Publications	4
2 Exponential families	7
2.1 Definition	7
2.2 Dual parameterisation	8
2.3 Example: multivariate normal distribution	11
2.4 Summary	12
3 Natural gradients	13
3.1 Limitations of gradient descent	13
3.2 Natural gradient descent	15
3.3 Properties	17
3.4 Exponential family natural gradients	19
3.5 Limitations	22
3.5.1 Practical difficulties	22
3.5.2 Theoretical criticisms	22
3.6 Summary	24
4 Inference	25
4.1 Maximum likelihood estimation	25
4.2 Markov chain Monte Carlo	27
4.3 Variational inference	29
4.3.1 Stochastic gradient variational Bayes	31
4.3.2 Conjugate-computation VI	32
4.3.3 Partitioned VI	33

4.3.4	Minimal conditional exponential family VI	36
4.4	Expectation propagation	37
4.4.1	EP variational problem	38
4.4.2	EP	40
4.4.3	Power EP	42
4.4.4	Double-loop EP	43
4.4.5	Stochastic natural gradient EP	46
4.4.6	EP as a black-box inference method	47
4.5	Summary	49
5	Optimising distributions with natural gradient surrogates	53
5.1	Introduction	53
5.2	Background	55
5.2.1	Natural gradient descent	56
5.2.2	Exponential family distributions	56
5.3	Surrogate natural gradient descent	57
5.3.1	Choice of surrogate	58
5.3.2	Equivalence with optimisation of f	60
5.3.3	Algorithms	60
5.4	Evaluation	62
5.4.1	Negative binomial distribution	64
5.4.2	Skew-elliptical distributions	65
5.4.3	Elliptical copulas	67
5.4.4	Mixture distributions	68
5.5	Related work	71
5.6	Future work	73
5.7	Summary	74
6	Fearless stochasticity in expectation propagation	75
6.1	Introduction	75
6.2	Background	76
6.2.1	Expectation propagation (EP)	77
6.3	Fearlessly stochastic EP algorithms	80
6.3.1	Natural gradient view of EP	80
6.3.2	EP- $\boldsymbol{\eta}$	83
6.3.3	EP- $\boldsymbol{\mu}$	84
6.3.4	Related work	86

6.4	Evaluation	88
6.5	Hyperparameter sensitivity analysis	92
6.6	Comparison with CVI	94
6.7	Future work	97
6.8	Summary	99
7	Conclusions and future work	101
	References	103
	Appendix A Gradient notation	113
	Appendix B Exponential family examples	115
B.1	Gamma distribution	115
B.2	Normal-inverse Wishart distribution	116
B.3	Mixture models	120
B.4	Exponential family natural gradients	123
	Appendix C Appendices for Chapter 5	125
C.1	Equivalence with optimisation of f	125
C.2	SNGD pseudocode details	128
C.3	Evaluation details	130
C.4	Parameter mappings	134
C.5	Additional results	134
C.6	Comparison with NGD under q	136
	Appendix D Appendices for Chapter 6	143
D.1	Conventional presentation of EP updates	143
D.2	Natural gradient view of EP	145
D.3	EP- $\boldsymbol{\eta}$ update derivation	147
D.4	EP- $\boldsymbol{\mu}$ update derivation	147
D.5	Bias of EP and EP- $\boldsymbol{\mu}$	147
D.6	Implicit geometries of EP variants	151
D.7	Computational cost analysis	154
D.8	Evaluation details	156
D.8.1	Hierarchical logistic regression	158
D.8.2	Cosmic radiation model	159
D.8.3	Neural response model	159

D.9 Pareto frontiers in wall-clock time	161
D.10 Pairwise marginals in CVI comparison	162
D.11 Cosmic radiation data	163

Nomenclature

General notation

w, z	Latent variables
x, y	Observed / given variables (or general function arguments)
$\mathcal{D}$	Set of observed / given variables
$\mathcal{F}$	Family of probability distributions
p	Probability distribution
q	Probability distribution subject to optimisation
$\tilde{q}$	Surrogate probability distribution
$p(\cdot)$	Density of (e.g.) probability distribution p
θ	Generic distribution / model parameters
Θ	Domain of generic distribution / model parameters
λ	Site parameters

Exponential family notation

η	Natural parameters
μ	Mean parameters
Ω	Natural domain
$\mathcal{M}^\circ$	Mean domain
$\nu(\cdot)$	Base measure

$s(\cdot)$	Sufficient statistic function
$A(\cdot)$	Log-partition function
$A^*(\cdot)$	Negative entropy function / convex dual of $A(\cdot)$

Acronyms / Abbreviations

BFGS	Broyden-Fletcher-Goldfarb-Shanno
CVI	Conjugate-computation variational inference
EF	Exponential family
ELBO	Evidence lower bound
EP	Expectation propagation
FIM	Fisher information matrix
GD	Gradient descent
HMC	Hamiltonian Monte Carlo
KL	Kullback Leibler
MCEF	Minimal conditional exponential family
MC	Monte Carlo
MCMC	Markov chain Monte Carlo
MLE	Maximum-likelihood estimation
MVN	Multivariate normal
NGD	Natural gradient descent
NIW	Normal-inverse Wishart
PDF	Probability density function
PMF	Probability mass function
PVI	Partitioned variational inference
SGVB	Stochastic gradient variational Bayes

SNEP Stochastic natural gradient expectation propagation

SNGD Surrogate natural gradient descent

VI Variational inference

Chapter 1

Introduction

Problems of inference are ubiquitous in scientific research, and are of fundamental importance in the fields of statistics and machine learning. Improving the computational efficiency of inference methods is therefore of significant interest. Aside from the direct savings – in time, energy, etc. – that these advancements can bring to existing applications, they can also enable new applications by expanding the set of problems that can be solved within given time or compute constraints. A notable example of this is the development of stochastic gradient optimisation methods (Robbins and Monro, 1951, Rosenblatt, 1958), which have been instrumental in the deep learning revolution of recent years. Such advancements therefore have the potential to accelerate progress across a wide range of fields, and their pursuit is the focus of this thesis.

Many inference problems can be expressed as optimisation problems. Common examples include maximum likelihood estimation (MLE), the workhorse estimation method of frequentist inference, which is concerned with point estimates of unobserved parameters or variables, and variational inference (VI), a popular class of methods for obtaining *distributional* estimates of those unobserved quantities. These optimisations can often be challenging, requiring many iterations to converge using standard methods. This means that these approaches are computationally demanding.

The natural gradient descent (NGD) method of Amari (1998) is particularly well suited for optimising the parameters of probability distributions – the typical setting in inference optimisation problems. NGD possesses several desirable theoretical properties when applied to certain inference problems, and when it can be applied, it often exhibits rapid convergence. Unfortunately, exact NGD can be performed efficiently only for a relatively small class of distribution families – most notably, minimal exponential families. Often, the distribution families encountered in inference problems live outside of this class. Copula distributions, for example, are often used in quantitative finance

for modelling correlation structure in high-dimensional data, yet do not generally live in exponential families.

When NGD is applied, it is performed with respect to a particular distribution. There is typically a natural choice for the NGD distribution in any given inference problem, i.e., the one whose parameters are being optimised. However, the NGD distribution is – to some extent – a free choice of the optimisation procedure, and it is perfectly legitimate to perform NGD with respect to a distribution other than the target of the optimisation. Exploitation of this idea is central to the developments presented in this thesis.

The first main contribution chapter of this thesis, Chapter 5, we explore the direct application of this idea as a method for optimising parameters of probability distributions, where a direct application of NGD would otherwise be computationally demanding. We apply this to MLE and VI tasks involving a number of distributions, including the negative binomial distribution, skew-elliptical distributions, elliptical copulas, and various mixture distributions. This method expands the set of distribution families that can efficiently be targeted with natural gradients, and as we demonstrate, can result in significantly faster convergence than standard methods. Furthermore, we show that several existing methods can also be interpreted as applying this same idea, including stochastic natural gradient EP (Hasenclever et al., 2017), and the natural gradient VI method of Lin et al. (2019). Although several such examples exist, we believe we are the first to propose it as a prescriptive method for solving optimisation problems. The contents of Chapter 5 are largely based on the paper “*Optimising Distributions with Natural Gradient Surrogates*” (So and Turner, 2024a) which appeared at AISTATS in 2024.

In the second main contribution chapter, Chapter 6, we show that the expectation propagation algorithm of Minka (2001) – a popular inference algorithm – can also be viewed as applying this same idea. Using this insight, we propose two new natural-gradient-based EP variants that have particularly desirable properties in *black-box* inference settings. Black-box inference methods allow practitioners to answer questions of inference without requiring expert knowledge of the underlying inference techniques, allowing them to focus on problems relevant to their expertise, such as model specification and posing questions of interest. They are considered “black-box” in the sense that their implementation is not dependent on the specifics of the model of interest, and few assumptions are typically made other than the existence of a (differentiable) potential function, which itself is often generated automatically from a user-defined probabilistic program. EP has previously been combined with Markov chain Monte

Carlo (MCMC) techniques as a way of performing black-box inference (Xu et al., 2014, Hasenclever et al., 2017, Vehtari et al., 2020). This approach has particularly desirable computational properties compared to a direct application of MCMC techniques, which is the predominant approach to black-box inference. Unfortunately, the updates of EP are not well-suited for sampled estimation, and previously proposed methods have attempted to address this issue in different ways (Xu et al., 2014, Hasenclever et al., 2017). Our new natural gradient-based EP variants are computationally more efficient than earlier approaches, and are significantly easier to tune, helping to pave the way for the effective use of EP as a black-box inference method. This has the potential to significantly expand the set of problems to which EP can be effectively applied, thereby reducing the computational demands of tackling them. The contents of Chapter 6 are largely based on the paper “*Fearless Stochasticity in Expectation Propagation*” (So and Turner, 2024b) which appeared with a spotlight award at NeurIPS in 2024.

The overall aim of this thesis is to advance methods for performing computationally efficient inference. We hope that our contributions will prove to be useful in their own right, and also that they may facilitate or inspire further advances in statistics, machine learning, or indeed any other field in which problems of inference are to be found.

1.1 Thesis outline

The remainder of this thesis is structured as follows:

Chapter 2 reviews exponential families of distributions, a set of distributional families that contains many important and well-known examples. Exponential families have several desirable properties, and are integral to all of the methods presented in later chapters.

Chapter 3 reviews the natural gradient method, including a discussion of its motivation, theoretical properties, and challenges. The natural gradient plays a central role in several developments in later chapters.

Chapter 4 reviews several classes of inference methods that are relevant to later chapters, namely, maximum likelihood estimation (MLE), Markov chain Monte Carlo (MCMC), variational inference (VI), and expectation propagation (EP).

Chapter 5 presents a novel natural-gradient-*based* method for optimising the parameters of a number of probability distributions. We apply the method to a number of classical distributions, namely, the negative binomial distribution, skew-elliptical distributions, elliptical copulas, and various mixture distributions.

Chapter 6 presents a novel natural-gradient interpretation of EP. We then use this to motivate new natural-gradient-based EP variants that are particularly well-suited to black-box inference settings, and have several advantages over their predecessors.

Chapter 7 concludes the thesis by summarising our contributions, their potential impact, and directions for future work.

1.2 Publications

Chapter 5 of this thesis is based on the paper “*Optimising Distributions with Natural Gradient Surrogates*” which appeared at AISTATS in 2024. Chapter 6 is based on the paper “*Fearless Stochasticity in Expectation Propagation*” which appeared with a spotlight award at NeurIPS in 2024. Additionally, Appendix B.2 describes a new – and, to the best of my knowledge, first – procedure for mapping from mean to natural parameters in the normal-inverse-Wishart distribution family, and is based on a short technical note that I posted on arXiv in 2024, titled “*Estimating the normal-inverse-Wishart distribution*”.

Below is a full list of publications I have worked on during my PhD, together with a summary of my individual contribution to each. Publications with content that appears in this thesis have bold titles, and an asterisk denotes main authorship.

J So*, RE Turner. (2024) **Fearless Stochasticity in Expectation Propagation**, *Advances in Neural Information Processing Systems (NeurIPS)* [Spotlight award]. I formulated the project, wrote the code, ran the experiments, and wrote the paper.

J So*, RE Turner. (2024) **Optimising Distributions with Natural Gradient Surrogates**, *International Conference on Artificial Intelligence and Statistics (AISTATS)*. I formulated the project, wrote the code, ran the experiments, and wrote the paper.

H Hälvä*, **J So**, RE Turner, A Hyvärinen. (2024) Identifiable Feature Learning for Spatial Data with Nonlinear ICA, *International Conference on Artificial Intelligence*

and Statistics (AISTATS). I helped to: formulate the project, the inference procedure, and write the code.

H Hälvä*, S Le Corff, L Lehericy, **J So**, Y Zhu, E Gassiat, A Hyvärinen. (2021) Disentangling identifiable features from noisy data with structured nonlinear ICA, *Advances in Neural Information Processing Systems (NeurIPS)*. I helped to formulate the inference procedure and write the code.

M Ashman*, **J So**, W Tebbutt, V Fortuin, M Pearce, RE Turner. (2020) Sparse Gaussian process variational autoencoders, *arXiv preprint*. I helped to formulate the project and inference procedure.

Chapter 2

Exponential families

In this chapter we give an overview of exponential families of distributions. Exponential families are a relatively rich class of probability distribution families, and contain many important and well-known examples, including: the gamma distribution, the multivariate normal (MVN) distribution, and the normal-inverse Wishart (NIW) distribution.¹ As well as containing many examples that are important in their own right, exponential families are often used as building blocks in probabilistic graphical models, in part because they have a number of convenient properties that lend themselves to mathematically elegant and computationally efficient inference algorithms. Exponential families also have several particularly desirable properties in the context of *natural gradients*, which are discussed in Chapter 3.

We begin in Section 2.1 with the definition of an exponential family. In Section 2.2, we discuss the dual parameterisation of exponential families and a number of related properties. Section 2.3 concludes the chapter with a look at the MVN family of distributions through the lens of exponential families. Much of the material in Sections 2.1 and 2.2 are based on Wainwright and Jordan (2008), and we refer the reader to that work for a far more detailed treatment.

2.1 Definition

The exponential family (EF) of distributions $\mathcal{F}$, defined by the (d -dimensional, vector-valued) sufficient statistic function $s(\cdot)$ and base measure $\nu(\cdot)$, has density function

$$p(z) = \exp(\eta^\top s(z) - A(\eta)), \quad (2.1)$$

¹We will not be particularly careful about making the distinction between a distribution and its density function unless doing so is of central importance for the point at hand.

taken with respect to $\nu(\cdot)$, where $A(\eta) = \log \int \exp(\eta^\top s(z)) \, d\nu(z)$ is known as the *log partition* function. $\eta \in \Omega$ are known as the *natural parameters*, and are used to index a specific member of $\mathcal{F}$. Ω is the set of normalisable natural parameters of $\mathcal{F}$, given by

$$\Omega = \left\{ \eta \in \mathbb{R}^d \mid \int \exp(\eta^\top s(z)) \, d\nu(z) < \infty \right\}, \quad (2.2)$$

and is known as the *natural domain* of $\mathcal{F}$. Ω is a convex set, and when it is open, the family $\mathcal{F}$ is said to be *regular*.

The natural parameters and sufficient statistics function of an exponential family are both vector valued, but we shall sometimes find it convenient however to define them as e.g. $\eta = (a, b, \dots)$, where a, b may be real, vector, or matrix-valued quantities. η should then be interpreted as the vectorised concatenation of those quantities.

When the components of $s(\cdot)$ are linearly independent, $\mathcal{F}$ is said to be *minimal*. For minimal families, each distribution in the family is associated with a unique natural parameter vector η . 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 . The exponential families we encounter in this thesis are all regular, and should be assumed to be minimal unless stated otherwise.

We will typically use e.g. p or q to refer to a distribution, and $p(\cdot)$ or $q(\cdot)$ for its density, however, see the footnote on the previous page. We will occasionally use a subscript, e.g. $p_\theta(\cdot)$, to emphasise dependence on a particular parameterisation.

2.2 Dual parameterisation

The expected sufficient statistics of a distribution with density $p(\cdot)$ with respect to base measure $\nu(\cdot)$, are given by $\mathbb{E}_{p(x)}[s(x)]$. Let $\mathcal{M}$ be the set of expected sufficient statistics that can be attained by *any* density (not necessarily in $\mathcal{F}$) with respect to $\nu(\cdot)$. That is,

$$\mathcal{M} = \left\{ \mu \in \mathbb{R}^d \mid \exists p(\cdot) : \int p(x) s(x) \, d\nu(x) = \mu \right\}. \quad (2.3)$$

All vectors in the interior of $\mathcal{M}$, $\mathcal{M}^\circ$, are realisable by a member of $\mathcal{F}$, and so $\mu \in \mathcal{M}^\circ$ provides an alternative parameterisation of $\mathcal{F}$, in which μ are known as the *mean parameters*. Similarly, $\mathcal{M}^\circ$ is known as the *mean domain* of $\mathcal{F}$. A pair η, μ such that $\mu = \mathbb{E}_{p_\eta(x)}[s(x)]$ are said to be *dually coupled*.

For minimal families, $A(\cdot)$ is strictly convex on Ω . The convex dual of $A(\cdot)$ is defined as

$$A^*(\mu) = \sup_{\eta \in \Omega} \eta^\top \mu - A(\eta), \quad (2.4)$$

and on $\mathcal{M}^\circ$, $A^*(\mu)$ is equal to the negative entropy of the member of $\mathcal{F}$ with mean parameter μ , where the entropy – taken with respect to $\nu(\cdot)$ – is defined as

$$H(\mu) = - \int p(x) \log p(x) \, d\nu(x). \quad (2.5)$$

For this reason we shall occasionally refer to $A^*(\cdot)$ as the *negative entropy function*. The log partition function can also be defined in terms of $A^*(\cdot)$ as

$$A(\eta) = \sup_{\mu \in \mathcal{M}} \mu^\top \eta - A^*(\mu). \quad (2.6)$$

The mean parameters of the member of $\mathcal{F}$ with natural parameter η can be obtained by $\mu = \nabla A(\eta)$.² For minimal families, this mapping is one-to-one, and the reverse map is given by $\eta = \nabla A^*(\mu)$. For this reason, $\nabla A(\cdot)$ and $\nabla A^*(\cdot)$ are sometimes referred to as the *forward mapping* and *backward mapping* respectively. These mappings are depicted in Figure 2.1. In this thesis, we say that a family $\mathcal{F}$ is *tractable* if both the forward and backward mappings can be evaluated efficiently. We will occasionally find it convenient to use the notation

$$\mu(\eta) = \nabla A(\eta), \quad (2.7)$$

$$\eta(\mu) = \nabla A^*(\mu), \quad (2.8)$$

to denote the forward and backward mappings, respectively. Due to our overloaded use of μ and η to denote both specific parameters and maps between them, we will use e.g. $\mu(\cdot)$ to disambiguate the latter if the intention is not clear from context. We now discuss some remarkable identities involving the Fisher information matrix of minimal exponential families.

²See Appendix A for an explanation of our gradient notation.

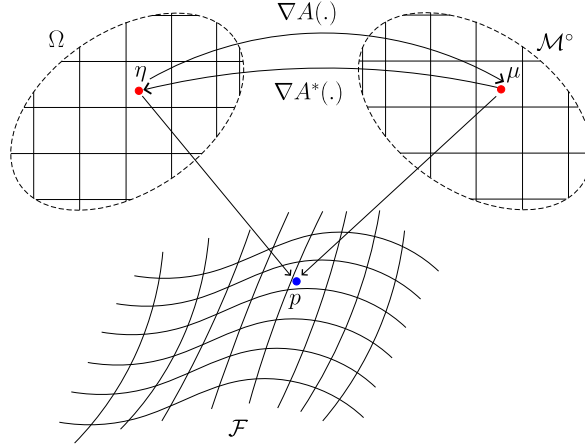


Fig. 2.1 Dual co-ordinates on a manifold of probability distributions. η and μ index the same member of $\mathcal{F}$, p , using different co-ordinate systems, with $\nabla A(\cdot)$ and $\nabla A^*(\cdot)$ mapping between the two. Ω and $\mathcal{M}^\circ$ are convex open subsets of $\mathbb{R}^d$ (here $d = 2$), and p lives on a manifold of probability distributions, also of dimension d .

The **Fisher information matrix** (FIM) of a distribution p over random variable x , with respect to parameters θ , is a measure of how much information is conveyed by x about θ , and is defined as

$$F(\theta) = \mathbb{E}_{p_\theta(x)} \left[\nabla_\theta (\log p_\theta(x)) \nabla_\theta (\log p_\theta(x))^\top \right]. \quad (2.9)$$

The Fisher information matrix appears in many places throughout statistics, but our interest here is primarily due to it playing a central role in the development of the natural gradient method of Amari (1998), which we discuss in Chapter 3. See Appendix A for an explanation of our gradient notation.

The FIM of an exponential family distribution, with respect to natural parameters η can easily be shown to satisfy $F(\eta) = \nabla^2 A(\eta)$, for $\nabla^2 A(\eta)$ the Hessian of $A(\cdot)$ at η . Furthermore, in minimal families, the FIM with respect to mean parameters μ satisfies $F(\mu) = \nabla^2 A^*(\mu)$. Using the forward and backward mappings, we also have

$$F(\eta) = \frac{\partial \mu}{\partial \eta} \quad (2.10)$$

$$F(\mu) = \frac{\partial \eta}{\partial \mu}, \quad (2.11)$$

and from the inverse function theorem,

$$F(\eta)^{-1} = \frac{\partial \eta}{\partial \mu} \quad (2.12)$$

$$F(\mu)^{-1} = \frac{\partial \mu}{\partial \eta}. \quad (2.13)$$

We shall see in Section 3.4 that these identities enable the efficient computation of natural gradients in exponential families, and are fundamental to later developments.

2.3 Example: multivariate normal distribution

We now present the exponential family of multivariate normal distributions as a worked example of the material from this Chapter. In Appendix B we also provide worked examples for the Gamma distribution, the normal-inverse Wishart distribution, and exponential family mixture models.

The multivariate normal (MVN) distribution is arguably the most fundamental distribution for modelling continuous random variables in $\mathbb{R}^d$. A multivariate normal distribution is fully characterised by its mean, $\xi \in \mathbb{R}^d$, and covariance $\Sigma \in S_{++}^d$,³ with density function given by

$$\begin{aligned} p(x) &= \frac{1}{|2\pi\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \xi)^\top \Sigma^{-1}(x - \xi)\right) \\ &=: \mathcal{N}(x; \xi, \Sigma). \end{aligned} \quad (2.14)$$

The density (2.14) can be recognised as that of an exponential family with sufficient statistic function $s(x) = (x, xx^\top)$, base measure $\nu(x) = (2\pi)^{-d/2}$, natural parameter⁴

$$\begin{aligned} \eta &= \left(\Sigma^{-1}\xi, -\frac{1}{2}\Sigma^{-1} \right) \\ &=: (\eta_1, \eta_2), \end{aligned} \quad (2.15)$$

and log partition function

$$A(\eta) = \frac{1}{4}\eta_1\eta_2^{-1}\eta_1 - \frac{1}{2}\log|\eta_2|. \quad (2.16)$$

³ S_{++}^d is the set of d -dimensional symmetric positive definite matrices.

⁴For convenience, here and in Appendix B, we define $s(\cdot)$, η and μ using tuples of matrix, vector, or scalar-valued quantities, but our notation should be interpreted as equating the left hand side with the (vectorised) concatenation of the tuple entries.

The mean parameters of the multivariate normal family are given by

$$\begin{aligned}\mu &= \left(\mathbb{E}_{p(x)}[x], \mathbb{E}_{p(x)}[xx^\top] \right) \\ &= \left(-\frac{1}{2}\eta_2^{-1}\eta_1, -\frac{1}{2}\eta_2^{-1} + \frac{1}{4}\eta_2^{-1}\eta_1\eta_1^\top\eta_2^{-1} \right) \\ &=: (\mu_1, \mu_2),\end{aligned}\tag{2.17}$$

and the backwards mapping is given by

$$\eta = \left((\mu_2 - \mu_1\mu_1^\top)^{-1}\mu_1, -\frac{1}{2}(\mu_2 - \mu_1\mu_1^\top)^{-1} \right).\tag{2.18}$$

Symmetry constraints The multivariate normal family as presented here does not appear at first to be minimal, due to the off-diagonal entries of the second-order statistic $xx^\top$ appearing twice, and hence making $s(\cdot)$ linearly dependent. However, Σ is also constrained to be symmetric, which removes the additional degrees of freedom introduced by the duplicated statistics. A more explicit minimal form can be recovered by defining the second-order statistics as $\text{vech}(xx^\top)$ and adjusting the natural parameters accordingly, where $\text{vech}(\cdot)$ is the half-vectorisation transformation for symmetric matrices. Dealing with this form can be unwieldy however, and so we stick with the more conventional choice of working with symmetric constraints, but note that extra care must be taken when performing computations that assume a minimal representation.

2.4 Summary

In this chapter we gave a brief overview of exponential family distributions, and discussed a number of properties relating to the duality between the natural and mean parameters of a given family. In the next chapter we introduce the *natural gradient method*, and show how the properties discussed in this chapter can be facilitate the efficient computation of natural gradients in exponential families. These properties will later be used extensively in our main contributions chapters.

Chapter 3

Natural gradients

In this chapter, we review the natural gradient method of Amari (1998), which is used to perform optimisations with respect to the parameters of probability distributions, and is central to the developments in this thesis.

The natural gradient method is rooted in the field of information geometry, which marries techniques from differential geometry with probability and statistics. The field of differential geometry is concerned with the study of smooth manifolds, and in information geometry, the focus is on manifolds of probability distributions. While an introduction to differential geometry lies beyond the scope of this thesis, fortunately (for the author), a background in this area is not required for using or understanding the natural gradient method. This chapter aims to introduce the method to readers with little or no knowledge of differential geometry.

We begin the chapter in Section 3.1 by introducing the problem setting, and motivating the natural gradient method by illustrating some limitations of gradient descent, the archetypal optimisation method. In Section 3.2 we use the limitations of gradient descent to motivate and derive the natural gradient method. In Section 3.3, we discuss several theoretical properties of the method, and their relevance for optimisation. In Section 3.4, we discuss the use of natural gradients in exponential families. We then conclude the chapter in Section 3.5 by taking a critical look at natural gradients and challenging some commonly held misconceptions about their use.

3.1 Limitations of gradient descent

In mathematical optimisation problems, the goal is to find parameters θ^* that minimise a real-valued function $f : \Theta \rightarrow \mathbb{R}$, known as the *objective function*. That is, we wish to

find θ^* such that

$$f(\theta^*) = \min_{\theta \in \Theta} f(\theta), \quad (3.1)$$

where Θ is the domain of $f(\cdot)$. For non-convex objective functions, we are typically concerned with finding *local* minimisers of $f(\cdot)$. When $\Theta = \mathbb{R}^d$, and $f(\cdot)$ is differentiable, the archetypal optimisation method is that of *gradient descent* (GD). Gradient descent updates a candidate solution in an iterative manner, by repeatedly moving in the direction in parameter space that gives the greatest decrease in $f(\cdot)$ for an infinitesimally small step size, which is equivalent to following the direction of the negative gradient. That is, it performs the sequence of updates

$$\theta \leftarrow \theta - \epsilon_t \nabla f(\theta), \quad (3.2)$$

where ϵ_t is the learning rate, or step size, at time t . Under certain assumptions on $f(\cdot)$ and the sequence $\epsilon_1, \dots, \epsilon_t$, the sequence (3.2) is guaranteed to converge to a *local* optimum of $f(\cdot)$ (Robbins and Monro, 1951). Another way to motivate the gradient descent update, is that it can be viewed as minimising a linear approximation of the objective function, with a penalty imposed on the squared Euclidean distance from the current iterate. That is,

$$\theta - \epsilon_t \nabla f(\theta) = \arg \min_{\theta'} \left[\underbrace{f(\theta) + (\theta' - \theta)^\top \nabla f(\theta)}_{\text{linearised objective}} + \underbrace{(2\epsilon)^{-1} \|\theta - \theta'\|^2}_{\text{distance penalty}} \right]. \quad (3.3)$$

In (3.3), ϵ^{-1} can be seen as a regularisation parameter, controlling the extent to which the parameter change is penalised. The distance penalty term ensures that the update does not stray too far from the current iterate, thereby maintaining proximity to the region where the linear approximation to $f(\cdot)$ is most accurate. This view of gradient descent helps to illustrate one of its main shortcomings: there is an implicit assumption that the Euclidean distance is a meaningful measure of dissimilarity between parameters, and in many cases this is not justified. For example, if $\theta = (\xi, \sigma^2)$, where ξ and σ^2 are the mean and variance of a univariate normal distribution over a variable measured in units u , the parameters ξ and σ^2 have units u and u^2 respectively, and so the Euclidean distance between $\theta_a = (\xi_a, \sigma_a^2)$ and $\theta_b = (\xi_b, \sigma_b^2)$, is given by $\sqrt{(\xi_a - \xi_b)^2 + (\sigma_a^2 - \sigma_b^2)}$, which is dimensionally inconsistent.

Of more practical concern, is that $f(\cdot)$ will typically be more sensitive to changes in some directions of parameter space than others, and those directions may be

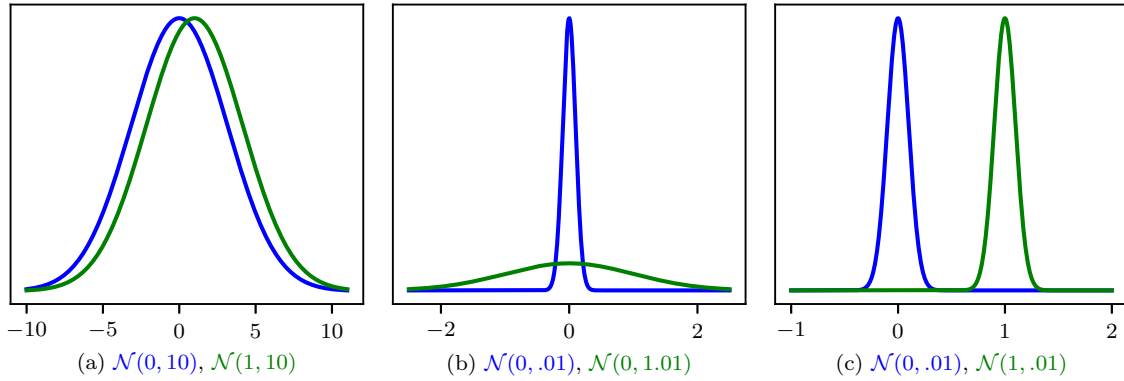


Fig. 3.1 Illustration of the pitfalls of using Euclidean distance in parameter space. Any reasonable dissimilarity measure should reflect the extent to which the two distributions are more similar in (a) than they are in (b) or (c), but the Euclidean distance in parameter space is the same in each case when using a mean-variance parameterisation. This figure is similar to one that appeared in Khan and Nielsen (2018), and is based on an example given by Hoffman et al. (2013).

location dependent. In statistical optimisation problems, the objective function $f(\cdot)$ often involves some divergence measure between a model distribution and some other distribution, or vice versa. As such, $f(\cdot)$ will typically be more sensitive to large changes in the distribution than smaller ones, and so we would like an optimisation algorithm move slower in directions in parameter space that effect large changes in distribution than those that effect small ones. Unfortunately the Euclidean distance, and therefore gradient descent, penalises all directions in parameter space equally. This limitation is illustrated in Figure 3.1. The practical consequence is that the learning rate of gradient descent must be set sufficiently small to make stable progress along the most sensitive directions in parameter space, which can result in unnecessarily slow progress along others.

The form of (3.3) suggests that an improved update could be derived by using a different measure of proximity than the Euclidean distance, and one which correlates more directly with changes in *distribution*. This is precisely what the natural gradient method of Amari (1998) gives us.

3.2 Natural gradient descent

We saw in the previous section that the Euclidean distance is typically not an appropriate measure of dissimilarity between parameters of probability distributions. An

alternative quantity, that is often used to measure dissimilarity between distributions, is the *Kullback-Leibler* (KL) divergence.

The **Kullback-Leibler (KL) divergence** from one continuous distribution, p_a , to another, p_b , both over $x \in \mathcal{X}$, is given by

$$\text{KL}[p_a \parallel p_b] = \int_{\mathcal{X}} p_a(x) \log \frac{p_a(x)}{p_b(x)} d\nu(x), \quad (3.4)$$

where $p_a(\cdot)$, $p_b(\cdot)$ are densities of p_a , p_b respectively, with respect to measure $\nu(\cdot)$. The KL divergence is not formally a distance as it is not symmetric, but nevertheless it is a widely used measure of dissimilarity between distributions. It is strictly non-negative, and equal to zero only when $p_a(x) = p_b(x) \forall x \in \mathcal{X}$. The KL divergence for discrete distributions is defined similarly, with integration replaced by summation.

If θ are the parameters of some probability distribution q_θ , then, motivated by the form of the gradient descent update, we could consider replacing the Euclidean distance in (3.3) with the KL divergence from q_θ to $q_{\theta'}$, giving

$$\theta \leftarrow \arg \min_{\theta'} \left[\underbrace{f(\theta) + (\theta' - \theta)^\top \nabla f(\theta)}_{\text{linearised objective}} + \underbrace{\epsilon_t^{-1} \text{KL}[q_\theta \parallel q_{\theta'}]}_{\text{KL penalty}} \right]. \quad (3.5)$$

This (hypothetical) update would penalise changes in the parameters to the extent that they result in changes in the distribution q_θ as measured by KL divergence. Unfortunately, this update cannot be computed in closed form in general, but we can recover a closed-form solution if we instead use a second order Taylor approximation of $\text{KL}[q_\theta \parallel q_{\theta'}]$ around θ , giving the *natural gradient update*

$$\begin{aligned} \theta &\leftarrow \arg \min_{\theta'} \left[\underbrace{f(\theta) + (\theta' - \theta)^\top \nabla f(\theta)}_{\text{linearised objective}} + \underbrace{(2\epsilon_t)^{-1} (\theta - \theta')^\top F(\theta) (\theta - \theta')}_{\text{approximate KL penalty}} \right] \\ &= \theta - \epsilon_t F(\theta)^{-1} \nabla f(\theta), \end{aligned} \quad (3.6)$$

where $F(\theta)$ is the Fisher information matrix (FIM) of q with respect to θ , which we restate below for convenience.

$$F(\theta) = \mathbb{E}_{q_\theta(x)} \left[\nabla_\theta (\log q_\theta(x)) \nabla_\theta (\log q_\theta(x))^\top \right] \quad (3.7)$$

Note in particular how the FIM depends on both the distribution *and* its parameterisation. The *natural gradient* is defined as

$$\tilde{\nabla} f(\theta) = F(\theta)^{-1} \nabla f(\theta), \quad (3.8)$$

and so using this notation, the natural gradient update is given by

$$\theta \leftarrow \theta - \epsilon_t \tilde{\nabla} f(\theta). \quad (3.9)$$

Note that, if (3.5) had instead been formulated using the “opposite” divergence, $\text{KL}[q_{\theta'} \parallel q_{\theta}]$, the exact solution would typically be different, but after taking a second-order Taylor approximation, the resulting update is the same as (3.9). However, there is an important special case in which the *exact* solution to (3.5) is yielded by the natural gradient, in which case the direction of KL divergence is important – namely, when θ are either natural or mean parameters of an exponential family, which we discuss in the next section.

Ollivier et al. (2017) showed that the NGD update moves in the direction in parameter space that gives the greatest decrease in the objective within an infinitesimally small KL ball centred around the current iterate. That view follows naturally from the one presented here, but is slightly less revealing (in our opinion), as it offers no interpretation of the update magnitude.

3.3 Properties

In the previous sections, we motivated the NGD update as the solution to an auxiliary optimisation problem, with an objective that consists of a linear approximation to the objective function, $f(\cdot)$, plus a penalty term that approximates (to second order) the KL divergence from the current iterate to the next (or vice-versa). We also argued that the KL divergence is a more appropriate measure of dissimilarity between distributions than the Euclidean distance in their parameters, as is implicitly relied upon by gradient descent. In this section we briefly summarise some other important properties of NGD.

Parameterisation invariance The NGD update follows, locally, the direction of steepest descent on a *statistical manifold* – a manifold of probability distributions – where geometric quantities such as distance and angles are derived from the Fisher information metric. As these geometric quantities are intrinsic to the manifold, they

are not dependent on any particular parameterisation or coordinate system. One consequence of this, is that the natural gradient update is *locally* invariant to arbitrary reparameterisations – where local refers to the fact that this is only strictly true for infinitesimally small step sizes. A further corollary however, is that this implies *global* invariance to *affine* reparameterisations. This dispenses with one of the well-known pathologies of gradient descent, whose performance is heavily affected by even element-wise rescaling of the parameters. However, see Section 3.5 for some nuance on this apparent advantage.

Hessian approximation in maximum-likelihood settings When the optimisation problem being tackled is one of maximum-likelihood estimation – introduced in Section 4.1 – the NGD update can be shown to approach Newton’s method near the optimum under certain conditions (Martens, 2020). More generally, the Fisher information matrix can be viewed as a *positive-definite* approximation to the Hessian of the maximum-likelihood objective, even when far from the optimum (Martens, 2020). This allows NGD to mitigate one of the main problems of Newton’s method when applied to non-convex problems, in which the Hessian may not be positive definite everywhere, potentially resulting in update directions that *increase* the objective function.

Fisher efficiency in online estimation settings In online estimation settings, the goal is to estimate parameters from a stream of observed datapoints, in such a way that the parameter updates at each step can depend only on the previous parameters and the current observation. When the parameters in this setting are updated by following the natural gradient of a single-observation log-likelihood loss, the resulting estimates can be shown to asymptotically achieve the Cramér-Rao lower bound as the number of datapoints tends to infinity. In other words, the online natural gradient estimator achieves, asymptotically, the lowest possible mean-squared error achievable by any unbiased *full-batch* estimator (Amari, 1998). This result has particular relevance in the modern deep-learning era, where datasets are typically processed in stochastic mini-batches, and are frequently large enough that no recycling of data is necessary, effectively recreating (a finite truncation of) the online setting.

Single-step convergence in exponential families In exponential families of distributions – which we introduced in Chapter 2 – the NGD update has the remarkable property that it will reach the optimum in a single step in certain parameterisations (and when $\epsilon_t = 1$), when the objective function consists of a KL divergence involving the distribution being optimised. We discuss this property more in Section 3.4.

3.4 Exponential family natural gradients

In this section, we discuss a number of remarkable properties related to the use of natural gradients in exponential families. The first is a direct consequence of the Fisher identities discussed in Section 2.2. Let η and μ be the natural and mean parameters, respectively, of some exponential family (EF) distribution. Let f_η and f_μ be functions related by $f_\eta = f_\mu \circ \mu$. That is, they are reparameterisations of the same function, in terms of natural, and mean parameters, respectively. It is a remarkable property of exponential families that the natural gradients of f_η and f_μ with respect to η and μ , respectively, are given by¹

$$\tilde{\nabla} f_\eta(\eta) = \nabla f_\mu(\mu(\eta)) \quad (3.10)$$

$$\tilde{\nabla} f_\mu(\mu) = \nabla f_\eta(\eta(\mu)). \quad (3.11)$$

That is, the natural gradient in one parameterisation is simply given by the flat gradient with respect to the other (Malagò and Pistone, 2015). Provided that we have an efficient way of converting between natural and mean parameters, this allows us to compute the natural gradient without explicitly instantiating or inverting $F(\theta)$, e.g. by using automatic differentiation through the forward or backward mappings.

Further special properties apply when NGD is applied to objectives involving a KL divergence (or related quantity) with respect to either the natural or mean parameters of an EF distribution, as the propositions below state. A proof of these propositions can be found in Appendix B.4.

Proposition 1. *For EF q with mean parameters $\mu \in \mathcal{M}^\circ$, and mean domain $\mathcal{M}^\circ$, let*

$$f(\mu) = -\mathbb{E}_{p^*(x)}[\log q_\mu(x)] + c \quad (3.12)$$

where p^* is any distribution with $\text{supp}[p^*] \subseteq \text{supp}[q]$, and c is constant with respect to μ . Then, $\forall \mu \in \mathcal{M}^\circ$,

$$\mu - \tilde{\nabla} f(\mu) = \arg \min_{\mu' \in \mathcal{M}^\circ} f(\mu'). \quad (3.13)$$

Proposition 2. *For EF q with natural parameters $\eta \in \Omega$, and natural domain Ω , let*

$$f(\eta) = \text{KL}[q_\eta \parallel p^*] + c \quad (3.14)$$

¹Again, see Appendix A for an explanation of our gradient notation.

where p^* is from the same EF as q . Then, $\forall \eta \in \Omega$,

$$\eta - \tilde{\nabla} f(\eta) = \arg \min_{\eta' \in \Omega} f(\eta'). \quad (3.15)$$

Proposition 1 says that (undamped) NGD with respect to the mean parameters of EF q will converge in a single step when the objective is a cross entropy from *any* distribution p^* to q_μ plus a constant that does not depend on μ . Two straightforward corollaries of this are that NGD also has single step convergence when $f(\mu) = \text{KL}[p^* \parallel q_\mu]$ (a forward KL divergence), or when $f(\mu) = -\sum_{i=1}^n \log q_\mu(x_i)$ (an MLE objective). Typically, when performing MLE of an EF distribution, we simply think of it as “solving” for the parameters, or “moment matching”, but proposition (1) shows that this is equivalent to performing (undamped) NGD with respect to μ . This link between NGD and moment matching has been made before by Bouchard-Côté et al. (2024).

Proposition 2 is an *almost* symmetric result regarding NGD with respect to the natural parameters of q , however it requires that p^* is from the same family, which is not the case for the corresponding result with respect to mean parameters. This result is similar to one given by Sato (2001) in the context of variational message passing.

The use of natural gradients in exponential families more broadly has been discussed extensively in the literature. For example: by Malagò et al. (2011) in their work on stochastic relaxation problems; by Ollivier et al. (2017), in the context of their information-geometric optimization (IGO) framework; by Kakade (2001), in their natural policy gradient framework for reinforcement learning; and by Khan et al. (2018), in the context of (variational) Bayesian deep learning, as well as variational optimisation more generally.

Equivalence with mirror descent In Section 3.2 we derived the NGD update by replacing the Euclidean penalty of gradient descent with an *approximate* KL-divergence penalty. There is an important special case however in which the exact solution to (3.5) can be obtained using the natural gradient update. Let θ be the natural parameters of a minimal exponential family, so that (3.5), for an objective function $f_\eta(\cdot)$, becomes

$$\begin{aligned} \eta &\leftarrow \arg \min_{\eta'} \left[f_\eta(\eta) + (\eta' - \eta)^\top \nabla f_\eta(\eta) + \epsilon_t^{-1} \text{KL}[q_\eta \parallel q_{\eta'}] \right] \\ &= \arg \min_{\eta'} \left[\eta'^\top \nabla f_\eta(\eta) + \epsilon_t^{-1} \text{KL}[q_\eta \parallel q_{\eta'}] \right]. \end{aligned} \quad (3.16)$$

Note that $\text{KL}[q_\eta \parallel q_{\eta'}]$ is strictly convex in η' , and is in fact the Bregman divergence between η' and η induced by $A(\eta)$, defined as

$$\begin{aligned} B_A(\eta', \eta) &= A(\eta') - A(\eta) - \nabla A(\eta)(\eta' - \eta) \\ &= \text{KL}[q_\eta \parallel q_{\eta'}]. \end{aligned} \quad (3.17)$$

(3.16) can then equivalently be stated as

$$\eta \leftarrow \arg \min_{\eta'} \left[\eta'^\top \nabla f_\eta(\eta) + \epsilon_t^{-1} B_A(\eta', \eta) \right], \quad (3.18)$$

which is known as *mirror descent* in η with respect to the proximity function B_A (Nemirovsky and Yudin, 1983). By taking derivatives of the minimand on the right-hand side of (3.18) with respect to η' , we find that at the minima, we must have

$$\begin{aligned} 0 &= \nabla f_\eta(\eta) + \epsilon_t^{-1} (\nabla A(\eta') - \nabla A(\eta)) \Rightarrow \\ \nabla A(\eta') &= \nabla A(\eta) - \epsilon_t \nabla f_\eta(\eta) \Leftrightarrow \\ \mu' &= \mu - \epsilon_t \nabla f_\eta(\eta). \end{aligned} \quad (3.19)$$

However, from (3.11), we also have that

$$\begin{aligned} \mu' &= \mu - \epsilon_t \nabla f_\eta(\eta) \\ &= \mu - \epsilon_t \tilde{\nabla} f_\mu(\mu), \end{aligned} \quad (3.20)$$

and so we have that the update defined by (3.16), and hence that of mirror descent with respect to B_A , is given by the natural gradient update with respect to μ . Furthermore, by almost identical reasoning, we can show that

$$\begin{aligned} \mu &\leftarrow \arg \min_{\mu'} \left[f_\mu(\mu) + (\mu' - \mu)^\top \nabla f_\mu(\mu) + \epsilon_t^{-1} \text{KL}[q_{\mu'} \parallel q_\mu] \right] \\ &= \arg \min_{\mu'} \left[\mu'^\top \nabla f_\mu(\mu) + \epsilon_t^{-1} B_{A^*}(\mu', \mu) \right], \end{aligned} \quad (3.21)$$

is solved when $\eta' = \eta - \epsilon_t \tilde{\nabla} f_\eta(\eta)$, and so we have the following remarkable result. Mirror descent in η with proximity function $\text{KL}[\eta \parallel \eta']$ is equivalent to NGD in μ , and conversely, mirror descent in μ with proximity function $\text{KL}[\mu' \parallel \mu]$ is equivalent to NGD in η . This result is due to Raskutti and Mukherjee (2015), and was also further articulated by Khan and Rue (2023).

3.5 Limitations

In this chapter we have discussed properties of NGD that can make it a desirable choice of method for performing optimisations involving probability distribution parameters. In this section we take a more critical look at NGD. First, we discuss the practical difficulties in applying NGD – often seen as the main obstacle to more widespread adoption of the method. We also, however, highlight several weaknesses in some of the theoretical justifications for its use – challenging the often-held view that NGD is the “optimal” optimisation method, if only it could be applied.

3.5.1 Practical difficulties

Perhaps the most significant limitation of the natural gradient method, is that its use, in the general case, requires the ability to compute and invert the Fisher information matrix (or at least, to compute inverse-vector products). In many cases, the Fisher matrix may not have a closed-form solution, and may require estimation using numerical or sampling-based methods. This problem is compounded by taking the inverse, potentially amplifying any errors, and causing sampling-based estimates to become biased. Furthermore, the cost of computing inverse-vector products is $O(m^3)$, for m the number of parameters. For even moderately large m , this cost can be prohibitively expensive. In Chapter 5, we present a novel method for sidestepping these practical difficulties for a variety of different probability distributions.

We stated in Section 3.3 that the NGD locally follows the direction of steepest descent on a statistical manifold. Unfortunately, the local qualifier means that this only holds for an infinitesimally small step size. For any finite step size, the NGD update can potentially move entirely off the manifold, which can be corrected using techniques such as backtracking. Lin et al. (2020) introduce a method, closely related to natural gradient descent, that ensures matrix-valued parameters remain positive definite, which ensures that updates to Gaussian parameters, for example, remain on the statistical manifold.

3.5.2 Theoretical criticisms

The first property we discussed in Section 3.3 was that of parameterisation invariance – global invariance, in the case of affine reparameterisations, and local invariance, in the general case. The local invariance property means that this invariance only holds for *infinitesimally small* step sizes. While this may be of theoretical interest, it is of

little relevance to its use as a practical optimisation method, which must deal with finite – ideally large – step sizes. In practice, the performance of NGD can in fact be heavily dependent on the particular parameterisation chosen. A related point, is that the direction of steepest descent on the manifold is by definition a very local property – it tells us little about the rate of function decrease over large straight-line (geodesic) distances on the manifold, let alone in parameter space.

A more fundamental question on the issue of parameterisation invariance, is how desirable this property is in the first place. It is well-known that the performance of gradient descent is heavily dependent on the relative scale of the parameters being optimised, and this is often used as a justification for seeking parameterisation invariant methods. However, this implicitly assumes that such an invariant method is invariably *good* – it would be simple enough to construct an optimisation method that is invariant to affine reparameterisations where the invariant performance is arbitrarily bad. Going further, in the age of modern deep learning, in which architectures and techniques have been developed specifically to work well with simple first-order optimisers, it is entirely conceivable that parameterisation invariance could worsen optimisation performance relative to simple first-order methods. Put another way, there is no reason a priori to expect an invariant optimisation method to be more effective than a non-invariant method operating in an appropriate parameterisation. Furthermore, Kerekes et al. (2021) argue that, in the context of deep learning, the parameter dependence of gradient-based optimisers plays an important role in their generalisation performance, and so the (approximate) parameterisation invariance of NGD can actually hinder generalisation.

Another criticism concerns the connection with Newton’s method. In the maximum-likelihood estimation setting, the Fisher and Hessian only coincide at the optimum – when far away from the optimum, they differ by a term that, while not guaranteed to be positive semidefinite, may nonetheless account for substantial curvature (Dauphin et al., 2024). This criticism may be particularly pertinent in deep learning settings, in which it is typical to terminate the optimisation procedure before fully converging on the training objective. Another obvious limitation of this property is that it only applies in the maximum-likelihood estimation setting, or, slightly more generally, when minimising a forward KL divergence. When minimising an alternative objective, the Fisher may bear little resemblance to the Hessian. For example, in the case of a *reverse* KL divergence objective, such as is used in variational inference (see Section 4.3), the Fisher can be substantially different from the objective curvature, even at the optimum (Tang and Ranganath, 2019).

3.6 Summary

In this chapter we gave a brief overview the natural gradient method. Whilst not a panacea, natural gradients have been applied effectively in a number of settings. The main obstacle to more widespread adoption is that the natural gradient update requires the computation of inverse-vector products using the Fisher information matrix. For general distributions, when the number of parameters is even moderately large, this computation is intractable. However, we showed that minimal exponential families are a special case, and have properties that allow us to compute the natural gradient efficiently. In later chapters we exploit these properties in order to optimise the parameters of distributions that are *not* from exponential families. First, we review a number of inference methods that are relevant to later chapters, several of which make (explicit or implicit) use of exponential family natural gradients.

Chapter 4

Inference

In this chapter we review a number of inference methods that have particular relevance for this thesis. We begin in Section 4.1 with a brief introduction to maximum-likelihood estimation – the workhorse estimation method of frequentist statistics – which is used to obtain point estimates of model parameters from observed data. We then review a number of Bayesian inference methods, which are concerned instead with *distributional* estimates of model parameters (or unknown quantities more generally). We begin our review of Bayesian methods in Section 4.2 with a brief introduction to Markov chain Monte Carlo methods, which are often viewed as the gold standard among Bayesian inference methods in terms of accuracy. We then take a more detailed look at variational inference (VI) in Section 4.3, and expectation propagation (EP) in Section 4.4, two popular families of methods for finding tractable parametric approximations to a probability distribution of interest. We shall relate the various methods to exponential families and natural gradients throughout.

At the end of the chapter, in Section 4.5, we conclude by highlighting connections between some of these methods, and discussing their relative strengths and weaknesses. We will also preview the contributions of subsequent chapters, and discuss how they affect the landscape of inference methods.

4.1 Maximum likelihood estimation

Frequentist statistics is concerned with making inferences about fixed, but unknown, quantities from data. The unknown quantities are typically the parameters of a statistical model, which we denote θ^* . In this setting, observed data $\mathcal{D}$ are assumed to be drawn from a statistical model. The true parameters, θ^* , are unknown, and so the goal is to learn about θ^* from $\mathcal{D}$.

We will use the notation θ (or variations of it) to denote generic parameters of some model or distribution that we wish to optimise. When dealing with a particular distribution or model family, we will sometimes use alternative notation, specific to that family. For example, we will typically use η or μ to denote the natural or mean parameters of an exponential family, respectively.

One of the most fundamental objects in frequentist inference is the *likelihood*. This is the density of $\mathcal{D}$ given θ , but considered as a function of θ , with $\mathcal{D}$ held fixed. Importantly, the likelihood is *not* a probability density over θ . The likelihood is the object through which all inferences about θ^* are made in frequentist statistics. *Maximum likelihood estimation* (MLE) refers to the general approach of finding the parameters that maximise the probability of the observed data. That is, the MLE estimate is given by

$$\theta_{\text{mle}} = \arg \max_{\theta} \log q(\mathcal{D} \mid \theta), \quad (4.1)$$

where $q(\mathcal{D} \mid \theta)$ is the likelihood of the assumed model.

We will typically use p to denote a distribution and $p(\cdot)$ for its density (or mass function), unless the distribution (or its density) is the subject of an optimisation (as above), in which case we will use q or $q(\cdot)$ accordingly. We may also occasionally use, e.g. p_{θ} or $p_{\theta}(\cdot)$, when we wish to highlight a dependence on parameters.

Maximum likelihood estimation is widely applicable. In its most general form, the only strict requirement is that the likelihood is known and can be easily evaluated, although the conditions required for it to yield sensible answers are rather stricter. We note in passing that estimation methods also exist for settings in which the likelihood cannot be evaluated, for example, when it is only known up to a normalising constant (Hyvärinen, 2005).

In a minimal exponential family, the MLE estimate of the natural parameters η having observed $\mathcal{D} = \{x_1, \dots, x_n\}$, is found by performing the backwards mapping on the empirical average sufficient statistics of the observed data, given by

$$\eta_{\text{mle}} = \nabla A^* \left(\frac{1}{n} \sum_{i=1}^n s(x_i) \right). \quad (4.2)$$

Solving for η_{mle} is also equivalent to performing a natural gradient ascent step (with learning rate equal to one) in a *mean* parameterisation of the likelihood, as shown by Bouchard-Côté et al. (2024) and in Section 3.4.

In some special cases, such as that described above, the MLE estimate of a distribution’s parameters has a closed form solution, or can otherwise be efficiently evaluated. A MLE estimate for a much more general class of models can be found by optimising the likelihood using gradient-based methods, provided that the likelihood is differentiable with respect to θ . When the optimisation objective is non-convex, such methods are typically only guaranteed to converge to a local optimum. However, such estimates are often sufficient to achieve excellent performance in downstream tasks, as exemplified by the success of supervised deep learning in recent years.

Unfortunately, first-order optimisation methods can be slow to converge when the objective function is poorly conditioned. Natural gradient descent has the potential to converge much faster, and has some particularly desirable properties when applied to MLE objectives, as discussed in Chapter 3. However, exact computation of the natural gradient update is intractable in general, and so approximations are usually required. In Chapter 5 we describe a novel technique for applying natural gradient methods to distributions for which a direct application of NGD is not feasible, with applications to maximum likelihood estimation and other tasks.

4.2 Markov chain Monte Carlo

Whereas maximum likelihood estimation is concerned with obtaining point estimates of unobserved parameters or quantities, Bayesian inference is concerned with obtaining *distributional* estimates. In Bayesian inference, the starting point for inference is a *prior distribution* over unobserved quantities, $p(z)$, that encodes all of our beliefs about z *before* having observed data.

We use z or w to denote latent quantities that we are inferring with distributional estimates, even when those quantities are model parameters. θ is reserved for parameters of models or distributions that we are estimating or optimising directly.

The main object of interest in Bayesian inference is the *posterior* distribution over z , having observed data $\mathcal{D}$. This is obtained by combining the prior distribution, and

the likelihood, by applying Bayes rule, yielding

$$p(z \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid z)p(z)}{p(\mathcal{D})}. \quad (4.3)$$

In some special cases (4.3) has a tractable form, such as when $p(z)$ is a conjugate prior for $p(\mathcal{D} \mid z)$. Unfortunately, such cases are limited. Typically, the exact posterior does not fall within some tractable family of distributions, and even evaluating its density can be challenging because computation of the denominator, $p(\mathcal{D})$, is often intractable. For these reasons, some level of approximation is typically required when performing Bayesian inference.

Monte Carlo methods, when applied to Bayesian inference problems, approximate the posterior distribution using a set of t samples, $\{z_1^{(s)}, \dots, z_t^{(s)}\}$ that are drawn (approximately, typically) from the posterior. This can be viewed as approximating the posterior distribution with a finite mixture of point-mass distributions, so that

$$p(z \mid \mathcal{D}) \approx \frac{1}{t} \sum_{i=1}^t \delta(z - z_i^{(s)}), \quad (4.4)$$

where $\delta(\cdot)$ is a Dirac delta function. Expected *functions* of z can then be estimated using their expectation under the finite mixture distribution:

$$\begin{aligned} \mathbb{E}_{p(z \mid \mathcal{D})}[f(z)] &\approx \int \frac{1}{t} \sum_{i=1}^t \delta(z - z_i^{(s)}) f(z) dz \\ &= \frac{1}{t} \sum_{i=1}^t f(z_i^{(s)}). \end{aligned} \quad (4.5)$$

Markov chain Monte Carlo (MCMC) methods are a particular class of Monte Carlo methods, and are often considered the gold standard for Bayesian inference, due to them having strong asymptotic guarantees. Their wide applicability and ease of use also mean that they have found widespread use across many domains.

MCMC, and other methods in this chapter, apply in settings other than Bayesian inference, and in particular, can be used to approximate unnormalised distributions more generally. We will therefore use the more general notation $p^*(z)$ to denote a distribution we wish to approximate. In the Bayesian setting, $p^*(z) = p(z \mid \mathcal{D})$.

MCMC methods work by taking an initial sample, $z_0^{(s)}$, and repeatedly applying a stochastic transition kernel, $T(z_t^{(s)} \rightarrow z_{t+1}^{(s)})$. The transition kernel is simply a probability distribution from which $z_{t+1}^{(s)}$ is drawn, conditioned on $z_t^{(s)}$, and is constructed in such a way that the marginal distribution of $z_t^{(s)}$ converges to $p^*(z)$, as $t \rightarrow \infty$. $p^*(z)$ is then

known as the *invariant* distribution of T , due to the fact that the transition kernel must leave it invariant – that is,

$$p^*(z) = \int p^*(z')T(z' \rightarrow z)dz'.$$

In practice, due to time and compute constraints, the MCMC transitions can only be repeated for a finite number of iterations. It is common practice therefore to run a (relatively large) number of iterations in order to *approximately* converge to the invariant distribution, discard those initial transitions (known as *burn in*), and then draw (and keep) a further set of samples from which to make inferences involving z .

Eventual convergence to the invariant distribution can be proven for any transition kernel that satisfies some basic properties, but the speed with which they converge can vary significantly depending on the choices made, and much effort has been spent designing kernels with improved efficiency. For continuous p^* , the popular class of Hamiltonian Monte Carlo (HMC) methods (Neal, 1996) is among the most widely used, and scalable, MCMC methods. HMC and its variants have become the de-facto standard method for performing Bayesian inference over continuous variables, and are used by a number of popular probabilistic programming packages, which have found widespread use across many scientific domains. In an idealised setting, when the target distribution is isotropic Gaussian, HMC is *relatively* scalable, with the amount of compute required to obtain independent samples being in $O(d^{5/4})$, where d is the dimensionality of z . In practice however, poor conditioning and non-constant curvature can slow things down significantly, especially so when d is large, and for many problems the time and compute required to converge to the invariant distribution and draw a sufficient number of (effectively) independent samples, can be prohibitively large. In the sections that follow, we will look at alternative approximate inference methods that often converge much faster than MCMC.

4.3 Variational inference

Variational inference (VI) is a popular method for performing approximate Bayesian inference. Variational inference involves finding a distribution q within some tractable family $\mathcal{F}$, such that q approximates p^* . More specifically, we perform the optimisation

$$\min_{q \in \mathcal{F}} \text{KL}[q \parallel p^*], \quad (4.6)$$

so that the resulting q is the member of $\mathcal{F}$ that best approximates $p^*(z)$ in a KL divergence sense. Typically $\mathcal{F}$ is taken to be some parametric family, with parameters θ , so that the optimisation becomes

$$\min_{\theta} \text{KL}[q_{\theta} \parallel p^*]. \quad (4.7)$$

Here the parameters θ are known as the *variational parameters*. In the Bayesian setting, $p^*(z) = p(z \mid \mathcal{D})$, in which case solving (4.7) is equivalent to performing the maximisation

$$\arg \max_{\theta} L(\theta)$$

where

$$\begin{aligned} L(\theta) &= \log p(\mathcal{D}) - \text{KL}[q_{\theta}(z) \parallel p(z \mid \mathcal{D})] \\ &= \mathbb{E}_{q_{\theta}(z)} \left[\log \frac{p(\mathcal{D} \mid z)p(z)}{q_{\theta}(z)} \right]. \end{aligned} \quad (4.8)$$

Because the KL divergence is non-negative, (4.8) shows that $L(\theta)$ is a lower bound on the log marginal likelihood, $\log p(\mathcal{D})$, also known as the *model evidence*, and for that reason $L(\theta)$ is commonly referred to as the evidence lower bound (ELBO).

In statistical physics, the negative ELBO is known as the *free energy*, and so variational inference is sometimes referred to as *variational free energy* (VFE) optimisation. This terminology helps to disambiguate from other approximate Bayesian inference methods that involve a variational optimisation (such as the expectation propagation family of Section 4.4), however, the use of variational inference to refer specifically to VFE optimisation is deeply ingrained in the machine learning community, and so we stick with this usage for the remainder of this thesis.

Note that there is a crucial distinction between the optimisations performed in MLE (4.1) and VI (4.7). In MLE, θ represents the quantity we are inferring directly (the model parameters) In VI, θ parameterises a *distribution* over the quantity of interest (which we denote z), which may itself be model parameters.

Note that we have not yet discussed *how* the optimisation (4.7) is performed. When both q and p^* are from the same exponential family, an (undamped) natural gradient ascent step in L with respect to the natural parameters of q will converge in a single step, as discussed in Section 3.4. Unfortunately this case is not particularly useful, as typically p^* will not lie exactly in the same family as q (although some methods we shall see can make use of *partially* conjugate structure). We now discuss several popular methods for performing the optimization that have particular relevance for

this thesis. All of the variational inference methods that follow – like the HMC method discussed in the previous section – can be considered black-box methods, in that they impose few assumptions on the model itself.

4.3.1 Stochastic gradient variational Bayes

Kingma and Welling (2014) introduced a new method for optimising $L(\theta)$ in certain settings. The key ingredient underlying their method is known as the *reparameterisation trick*. When we wish to compute gradients of an expectation,

$$\mathbb{E}_{q_\theta(z)} [f(z)], \quad (4.9)$$

with respect to θ , the reparameterisation trick proceeds by reparameterising the random variable $z \sim q_\theta$ using a differentiable transformation $z = g_\theta(\varepsilon)$, where ε is drawn from a known, simple, “noise” distribution p . Then, we have

$$\mathbb{E}_{q_\theta(z)} [f(z)] = \mathbb{E}_{p(\varepsilon)} [f(g_\theta(\varepsilon))], \quad (4.10)$$

and the key difference between the two sides of (4.10) is that gradients with respect to θ can be taken inside the expectation on the right hand side. That is,

$$\nabla_\theta \left(\mathbb{E}_{q_\theta(z)} [f(z)] \right) = \mathbb{E}_{p(\varepsilon)} \left[\nabla_\theta (f(g_\theta(\varepsilon))) \right], \quad (4.11)$$

and so an unbiased, stochastic estimate of (4.11) can be obtained by sampling from $\varepsilon \sim p$ and then computing $\nabla_\theta (f(g_\theta(\varepsilon)))$. Applying the reparameterisation trick to (4.3) yields

$$\nabla L(\theta) = \mathbb{E}_{p(\varepsilon)} \left[\nabla_\theta \left(\log \frac{p(\mathcal{D} \mid g_\theta(\varepsilon)) p(g_\theta(\varepsilon))}{q_\theta(g_\theta(\varepsilon))} \right) \right], \quad (4.12)$$

which can then be used to optimise $L(\theta)$ with gradient-based methods. This approach is known as stochastic gradient variational Bayes (SGVB). We note that an important precursor to SGVB is the work of Salimans and Knowles (2012), who proposed an efficient stochastic algorithm for optimising the natural parameters of an exponential family variational approximation.

Many distribution families permit a differentiable transformation g_θ , including location-scale families of distributions (e.g. Gaussian), and families with tractable inverse cumulative density functions. There are notable exceptions however, such as

discrete distribution families. Furthermore, while SGVB is fairly widely applicable, convergence can be slow. One reason for this is that first-order (gradient-based) optimisers can be slow to converge when the objective function is poorly conditioned, as we discussed in Section 3.1. We now discuss alternative variational inference methods that address this issue using natural gradients.

4.3.2 Conjugate-computation VI

The conjugate-computation variational inference (CVI) method of Khan and Lin (2017) uses natural gradients to optimise a VI objective with respect to the natural parameters of a tractable exponential family distribution $q \in \mathcal{F}$.

Without loss of generality, we can decompose $\log p(\mathcal{D}, z)$ into a conjugate term $\eta_0^\top s(z)$ – which may be zero – and a non-conjugate term $\ell(z)$, so that

$$\log p(\mathcal{D}, z) = \eta_0^\top s(z) + \ell(z), \quad (4.13)$$

where $s(\cdot)$ is the statistic function of $\mathcal{F}$. The ELBO with respect to q_η is then given by

$$L(\eta) = \mathbb{E}_{q(z)} \left[\eta_0^\top s(z) + \ell(z) - \log q(z) \right]. \quad (4.14)$$

Using the results of Section 3.4, we have that the natural gradient of L with respect to natural parameters, η , is given by the *flat* gradient with respect to mean parameters, μ , yielding

$$\begin{aligned} \tilde{\nabla} L(\eta) &= \nabla_\mu \left(\mathbb{E}_{q(z)} \left[\eta_0^\top s(z) + \ell(z) \right] - \mathbb{E}_{q(z)} [\log q(z)] \right) \\ &= \eta_0 + \nabla_\mu \mathbb{E}_{q(z)} [\ell(z)] - \eta, \end{aligned} \quad (4.15)$$

where we have used the property that on the interior of $\mathcal{M}$ (the mean domain of $\mathcal{F}$), the negative entropy of q is given by $A^*(\mu)$. CVI performs natural gradient ascent in the ELBO using (4.15). A natural gradient update with step size ϵ is given by

$$\eta \leftarrow \eta + \epsilon \left(\eta_0 + \nabla_\mu \left(\mathbb{E}_{q(z)} [\ell(z)] \right) - \eta \right). \quad (4.16)$$

If we define $\lambda = \eta - \eta_0$, we can express this as update in λ as

$$\lambda \leftarrow \lambda + \epsilon \left(\nabla_\mu \left(\mathbb{E}_{q(z)} [\ell(z)] \right) - \lambda \right), \quad (4.17)$$

and λ can be roughly interpreted as the natural parameters of an $\mathcal{F}$ approximation to $\ell(z)$. CVI can often exhibit rapid convergence to a (local) optimum of the ELBO.

We note that Khan and Lin (2017) also extended CVI to make use of *conditionally* conjugate structure in $\log p(\mathcal{D}, z)$, resulting in a method that generalises variational message passing (Winn and Bishop, 2005). Also related to CVI is the non-conjugate variational message passing method of Knowles and Minka (2011) for multinomial and binary regression, which can be seen as a special case of CVI in the case that gradients are exact (non-stochastic) and when $\epsilon = 1$.

CVI has been developed further since its introduction in Khan and Lin (2017). Notably, it was adapted for use in Bayesian deep learning settings by Khan et al. (2018) and Osawa et al. (2019). Shen et al. (2024) built on this yet further still, incorporating the ideas of Lin et al. (2020) for enforcing positivity constraints, and successfully applying natural-gradient variational learning to achieve competitive performance on ImageNet. Furthermore, Khan and Rue (2023) generalised the ideas underlying CVI beyond the variational inference setting, yielding a general learning rule that incorporates several old and new algorithms, including both Bayesian and non-Bayesian algorithms.

One of the main limitations of CVI is the restriction that $\mathcal{F}$ be a minimal exponential family – in Chapter 5 we introduce a novel natural-gradient-based method for performing variational inference when $\mathcal{F}$ is *not* an exponential family.

4.3.3 Partitioned VI

Bui et al. (2018) introduced the partitioned variational inference (PVI) framework for performing variational inference in federated and continual learning settings. Federated learning uses distributed computing resources to process data in a decentralised manner. In continual learning, new data and tasks arrive over time, and must be incorporated without forgetting those which arrived earlier. In the context of variational inference, what is needed in both of these settings is the ability to update the variational approximation to reflect a single *local* block of data at a time, without forgetting or destroying the contribution from other blocks. This is the setting addressed by the partitioned VI framework.

Again, let $\mathcal{F}$ be an exponential family with statistic function $s(\cdot)$. Assume without loss of generality that the log joint $\log p(\mathcal{D}, z)$ decomposes into a conjugate term $\eta_0^\top s(z)$,

and m non-conjugate terms $\ell_i(z)$ for $i = 1, \dots, m$, so that

$$\log p(\mathcal{D}, z) = \eta_0^\top s(z) + \sum_i \ell_i(z). \quad (4.18)$$

Here $\eta_0^\top s(z)$ may correspond to a prior on z , and each $\ell_i(z)$ to the likelihood $\log p(y_i|z)$ for a single batch of data, although we require no such interpretation.¹ In PVI, we are interested in finding a variational approximation

$$q(z) \propto \exp \left(\left(\eta_0 + \sum_i \lambda_i \right)^\top s(z) \right) \quad (4.19)$$

to approximate $p(\mathcal{D}, z)$, but in a way that updates each λ_i using only information in the local term $\ell_i(z)$. PVI updates each λ_i to increase the local (negative) variational free energy, defined for batch i as

$$L_i(\lambda_i) = \mathbb{E}_{q(z)} [\ell_i(z) - \lambda_i^\top s(z)] + A \left(\eta_0 + \sum_{j=1}^m \lambda_j \right) - c, \quad (4.20)$$

where c is a constant that does not depend on λ_i . It can be shown that maximising $L_i(\cdot)$ is equivalent to performing the minimisation

$$q(z) = \arg \min_{q(z) \in \mathcal{F}} \text{KL}[q(z) \parallel q_i(z)], \quad (4.21)$$

where

$$q_i(z) \propto \exp \left(\left(\eta_0 + \sum_{j \neq i} \lambda_j \right)^\top s(z) \right) \ell_i(z), \quad (4.22)$$

which is known as the (i -th) *tilted distribution* in the EP literature. The PVI framework allows the local objective (4.20) to be optimised in several ways, but here we shall focus on natural gradient ascent. The natural parameters of q are given by $\eta = \eta_0 + \sum_i \lambda_i$. We can rearrange this to obtain λ_i as a function of η , given by

$$\lambda_i(\eta) = \eta - \eta_0 - \sum_{j \neq i} \lambda_j, \quad (4.23)$$

¹Note that the prior log-partition $A(\eta_0)$ can be subsumed into one of the f_i .

where all $\lambda_{j \neq i}$ are considered fixed. This map between η and λ_i (and vice-versa) is affine, and so clearly optimisation of a reparameterisation of L_i in terms of η is equivalent to the one with respect to λ_i , in that a solution to one can trivially be converted to a solution for the other. The natural gradient of L_i with respect to η , is given by

$$\begin{aligned} \tilde{\nabla}_\eta \left(L_i \left(\eta - \eta_0 - \sum_{j \neq i} \lambda_j \right) \right) &= \nabla_\mu \left(\mathbb{E}_{q(z)} [\ell_i(z) - \lambda_i^\top s(z)] + A \left(\eta_0 + \sum_{j=1}^m \lambda_j \right) \right) \\ &= \nabla_\mu \left(\mathbb{E}_{q(z)} [\ell_i(z)] \right) - \frac{\partial \lambda_i}{\partial \mu} \mu - \lambda_i + \frac{\partial \lambda_i}{\partial \mu} \mu \\ &= \nabla_\mu \left(\mathbb{E}_{q(z)} [\ell_i(z)] \right) - \lambda_i. \end{aligned} \quad (4.24)$$

Note that λ_i is a function of η and also therefore depends on μ . Using this to perform a natural gradient update of η with step size ϵ results in

$$\eta \leftarrow \eta + \epsilon \left(\nabla_\mu \left(\mathbb{E}_{q(z)} [\ell_i(z)] \right) - \lambda_i \right). \quad (4.25)$$

By applying the map (4.23) to the updated value of η , we can obtain an equivalent update in terms of λ_i directly, given by

$$\lambda_i \leftarrow \lambda_i + \epsilon \left(\nabla_\mu \left(\mathbb{E}_{q(z; \lambda_i)} [\ell_i(z)] \right) - \lambda_i \right). \quad (4.26)$$

Note that if $m = 1$, then update (4.26) is precisely the same as that of CVI (4.17). We can also show an equivalence when $m > 1$. Let $\lambda = \sum_i \lambda_i$, then a *parallel* update of λ_i for all i results in the following update for λ

$$\begin{aligned} \lambda &\leftarrow \sum_i \left(\lambda_i + \epsilon \left(\nabla_\mu \left(\mathbb{E}_{q(z)} [\ell_i(z)] \right) - \lambda_i \right) \right) \\ &= \lambda + \epsilon \left(\nabla_\mu \left(\mathbb{E}_{q(z)} [\ell(z)] \right) - \lambda \right), \end{aligned} \quad (4.27)$$

where we have defined $\ell(z) = \sum_i \ell_i(z)$. A *parallel* natural gradient PVI update of $[\lambda_i]_i$ therefore results in an update to λ that is equivalent to that of CVI. It was shown by Bui et al. (2018) that update (4.26) is also equivalent to a limiting case of expectation propagation, the subject of Section 4.4. Natural gradient PVI shares the same limitation as CVI, in that it assumes an exponential family approximation, although the PVI framework itself is more general, at least in principle.

4.3.4 Minimal conditional exponential family VI

Lin et al. (2019) introduced the class of *minimal conditional exponential family* (MCEF) distributions, and further showed that natural gradients with respect to the natural parameters of MCEF distributions can be computed cheaply using an identity analogous to that used for exponential family distributions.

A joint distribution $q(z, w)$ is from a *conditional exponential family* when its components have the following form

$$\begin{aligned} q(w) &= \exp \left(\eta_w^\top s_w(w) - A_w(\eta_w) \right) \\ q(z | w) &= \exp \left(\eta_z^\top s_z(z; w) - A_z(\eta_z; w) \right), \end{aligned} \quad (4.28)$$

where $q(w)$ is a density with respect base measure $\nu_w(w)$, and $q(z | w)$ with respect to $\nu_z(w | z)$. $\eta = (\eta_z, \eta_w)$ are known as the *natural parameters*. The *mean parameters* of a CEF distribution are defined as $\mu = (\mu_z, \mu_w)$, where

$$\begin{aligned} \mu_w &= m_w(\eta_w) = \mathbb{E}_{q(w)}[s_w(w)] \\ \mu_z &= m_z(\eta_z, \eta_w) = \mathbb{E}_{q(z, w)}[s_z(z; w)]. \end{aligned} \quad (4.29)$$

A *minimal conditional exponential family* (MCEF) is a CEF for which both $m_w(\cdot)$ and $m_z(\cdot, \eta_w)$ are one-to-one for all $\eta_w \in \Omega_w$, where Ω_w is the natural domain of the exponential family that $q(w)$ belongs to. Lin et al. (2019) showed that for a MCEF distribution q , the following identity holds

$$[F(\eta)]^{-1} \nabla f_\eta(\eta) = \nabla f_\mu(\mu), \quad (4.30)$$

where $F(\eta)$ is the Fisher identity matrix of q with respect to η , and f_η and f_μ are simply reparameterisations of the same function. This identity is analogous to the results for exponential family natural gradients, and means that natural gradients with respect to natural parameters of an MCEF can be efficiently computed provided that the backward mapping from μ to η is also tractable. This result expands the set of distributions with tractable natural gradients beyond just minimal exponential families.

Lin et al. (2019) also showed that several known distributions can be expressed as the *marginal* (over z) of a MCEF distribution, including: finite mixtures of exponential families, multivariate skew-normal, and multivariate Student's- t distributions. The authors then proceeded to perform natural gradient ascent of $L[q(z)]$ with respect to a MCEF distribution with density $\tilde{q}(z, w)$, for which $q(z)$ is the z -marginal. It is

noteworthy that the ELBO $\mathcal{L}[q(z)]$ is equal (up to a constant) to a KL divergence involving $q(z)$ (the MCEF marginal), but the authors performed natural gradients by using the FIM of $\tilde{q}(z, w)$ (the MCEF joint), because inverse-vector products are tractable for the FIM of the latter but not for the former.

In this thesis we use tilde notation, e.g. $\tilde{q}$, to denote a distribution or density that is being used as a *surrogate* for another, e.g. q , for the purposes of optimisation.

This apparent mismatch in distributions was not explicitly discussed by the authors, perhaps because the two distributions are so closely related (by marginalisation). In Chapter 5 we use this kind of mismatch by design, and show that using a tractable *surrogate* distribution in place of an otherwise intractable *target* distribution for the purposes of computing natural gradients, can allow us to perform rapid optimisation of the target distribution parameters.

Lin et al. (2019) expanded the set of distributions for which natural gradients are tractable beyond just exponential families. Unfortunately the set of MCEF distributions, or distributions that can be expressed as a MCEF marginal, is still rather limited.

One potential drawback of variational inference is that its targeting of the reverse KL divergence means that it tends to favour mode-seeking, or zero-avoiding, solutions. In some settings this may be undesirable, and finding approximations with mode-covering, or other characteristics, may be more desirable. The reverse KL divergence is just one example from the family of *alpha* divergences (Amari, 2012), which also includes the mode-covering *forward* KL divergence. The expectation propagation family of algorithms, described in the next section, allow us to obtain variational approximations that (approximately) minimise *any* alpha divergence.

4.4 Expectation propagation

Expectation propagation (EP) (Minka, 2001, Opper and Winther, 2000) is another family of algorithms for performing approximate Bayesian inference. Hasenclever et al. (2017) derived a variational problem that is solved by EP and its variants, using an approach that drew heavily on Lagrangian optimisation and convex duality in exponential families. We largely follow their presentation here, and refer the reader to their original paper for a more detailed treatment. Although this is not the typical presentation of EP, it allows us to present the different variants in a unified way, and will also provide important context for the developments of Chapter 6. We note that

the presentation of Hasenclever et al. (2017) builds on earlier work, in particular by Minka (2007), who presented a similar variational objective for EP, but without the emphasis on convex duality in exponential families.

We then look at several existing methods for solving the EP variational problem, beginning with the original EP algorithm of Minka (2001), which can be viewed as performing a sequence of KL divergence minimisations in order to reach a fixed point. We then discuss the more general power EP algorithm (Minka, 2004), which can be used to target arbitrary alpha divergences (Amari, 2012, Trottini and Spezzaferri, 2002). We then present the double-loop EP algorithm of Heskes and Zoeter (2002), which provably converges to a solution of the EP variational problem. Finally, we present the stochastic natural gradient EP algorithm of Hasenclever et al. (2017), another double-loop algorithm, that has a number of advantages in stochastic settings.

All of the methods presented in this section can be viewed as alternative ways of solving the EP variational problem. Note that we use EP both as an umbrella term that encompasses all of these methods, and as a term to refer specifically to the original EP algorithm of Minka (2001). We hope that the intended usage will be clear from context.

4.4.1 EP variational problem

Let $\mathcal{F}$ be the minimal exponential family defined by the statistic function $s(\cdot)$ with respect to base measure $\nu(\cdot)$, with the log partition and its convex dual given by $A(\cdot)$ and $A^*(\cdot)$ respectively. Let Ω and $\mathcal{M}^\circ$ be the natural and mean domain of $\mathcal{F}$, respectively. We shall refer to $\mathcal{F}$ as the *base family*.

Let $p^*(z)$ be a density of the form

$$p^*(z) \propto p_0(z) \prod_i \exp(\ell_i(z)), \quad (4.31)$$

where $p_0(z)$ is the member of $\mathcal{F}$ with natural parameter η_0 . Note that we can assume this form without loss of generality if we allow $p_0(z)$ to be improper. Each $\ell_i(z)$ may correspond to the log likelihood $\log p(\mathcal{D}_i | z)$ corresponding to a single observation or batch of observations $\mathcal{D}_i$, but we do not require such an interpretation. Despite the assumption of a tractable $p_0(z)$, exact inference under $p^*(z)$ is intractable in general due to the presence of non-conjugate $\exp(\ell_i(z))$ factors.

Without loss of generality we can consider p^* as a member of an *extended exponential family*, with sufficient statistic vector $\bar{s}(z) = (s(z), \ell_1(z), \dots, \ell_m(z))$, natural parameters $\bar{\eta} = (\eta_0, \mathbf{1}_m)$, and log-partition function $\bar{A}(\bar{\eta})$. Solving the inference problem then

amounts to computing the log partition function of this extended family member, which, by convex duality, is equivalent to the concave maximisation problem

$$\bar{A}(\bar{\eta}) = \max_{\bar{\mu} \in \bar{\mathcal{M}}} \bar{\eta}^\top \bar{\mu} - \bar{A}^*(\bar{\mu}) \quad (4.32)$$

where $\bar{\mathcal{M}}$ is the set of realisable extended family moments, and $\bar{A}^*(.)$ is the convex dual of $\bar{A}(.)$. Note that (4.32) has a maximum at $\bar{\mu} = \mathbb{E}_{p^*(z)}[\bar{s}(z)]$, and so by performing the maximisation in (4.32), we recover both the normaliser for our target distribution, as well as its (base family) moments.

Despite the formulation as a concave maximisation problem, the variational optimisation (4.32) remains generally intractable for two reasons. First, the set of realisable mean parameters is not easily characterised, and second, the negative entropy function, $\bar{A}^*(\bar{\mu})$, is not typically easy to compute (Wainwright and Jordan, 2008). Further approximations are needed to deal with these issues.

For each factor i , let the i -th *locally extended exponential family* $\mathcal{F}_i$ be defined by the statistics function $s_i(z) = (s(z), \ell_i(z))$, with $\Omega_i, \mathcal{M}_i^\circ, A_i(.), A_i^*(.), q_i(.)$ the corresponding natural domain, mean domain, log partition function, negative entropy, and density, respectively. The locally extended families are closely related to what are referred to as *tilted distributions* in the EP literature. Specifically, the i -th tilted distribution lives in the i -th extended family. Going forwards we assume i and j to live in the index set $\{1, \dots, m\}$ and hence use e.g. $\forall i$ as shorthand for i in $1, \dots, m$.

The first approximation is that the negative entropy is replaced with the following approximation

$$\bar{A}^*(\bar{\mu}) \approx A^*(\mu) + \sum_i \beta_i (A_i^*(\mu, \omega_i) - A^*(\mu)) \quad (4.33)$$

where $\bar{\mu} = (\mu, \omega_1, \dots, \omega_m)$ is the vector of extended family mean parameters. This entropy approximation can be seen as a generalisation of the Bethe entropy approximation (Wainwright and Jordan, 2008).

The second approximation is that the set $\bar{\mathcal{M}}$ is replaced by a relaxed set of local consistency constraints, characterised by

$$\{(\mu, \omega_1, \dots, \omega_m) : (\mu, \omega_i) \in \mathcal{M}_i \forall i\}, \quad (4.34)$$

which provides a convex outer bound on $\bar{\mathcal{M}}$. Finally, we introduce a locally extended mean parameter vector (μ_i, ω_i) for each i , with an additional constraint to ensure

$\mu_i = \mu \forall i$. Incorporating these changes into (4.32) yields the EP variational problem

$$\max_{\mu, [\mu_i, \omega_i]_i} \eta_0^\top \mu + \sum_i \omega_i - A^*(\mu) - \sum_i \beta_i (A_i^*(\mu_i, \omega_i) - A^*(\mu_i)) \quad (4.35)$$

subject to

$$\begin{aligned} \mu &\in \mathcal{M} \\ (\mu_i, \omega_i) &\in \mathcal{M}_i \forall i \\ \mu &= \mu_i \forall i. \end{aligned} \quad (4.36)$$

Each hyperparameter β_i controls the characteristics of the approximation to the i -th factor. It corresponds to the power parameter of power EP (Minka, 2004) – discussed in Section 4.4.3 – with the original EP algorithm of Minka (2001) corresponding to $\beta_i = 1 \forall i$.

4.4.2 EP

The updates of EP can be derived by introducing Lagrange multipliers $[\lambda_i]_i$ to enforce the equality constraints in (4.36) and then using critical points of the Lagrangian to derive fixed point iterations (Hasenclever et al., 2017). Let us begin by defining the Lagrangian

$$\begin{aligned} L(\mu, [\mu_i, \omega_i, \lambda_i]_i) &= \eta_0^\top \mu + \sum_i \omega_i - A^*(\mu) \\ &\quad - \sum_i \left(\lambda_i^\top (\mu_i - \mu) + \beta_i (A_i^*(\mu_i, \omega_i) - A^*(\mu_i)) \right), \end{aligned} \quad (4.37)$$

so that the EP variational problem becomes

$$\max_{\mu, [\mu_i, \omega_i]_i} \min_{[\lambda_i]_i} L(\mu, [\mu_i, \omega_i, \lambda_i]_i) \quad (4.38)$$

subject to

$$\begin{aligned} \mu &\in \mathcal{M} \\ (\mu_i, \omega_i) &\in \mathcal{M}_i \forall i. \end{aligned} \quad (4.39)$$

By equating $\frac{\partial L}{\partial \mu}$ to 0, we know that at a critical point of L ,

$$\nabla A^*(\mu) = \eta_0 + \sum_i \lambda_i. \quad (4.40)$$

Recall from the definition of the extended exponential family distribution that μ was originally introduced as the expected sufficient statistics under p^* . This is now only approximately true, due to the approximations we have introduced, but nevertheless it implies that $p^*(z)$ can be approximated by

$$q(z) \propto \exp \left(\left(\eta_0 + \sum_i \lambda_i \right)^\top s(z) \right). \quad (4.41)$$

Given $\mu_i = \mu \forall i$, and by equating $\frac{\partial L}{\partial \mu_i}$ and $\frac{\partial L}{\partial \omega_i}$ to 0, we also have that at a critical point,

$$\begin{aligned} \nabla_{\mu_i} A_i^*(\mu_i, \omega_i) &= \eta_0 + \sum_{j=1}^m \lambda_j - \beta_i^{-1} \lambda_i \\ \nabla_{\omega_i} A_i^*(\mu_i, \omega_i) &= \beta_i^{-1}, \end{aligned} \quad (4.42)$$

and so μ_i must be equal to the expected sufficient statistics under the $\mathcal{F}_i$ member with natural parameter $(\eta_0 + \sum_{j=1}^m \lambda_j - \beta_i^{-1} \lambda_i, \beta^{-1})$, with density proportional to

$$\exp \left(\left(\eta_0 + \sum_{j=1}^m \lambda_j - \beta_i^{-1} \lambda_i \right)^\top s(z) + \beta_i^{-1} \ell_i(z) \right), \quad (4.43)$$

which is known as the i -th *tilted distribution*. In the original EP algorithm of Minka (2001), $\beta_i = 1 \forall i$, in which case (4.43) simplifies to

$$\exp \left(\left(\eta_0 + \sum_{j \neq i} \lambda_j \right)^\top s(z) + \ell_i(z) \right). \quad (4.44)$$

Taken together, these optimality conditions tell us that $\eta_0 + \sum_i \lambda_i = \nabla A^*(\mu_i) \forall i$, where μ_i are the expected sufficient statistics under the i -th tilted distribution. The EP algorithm of Minka (2001) works by updating λ_i to locally enforce this constraint (for the current i), resulting in the update

$$\lambda_i \leftarrow \nabla A^*(\mu_i) - \eta_0 - \sum_{j \neq i} \lambda_j, \quad (4.45)$$

and then repeats this for all i according to some schedule (or in parallel) until convergence. These fixed point updates are not guaranteed to converge, but convergence can often be aided by using damped updates

$$\begin{aligned}\lambda_i &\leftarrow (1 - \alpha)\lambda_i + \alpha \left(\nabla A^*(\mu_i) - \eta_0 - \sum_{j \neq i} \lambda_j \right) \\ &= \lambda_i - \alpha \left(\eta_0 + \sum_j \lambda_j - \nabla A^*(\mu_i) \right),\end{aligned}\tag{4.46}$$

where the level of damping is given by $1 - \alpha$.

Update (4.46) is that of the EP algorithm of Minka (2001), which we derived – again due to Hasenclever et al. (2017) – by finding critical points of the Lagrangian obtained from the EP variational problem. While this is not the standard presentation of the EP updates, it provides important context for the developments that follow. The more typical presentation of EP derives it as a series of local forward KL divergence minimisations, iterated until convergence. In Appendix D.1 we show that this more conventional presentation of EP results in the same update.

4.4.3 Power EP

The power EP algorithm of Minka (2004) corresponds to choices of β_i^{-1} other than 1 in the EP variational problem (4.36). The power EP update is still given by (4.46), but note that the dependence on β_i^{-1} comes through the definition of the tilted distribution, and hence through μ_i . Whereas standard EP is equivalent to performing a sequence of local KL divergence minimisations (see Appendix D.1), the fixed points of power EP can be shown to have the same fixed points as an EP-style algorithm that instead minimises local *alpha* divergences (Amari, 2012) on each iteration (Wiegerinck and Heskes, 2002, Minka, 2004). By taking the limit as $\beta_i \rightarrow \infty$, the alpha divergence becomes the reverse KL divergence, and in fact in this limiting case, update (4.46) becomes exactly equivalent to that of natural gradient PVI (4.26) if we also scale the step size by β_i (Bui et al., 2018).

For finite β_i , update (4.46) does not have any obvious connection with natural gradients, and we are not aware of any such connection having been made by earlier works. However, we show in Chapter 6 that it can in fact be viewed as performing NGD in a variational problem that is described in the next section.

4.4.4 Double-loop EP

Heskes and Zoeter (2002) introduced an algorithm that, unlike standard EP, is guaranteed to converge to a solution of the EP variational problem. The algorithm is known as double-loop EP, due to it consisting of an inner minimisation loop coupled with an outer maximisation loop, which work together to reach a solution of a saddle-point optimisation problem. We shall first derive the saddle-point problem following the approach of Hasenclever et al. (2017). We will then use the saddle-point problem to derive the updates of double-loop (power) EP.

We begin by introducing an auxiliary variable $\eta \in \Omega$,² and subtracting the non-negative term $\sum_i \text{KL}[q_{\mu_i} \parallel q_\eta]$ from (4.35), where q_{μ_i} and q_η are the members of $\mathcal{F}$ with mean parameters μ_i and natural parameters η , respectively. This creates a lower bound on the EP variational objective,

$$\max_{\eta, \mu, [\mu_i, \omega_i]_i} \eta_0^\top \mu - A^*(\mu) + \sum_i \left(\omega_i - \beta_i \left(A_i^*(\mu_i, \omega_i) - \mu_i^\top \eta + A(\eta) \right) \right)$$

subject to

$$\begin{aligned} \eta &\in \Omega \\ \mu &\in \mathcal{M} \\ (\mu_i, \omega_i) &\in \mathcal{M}_i \quad \forall i \\ \mu_i &= \mu \quad \forall i. \end{aligned} \tag{4.47}$$

Double-loop EP operates by performing co-ordinate ascent in (4.47), alternating between maximising η for fixed $\mu, [\mu_i, \omega_i]_i$, and then performing (constrained) maximisation with respect to $\mu, [\mu_i, \omega_i]_i$, for fixed η . Provided that the constrained objective is bounded above, this will be guaranteed to converge to a local optimum. Note that by taking derivatives of the auxiliary objective with respect to η , the objective is maximised when $\eta = \nabla A^*(\mu)$ (because $\mu_i = \mu \quad \forall i$ due to the constraints), at which point the auxiliary KL divergence terms vanish and the bound on (4.36) becomes tight. All that remains then is to solve the constrained maximisation with respect to $\mu, [\mu_i, \omega_i]_i$. As before, we add Lagrange multipliers to enforce the equality constraints

²Hasenclever et al. (2017) introduce m separate auxiliary variables, one for each factor. However, the optimality conditions dictate that they are all equal, and so we simplify here by using a single tied variable η . The use of distinct auxiliary variables only becomes meaningful when additional structure is assumed in (4.31).

$\mu_i = \mu \forall i$, resulting in the following modified problem

$$\begin{aligned}
& \max_{\eta} \max_{\mu, [\mu_i, \omega_i]_i} \min_{[\lambda_i]_i} \eta_0^\top \mu - A^*(\mu) + \sum_i \left(\right. \\
& \quad \left. \omega_i - \lambda_i^\top (\mu_i - \mu) - \beta_i \left(A_i^*(\mu_i, \omega_i) - \mu_i^\top \eta + A(\eta) \right) \right) \\
& \text{subject to } \mu \in \mathcal{M} \\
& \quad \eta \in \Omega \\
& \quad (\mu_i, \omega_i) \in \mathcal{M}_i \forall i.
\end{aligned} \tag{4.48}$$

Because the objective in (4.48) is concave in $\mu, [\mu_i, \omega_i]_i$, and Slater's condition holds, we have strong duality in the inner problem and can reverse the order of the min/max. Next, substituting the definitions of the log-partition functions in terms of their convex duals, we recover the *EP saddle-point problem*

$$\begin{aligned}
& \max_{\eta} \min_{[\lambda_i]_i} L(\eta, [\lambda_i]_i) \\
& \text{subject to } \eta \in \Omega
\end{aligned} \tag{4.49}$$

where

$$L(\eta, [\lambda_i]_i) = A\left(\eta_0 + \sum_i \lambda_i\right) + \sum_i \left(\beta_i \left(A_i(\eta - \beta_i^{-1} \lambda_i, \beta_i^{-1}) - A(\eta) \right) \right). \tag{4.50}$$

Note that $L(\eta, [\lambda_i]_i)$ is convex in $[\lambda_i]_i$, and so by solving a convex optimisation problem, we can recover the $[\lambda_i]_i$ that solve the inner constrained max-min problem of (4.48). Furthermore, taking derivatives of (4.48) with respect to μ , we also have that $\nabla A^*(\mu) = \eta_0 + \sum_i \lambda_i$ at critical points, which is the quantity we need to perform the (outer) maximisation with respect to η . Co-ordinate ascent in (4.47) then implies alternating two steps:

1. $[\lambda_i]_i \leftarrow \arg \min_{[\lambda_i]_i} L(\eta, [\lambda_i]_i)$
2. $\eta \leftarrow \eta_0 + \sum_i \lambda_i$.

Heskes and Zoeter (2003) proposed two different ways of performing the inner minimisation (step 1). The first simply follows the gradient with respect to λ_i , and the second uses a (damped) fixed-point update, similar to that of standard EP. The phrase double-loop EP is typically used to refer specifically to the latter method, which we focus on here. Taking derivatives of $L(\eta, [\lambda_i]_i)$ with respect to λ_i and equating to 0, we

have the optimality condition

$$\begin{aligned}\nabla A\left(\eta_0 + \sum_i \lambda_i\right) &= \nabla A_i\left(\eta - \beta_i^{-1}\lambda_i, \beta_i^{-1}\right) \\ &= \mathbb{E}_{q_i(z)}[s(z)],\end{aligned}\tag{4.51}$$

where $q_i(z)$ is the locally extended family member with natural parameter $(\eta - \beta_i^{-1}\lambda_i, \beta_i^{-1})$ and density

$$q_i(z) \propto \exp\left(\left(\eta - \beta_i^{-1}\lambda_i\right)^\top s(z) + \beta_i^{-1}\ell_i(z)\right).\tag{4.52}$$

Note that this distribution is closely related to the tilted distribution of standard (power) EP (4.43) – the two distributions are equivalent when $\eta = \eta_0 + \sum_i \lambda_i$, which as we have shown above, will be true after every “outer” update. We shall see later that the double-loop algorithm can be viewed as including (power) EP as a special case, and so going forward we shall reserve the term *tilted distribution* exclusively for referring to q_i as defined by (4.52). We can use optimality condition (4.51) to define a fixed-point update for λ_i , yielding

$$\lambda_i \leftarrow \nabla A^*\left(\mathbb{E}_{q_i(z)}[s(z)]\right) - \eta_0 - \sum_{j \neq i} \lambda_j,\tag{4.53}$$

which with damping, becomes

$$\lambda_i \leftarrow \lambda_i - \alpha \left(\eta_0 + \sum_j \lambda_j - \nabla A^*\left(\mathbb{E}_{q_i(z)}[s(z)]\right) \right).\tag{4.54}$$

Heskes and Zoeter (2003) showed that update (4.54) follows a decreasing direction in $L(\eta, [\lambda_i]_i)$, and so for sufficiently small α , a sequence of updates is guaranteed to converge to a minimum of $L(\eta, [\lambda_i]_i)$. Note that this update is equivalent to (4.46) whenever $\eta = \eta_0 + \sum_j \lambda_j$, which is true immediately after an “outer” update (of η). Therefore, standard (power) EP can be viewed as a special case of double-loop (power) EP, but with only a single “inner” update (4.54) performed for each outer update (Jylänki et al., 2011). Of course, without performing the inner optimisation to convergence, the overall procedure is no longer guaranteed to converge, but in practice, performing some truncated number (≥ 1) of inner updates is typically sufficient (Jylänki et al., 2011, Hasenclever et al., 2017).

4.4.5 Stochastic natural gradient EP

The stochastic natural gradient EP (SNEP) algorithm of Hasenclever et al. (2017) is also a double-loop algorithm, and differs from the double-loop EP of Heskes and Zoeter (2002) only in how it performs the *inner* optimisation of (4.49) with respect to $[\lambda_i]_i$. In SNEP, each site parameter λ_i is treated as the natural parameters of a bone fide distribution in $\mathcal{F}$, and then (4.50) is optimised using natural gradient descent with respect to *mean* parameters of those distributions.

Let $\tilde{q}_{\lambda_i}$ be the member of $\mathcal{F}$ with natural parameter λ_i . Let γ_i be the mean parameters of $\tilde{q}_{\lambda_i}$. Then, following the results of Section 3.4, the natural gradient of $L(\eta, [\lambda_j]_j)$ with respect to *mean* parameters of $\tilde{q}_{\lambda_i}$ is simply the flat gradient with respect to λ_i , given by

$$\begin{aligned}\nabla_{\lambda_i}(L(\eta, [\lambda_j]_j)) &= \nabla A\left(\eta_0 + \sum_j \lambda_j\right) - \nabla A_i\left(\eta - \beta_i^{-1}\lambda_i, \beta_i^{-1}\right) \\ &= \nabla A\left(\eta_0 + \sum_j \lambda_j\right) - \mathbb{E}_{q_i(z)}[s(z)].\end{aligned}\tag{4.55}$$

A (damped) natural gradient descent step in L with respect to γ_i is then given by

$$\gamma_i \leftarrow \gamma_i - \epsilon \left(\nabla A\left(\eta_0 + \sum_j \lambda_j\right) - \mathbb{E}_{q_i(z)}[s(z)] \right).\tag{4.56}$$

SNEP uses update (4.56) to perform the inner minimisation, and converts back to natural parameters using the reverse map $\lambda_i \leftarrow \nabla A^*(\gamma_i)$ before performing the outer update with respect to η . Note that the real subjects of the optimisation are the distributions q and q_i for $i = 1$ to m . SNEP then uses $\tilde{q}_{\lambda_i}$ as a *surrogate* distribution for performing natural gradient optimisation, because performing it (or its natural extension) with respect to a joint statistical manifold for $q, q_i \forall i$, is intractable in general. In this way, SNEP can be viewed as an example of a more general technique that we present in Chapter 5.

SNEP has some notable advantages over (power) EP and its double-loop variants. In particular, because its updates are simply NGD of a defined objective, and those updates are linear in the tilted distribution moments, SNEP is amenable to stochastic estimation of those moments. The updates of EP on the other hand, are not robust to stochastic estimates of the tilted distribution moments, for reasons we discuss later, and so must use a large number of samples to minimise noise in the estimates as a

result. This means that SNEP updates can use far fewer samples per update while still remaining stable, which can be more computationally efficient, as it is not necessary to commit a large number of samples (and hence much computation) in order to compute an accurate update direction which subsequently may change. SNEP is, as a result, better-suited for use in black-box settings, potentially enabling its use on a much wider class of models than standard (power) EP.

Unfortunately SNEP still has some challenges. One problem with the SNEP approach is that, by performing natural gradient descent in a mean parameterisation of the site potentials, they are constrained to lie in the natural domain Ω , which is not the case in the original objective (4.49). In fact, it is perfectly valid, and not uncommon, for converged EP solutions to have site parameters that do not *individually* lie in the natural domain. This means that the optimum of the inner minimisation in (4.49) may not be attainable with SNEP, and so the convergence guarantees of double-loop EP no longer apply.

The second, and perhaps more consequential issue, is that despite its apparent advantages, the SNEP update takes considerably longer to converge than that of (power) EP, even in settings where the moments must be estimated with samples. This performance differential is apparent in the experiments of Chapter 6, and was also observed by Vehtari et al. (2020). It is also not inconsistent with the results of Hasenclever et al. (2017), which evaluated convergence by monitoring metrics such as classification accuracy and predictive RMSE, but did not monitor convergence of the posterior approximation itself. We believe the reason for the poor relative performance of SNEP is that NGD is performed with respect to the distributions $\tilde{q}_{\lambda_1}, \dots, \tilde{q}_{\lambda_m}$, whereas the objective L involves KL divergences in the distributions $q, q_1, \dots, q_m$, and the former may bear little resemblance to the latter. This means that we have no reason to believe that NGD with respect to $\tilde{q}_{\lambda_1}, \dots, \tilde{q}_{\lambda_m}$ will be efficient. We make this point more concretely in Appendix D.6.

4.4.6 EP as a black-box inference method

The EP updates depend on the computation of tilted distribution moments. EP is typically applied in settings for which these either have convenient closed-form solutions, or in which the expectations are over sufficiently low-dimensional quantities (often making use of additional structural assumptions in p^*) that they can be estimated using quadrature methods. Unfortunately, these constraints mean that the set of models EP (and most of its variants) can be applied to is rather limited.

We saw in Section (4.2) however that MCMC methods can be used to estimate expectations under a very broad set of distributions, with few restrictions. EP can in principle be used in a *black-box* manner, by using MCMC methods to estimate the moments needed for its updates. This combination of EP with MCMC methods has the potential to dramatically expand the set of models EP can be applied to, and can also be viewed as a framework for performing parallel and distributed MCMC (Xu et al., 2014, Vehtari et al., 2020, Hasenclever et al., 2017). This usage of EP has several computational advantages over direct MCMC, some of which we believe have been overlooked or under-appreciated in previous work. We summarise these advantages below.

Inherently distributed and parallel The updates of EP can naturally be performed in parallel over i . Furthermore, the updates interact only through the site parameters, making them well suited to distributed architectures. A particularly relevant example is when a dataset can be partitioned such that each $\ell_i(z)$ operates on only a single block of size n/m , for n the full dataset size. In this case, each EP update is computed using MCMC steps that are approximately m times cheaper than those on the full dataset. If the updates are performed in parallel this can lead to a significant reduction in wall-clock time.³ The distributed nature of EP is also suitable for federated learning settings, as there is no direct transfer of dataset blocks between nodes.

Simpler geometry The MCMC target distributions for computing moments are the tilted distributions. These distributions are *approximately* members of $\mathcal{F}$, consisting of a member of $\mathcal{F}$ “tilted” by a single non- $\mathcal{F}$ factor. This will typically have simpler geometry than $p^*(z)$. For example, when $\mathcal{F}$ is a multivariate normal family, the log tilted distribution will have near constant curvature, the ideal landscape for Hamiltonian Monte Carlo. This means that using (well-tuned) HMC on the tilted distributions should be more efficient (fewer steps per effective sample) than using HMC on p^* .

Tractable approximation Even before convergence, $q \in \mathcal{F}$ serves as an approximation to q_i (differing only by a single factor). q can therefore be used to aid MCMC sampling from q_i in several ways, e.g. the covariance of q can be used to set the mass

³The computation of gradients for e.g. HMC can also be distributed in this way, however this requires data to be communicated across nodes at every step of the leapfrog integrator, as opposed to once every EP update. Communication overheads are often the main performance bottleneck in distributed systems.

matrix in a HMC scheme, and samples from q can be used to initialise or resample MCMC chains. We discuss these ideas in more detail in Section 6.7.

Fewer sampling dimensions The tilted distributions often operate on fewer dimensions than the full target distribution. For example, in hierarchical Bayesian models, a typical model structure is given by $p_0(z) \prod_{i=1}^m p(w_i | z) p(\mathcal{D}_i | w_i)$, where z is a global parameter vector, and w_i and $\mathcal{D}_i$ are the local parameters and observed data for a particular sub-problem. The full joint posterior $p(z, w_1, \dots, w_m | \mathcal{D})$ requires sampling jointly over $n_z + mn_w$ variables, for n_z and n_w the dimensionalities of z and w_i respectively. By letting $\ell_i(z) = \log \int p(\mathcal{D}_i, w_i | z) dw_i$ however, estimating expectations under the tilted distributions reduces to m sampling problems in $n_z + n_w$ dimensions. This can lead to significant efficiency gains when m is large. This idea was leveraged by Vehtari et al. (2020) to apply EP to a number of challenging hierarchical Bayesian inference problems. Another setting in which the tilted distributions have reduced dimensionality is when $\mathcal{F}$ is closed under marginalisation and the target factors operate on cliques of smaller dimension than z , see Seeger (2008) for details.

Unfortunately, noise in the moment estimates of EP leads to biased updates and an unstable procedure, for reasons we discuss in detail in Chapter 6. This problem has been addressed in the past using debiasing techniques combined with a large number of MCMC samples (Xu et al., 2014, Vehtari et al., 2020). This approach has several problems however, including poor computational efficiency, and the introduction of additional hyperparameters that are difficult to tune. The stochastic natural gradient EP (SNEP) method introduced in Section 4.4.5 provides an alternative way of dealing with stochastic moment estimates in EP, yet this suffers from slow convergence. In Chapter 6 we introduce two new EP variants that are ideally suited to settings in which only noisy moment estimates are available, combining the relative strengths of all of the EP variants discussed in this section.

4.5 Summary

In this chapter we looked at a number of popular methods for performing inference. Several of these methods involve performing an optimisation with respect to the parameters of a distribution, q . In the case of maximum-likelihood estimation (MLE), q is the model itself. In variational inference (VI) and expectation propagation (EP), q is a tractable approximation to a distribution of interest – typically a posterior

distribution over unknown quantities. For MLE and VI, the optimisations can be performed for a very general class of q distributions by using first-order (gradient-based) optimisation. Unfortunately, first-order methods can be slow to converge when the objective is poorly conditioned.

When q is from an exponential family, the MLE optimisation can be solved in a single step, which is equivalent to performing a natural gradient step in the mean parameters of q . For VI, a similar result holds when performing a natural gradient step in the *natural* parameters of q , but only in the case that p^* – the distribution we are approximating – is fully conjugate, i.e. in the same exponential family as q . The conjugate-computation VI method of Khan and Lin (2017) generalises natural gradient VI to the case of non-conjugate models, but the approximation family $\mathcal{F}$ is still required to be an exponential family. In Chapter 5 we introduce a novel way of optimising the parameters of distributions that are not in an exponential family, by performing natural gradient descent with respect to an exponential family *surrogate* distribution. This allows efficient natural-gradient-based methods to be applied to distributions beyond exponential families. The minimal conditional exponential family VI method of Lin et al. (2019) and the stochastic natural gradient EP method of Hasenclever et al. (2017) can be viewed as (implicitly) applying this same idea.

In Bayesian inference problems, Markov chain Monte Carlo (MCMC) methods are often considered the gold standard, due to strong theoretical guarantees. Unfortunately, when the target distribution p^* is very high-dimensional, or has particularly challenging geometry, the time and compute required in order to reach the asymptotic regime can be intractably large. The variational inference methods discussed in this chapter can often yield results far faster than MCMC. However, the zero-avoiding nature of the reverse KL divergence targeted by VI, means that the approximation it obtains can significantly underestimate uncertainty.

Expectation propagation, when it can be applied, can also yield results far faster than MCMC, and is often found to have better calibrated uncertainties than VI (Bui et al., 2017). Furthermore, the more general power EP method of Minka (2004) allows one to (approximately) target arbitrary alpha divergences (Amari, 2012), using the power parameters (β_i for $i = 1$ to m) to smoothly interpolate between the zero-avoiding behaviour of VI and the zero-covering behaviour of EP. The updates of EP involve computing expectations under certain distributions – the tilted distributions – which means that EP has typically only been applied to models for which these expectations can be efficiently computed or estimated. Unfortunately this means that existing EP

variants are not as readily applied to the full set of problems that can be targeted with MCMC and VI.

EP can in principle be used in a *black-box* manner, by using MCMC methods to estimate the moments needed for their updates. This combination of EP with MCMC methods has been motivated as a framework for performing parallel and distributed black-box inference (Xu et al., 2014, Vehtari et al., 2020, Hasenclever et al., 2017). In principle, this approach has several computational advantages over direct applications of MCMC, as discussed in Section 4.4.6. Unfortunately, most existing EP variants are not robust in the presence of Monte Carlo noise. The stochastic natural gradient EP method of Hasenclever et al. (2017) addresses this issue by introducing new updates that are robust in the presence of noisy moment estimates, but this comes at the cost of the updates taking far longer to converge than those of standard (power) EP. In Chapter 6 we introduce new EP variants that, like SNEP, are robust to noise in the moment estimates, but unlike SNEP, our variants result in faster convergence than their standard (power) EP counterparts. As a result, these new variants dramatically expand the set of models that EP can be effectively applied to.

In this chapter we also highlighted several (known) connections between the various inference methods discussed. In particular, CVI can be seen as a generalisation of natural gradient VI to non-conjugate models, and PVI, when combined with natural gradient updates, can be seen as applying CVI in a distributed manner (Bui et al., 2018). The limiting case of power EP, when $\beta_i \rightarrow \infty \forall i$, is equivalent to natural gradient PVI (Bui et al., 2018). Additionally, Wilkinson et al. (2023) showed that natural gradient VI, EP, and posterior linearisation (García-Fernández et al., 2017), can all be viewed as optimising the parameters of posterior distribution parameters using a generalisation of Newton’s method. In Chapter (6) we present a new natural gradient interpretation of the updates of (power) EP. Namely, that they can be interpreted as performing natural-gradient-based optimisation of the EP variational objective, with respect to certain surrogate distributions, similar to SNEP. Unlike SNEP however, the surrogates used by (power) EP are closely related to the distributions being optimised, resulting in better optimisation performance, a point we discuss in more detail in Appendix D.6.

We hope that the contributions contained in the chapters to follow will be directly useful in their own right, and also that they may inspire further work and new advances in efficient inference methods.

Chapter 5

Optimising distributions with natural gradient surrogates

Many inference problems, such as MLE and VI, can be expressed as the optimisation of the parameters of probability distributions. In cases where the optimisation objective is differentiable, these optimisations can often be performed using standard first-order methods, but first-order optimisers tend to perform poorly when the objective is ill-conditioned. In some special cases – notably when the distribution being optimised is from an exponential family – the optimisation can be performed efficiently using the natural gradient method introduced in Chapter 3, which often has desirable properties in such inference optimisations. Unfortunately there are many distribution families for which the natural gradient cannot be efficiently computed. In this chapter we discuss a novel method for optimising the parameters of such families using natural-gradient-based methods. More specifically, given an optimisation problem for which a direct application of natural gradient methods is not feasible, we transform the problem into one for which natural gradient methods can easily be applied. This chapter is largely based on the paper “*Optimising Distributions with Natural Gradient Surrogates*” (So and Turner, 2024a), which appeared at AISTATS in 2024.

5.1 Introduction

Problems that require the optimisation of probability distribution parameters are common across many scientific domains, and are of fundamental importance in the fields of machine learning and statistics. The goal of these optimisations is to minimise a function $f(\theta)$ where $\theta \in \Theta$ is the parameter vector of a probability distribution q .

Notable examples include maximum-likelihood estimation (MLE), variational inference (VI), and score matching (Hyvärinen, 2005).

When f is differentiable, the archetypal optimisation method is gradient descent (GD). GD takes steps in the direction in parameter space that gives the greatest decrease in f for an infinitesimally small step size, which is equivalent to following the direction of the negative gradient. While GD has many desirable properties, convergence can often be slow, and much effort has been spent in designing methods that offer improved convergence properties.

Natural gradient descent (NGD) – introduced in Chapter 3 – is one such method for optimising probability distribution parameters (Amari, 1998). In contrast with GD, NGD moves the parameters in the direction of steepest descent on a manifold of probability distributions, where steepness is defined with respect to a divergence measure between distributions. This notion of steepness is intrinsic to the manifold, and so it is not dependent on parameterisation, dispensing with a well-known pathology of GD.

There are some distributions for which the natural gradient can be computed efficiently. However, in its most general form, computing the natural gradient involves instantiating and inverting the Fisher information matrix, defined as

$$F(\theta) = \mathbb{E}_{q_\theta(x)} \left[\nabla_\theta (\log q_\theta(x)) \nabla_\theta (\log q_\theta(x))^\top \right], \quad (5.1)$$

posing several challenges. First, for some distributions, the Fisher can become singular, in which case the natural gradient is undefined. Second, it may not be available in closed form, in which case it must be estimated using sampling or other numerical methods. Finally, instantiating and inverting it explicitly has memory and computation costs that are quadratic and cubic, respectively, in the number of parameters. For even moderately large numbers of parameters, this can be prohibitively expensive.

In this chapter we propose a simple technique for tackling optimisation problems in which computing the natural gradient is problematic. Namely, we reframe the problem as an optimisation with respect to a *surrogate* distribution $\tilde{q}$, for which computing natural gradients is easy, and perform the optimisation in that space. The modified problem is equivalent in its solutions (provided certain conditions are met), but with a judiciously chosen surrogate, convergence can be rapid.

We find that a number of existing methods can be interpreted as applying this technique: the natural gradient VI method of Lin et al. (2019) for exponential family mixtures, stochastic natural gradient expectation propagation (Hasenclever et al., 2017), a fixed-point iteration scheme for optimising elliptical copulas (Hernández et al.,

2014), as well as the typical use of natural gradients in large-scale supervised learning settings. We also describe a new method for applying the technique to a wide variety of problems using exponential family (EF) surrogate distributions.

We present a number of experiments in which we find that our method significantly reduces the time to convergence compared to existing best-practice methods, both in number of iterations and wall-clock time. Our experiments consist of a variety of MLE and VI tasks, but we note that the method is applicable more generally to optimisations involving probability distributions. Our main contributions can be summarised as follows:

1. We propose a novel technique for optimising the parameters of probability distributions.
2. We prove the validity of this technique under stated conditions.
3. We find several examples of existing methods that can be viewed as applying this technique.
4. We describe a simple new method for applying it, using known properties of EF distributions.
5. We present a variety of MLE and VI experiments in which our method achieves significantly faster convergence than existing best-practice methods.

This chapter is structured as follows. In Section 5.2 we provide a brief overview of NGD and EF distributions. Section 5.3 covers items 1 and 4 of the contributions above, as well as an overview of item 2 (details found in Appendix C.1). Sections 5.4 and 5.5 cover items 5 and 3 respectively. Finally, in Section 5.7, we discuss limitations of our method, and avenues for future research.

5.2 Background

In this section we begin by briefly reviewing natural gradient descent, as well as the properties of exponential families that support efficient natural gradient computation. Natural gradient descent and exponential family distributions will together serve as a foundation for the method we introduce in Section 5.3.

5.2.1 Natural gradient descent

We now briefly summarise the natural gradient descent (NGD) method introduced in Chapter 3. NGD updates the parameters of a distribution q by taking a step proportional to the gradient of the objective f , preconditioned by the inverse of the Fisher matrix of q . That is, the update in parameters at step t of the optimisation is given by

$$\theta \leftarrow \theta - \epsilon_t [F(\theta)]^{-1} \nabla f(\theta) \quad (5.2)$$

where $F(\theta)$ is given by (5.1). The step size ϵ_t may follow some pre-defined schedule, or be found by line search. It can be shown that as $\epsilon_t \rightarrow 0^+$, update (5.2) moves θ in the direction of steepest descent in f , on the manifold of probability distributions spanned by q , where steepness is defined with respect to a divergence measure between distributions (Ollivier et al., 2017). Because this notion of steepness is intrinsic to the manifold, NGD is locally invariant to parameterisation.

When NGD is applied to online MLE problems, it is Fisher-efficient (Amari, 1998). For MLE problems more generally, it can be seen as a robust approximation to Newton's method (Martens and Grosse, 2015). These properties do not, in general, apply in other settings. Nevertheless, natural gradient methods have been applied in several other settings, often resulting in rapidly converging procedures (Kakade, 2001, Hoffman et al., 2013, Khan et al., 2017).

5.2.2 Exponential family distributions

We now briefly review the exponential families of distributions described in Chapter 2, whose properties support efficient natural gradient computation. The exponential family (EF) of distributions defined by the vector-valued statistic function $s(\cdot)$ and base measure $\nu(\cdot)$, has density (with respect to $\nu(\cdot)$)

$$q_\eta(x) = \exp \left(s(x)^\top \eta - A(\eta) \right), \quad (5.3)$$

where A is the log partition function, and η are the *natural parameters*. When the components of $s(\cdot)$ are linearly independent, the family is said to be *minimal*. There is an alternative parameterisation of q , given by

$$\mu(\eta) = \mathbb{E}_{q_\eta(x)}[s(x)], \quad (5.4)$$

where μ are known as the *mean parameters* of q . In minimal families the correspondence between η and μ is one-to-one, and we denote the reverse map as $\eta(\cdot)$.¹ In this thesis we only consider minimal EFs. Notable examples of EF distributions include the multivariate normal, gamma, categorical, and Dirichlet distributions to name but a few.

Let f_η and f_μ be functions related by $f_\eta = f_\mu \circ \mu$. It is a remarkable property of EFs that the natural gradients of f_η and f_μ with respect to η and μ , respectively, are given by

$$\tilde{\nabla} f_\eta(\eta) = \nabla f_\mu(\mu(\eta)) \quad (5.5)$$

$$\tilde{\nabla} f_\mu(\mu) = \nabla f_\eta(\eta(\mu)). \quad (5.6)$$

That is, the natural gradient in one parameterisation is simply given by the regular gradient with respect to the other (Malagò and Pistone, 2015). Provided that we have an efficient way of converting between natural and mean parameters, this allows us to compute the natural gradient without explicitly instantiating or inverting $F(\theta)$.

5.3 Surrogate natural gradient descent

Exponential families provide us with a relatively rich set of distributions for which we can efficiently compute natural gradients. However, there are many distribution families for which computing natural gradients remains difficult. In this section, we present details of our technique, which expands the set of distributions that we can efficiently target with natural gradients.

The main idea behind our technique is simple. When faced with an optimisation objective $f(\theta)$, where $\theta \in \Theta$ are the parameters of a distribution q for which computing the natural gradient update is problematic, we solve the problem in two steps. First, we re-parameterise f as a function of some other parameters $\tilde{\theta} \in \tilde{\Theta}$, related by $\theta = g(\tilde{\theta})$, and define the reparameterised objective

$$\tilde{f}(\tilde{\theta}) = f(g(\tilde{\theta})). \quad (5.7)$$

Second, we *interpret* $\tilde{\theta}$ as the parameters of a surrogate distribution $\tilde{q}$, for which computing natural gradients is easy, and then perform NGD in $\tilde{f}$ with respect to $\tilde{q}$

¹Recall that, due to our overloaded use of μ and η to denote both specific parameters and maps between parameterisations, we will use e.g. $\mu(\cdot)$ to disambiguate the latter if the intention is not clear from context.

and $\tilde{\theta}$. That is, we perform a sequence of updates

$$\tilde{\theta} \leftarrow \tilde{\theta} - \epsilon_t [\tilde{F}(\tilde{\theta})]^{-1} \nabla \tilde{f}(\tilde{\theta}), \quad (5.8)$$

where $\tilde{F}(\tilde{\theta})$ is the Fisher information matrix of $\tilde{q}$. We refer to q and $\tilde{q}$ as the *target* and *surrogate* distributions respectively. Update (5.8) has the straightforward interpretation of performing preconditioned GD in a reparameterised objective. Upon converging to a local minimiser $\tilde{\theta}^*$ of $\tilde{f}$, a solution to the original problem is obtained by $\theta^* = g(\tilde{\theta}^*)$. We call this technique *surrogate natural gradient descent* (SNGD).

As a simple example, when q is a multivariate Student's- t distribution with known degrees of freedom, we can choose $\tilde{q}$ to be multivariate normal. If $\tilde{\theta}$ is the mean and (vectorised) scale-matrix parameters of $\tilde{q}$, and θ the mean and covariance matrix of q , then a natural choice for g is simply the identity map on $\tilde{\Theta}$.

It is natural to question which properties SNGD shares with NGD under q . Given that SNGD is NGD (with respect to $\tilde{q}$), it remains locally invariant to parameterisation, and performs steepest descent in a statistical manifold (that of $\tilde{q}$). What is not guaranteed, is that the statistical manifold of $\tilde{q}$ remains useful for the optimisation of $\tilde{f}$. For example, in MLE settings, SNGD will not in general retain the asymptotic efficiency of NGD under q . However, what it gains is tractability, and as we demonstrate in Section 5.4, the practical performance benefits can be significant. In some cases, remarkably, SNGD can actually outperform NGD under q , with respect to both the original parameters θ , and the reparameterisation defined by $g(\cdot)$.

5.3.1 Choice of surrogate

The performance of SNGD relies crucially on appropriate choices for $\tilde{q}$ and $g(\cdot)$. A natural assumption would be that $\tilde{q}_{\tilde{\theta}}$ should be an approximation to $q_{g(\tilde{\theta})}$, and in some cases this can be a useful guide. However, often an effective surrogate can be found that has support over an entirely different space than that of q . In fact, this is true for most of the examples in this chapter, as can be seen in Table 5.1.

In Chapter 3 we discussed the interpretation of NGD, due to Ollivier et al. (2017), that it moves in the direction in parameter space that gives the greatest decrease in the objective within an infinitesimally small KL-divergence ball around the current iterate. SNGD therefore can be interpreted as moving in the direction in $\tilde{\theta}$ that gives the greatest decrease in $\tilde{f}$ within an infinitesimal KL-divergence ball for $\tilde{q}$ as opposed to q . This relationship is illustrated in Figure 5.1. A more general principle therefore is that a surrogate should be chosen such that $\tilde{\theta}$ has similar *local* effects on

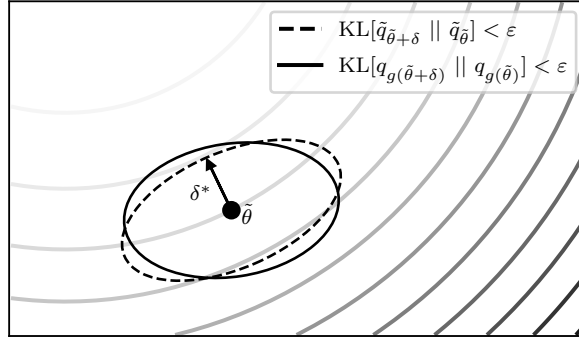


Fig. 5.1 Illustration of SNGD. An infinitesimal KL ball (ellipse, in parameter space) of radius ε around $\tilde{\theta}$ with respect to $\tilde{q}$ (dashed), and q (solid). Contours are of $\tilde{f}(\tilde{\theta})$. The SNGD direction, δ^* points toward the minimum point in the dashed region. In general this will differ from the direction of NGD in q , which would instead point toward the minimum point in the solid region (not marked).

Kullback-Leibler (KL) divergences in q than it does in $\tilde{q}$. More specifically, the effect of a small (infinitesimal) perturbation δ in $\tilde{\theta}$ should have a similar impact on the KL divergence from $\tilde{q}_{\tilde{\theta}+\delta}$ to $\tilde{q}_{\tilde{\theta}}$ as it does on that from $q_{g(\tilde{\theta}+\delta)}$ to $q_{g(\tilde{\theta})}$. That is, we would like:

$$\nabla_{\delta} \left(\text{KL}[\tilde{q}_{\tilde{\theta}+\delta} \parallel \tilde{q}_{\tilde{\theta}}] \right) \Big|_{\delta=0} = \nabla_{\delta} \left(\text{KL}[q_{g(\tilde{\theta}+\delta)} \parallel q_{g(\tilde{\theta})}] \right) \Big|_{\delta=0}. \quad (5.9)$$

If equality (5.9) is satisfied then the SNGD update for $\tilde{\theta}$ will be identical to the NGD update for (reparameterised) q . Finding tractable surrogates for which equality holds exactly will not typically be possible, but this motivates choosing $\tilde{q}$, $g(\cdot)$, for which it is approximately true. This perspective also highlights the reason that q and $\tilde{q}$ do not need to be distributions over the same space. All that matters is that the effect of (small) changes in $\tilde{\theta}$ on $\tilde{q}$ and q is similar in a KL-divergence sense. This perspective can help guide us toward a choice of $\tilde{q}$ and $g(\cdot)$. As a concrete example, let q_{θ} be a mixture distribution, with PDF given by

$$q_{\theta}(x) = \sum_{i=1}^k \pi_i q_{\theta_i}(x), \quad (5.10)$$

where q_{θ_i} are the component distributions, $\theta = (\pi_i, \theta_i)_{i=1}^k$, and $\pi \in \Delta^{k-1}$. Let us consider as a surrogate for q_{θ} the corresponding mixture *model* (joint distribution) with PDF

$$\tilde{q}_{\tilde{\theta}}(z, x) = \pi_z q_{\theta_z}(x), \quad (5.11)$$

where $\tilde{\theta} = (\pi_i, \theta_i)_{i=1}^k$, so that g is simply the identity map, and $\tilde{q}_{\tilde{\theta}}(x) = \sum_{i=1}^k \tilde{q}_{\tilde{\theta}}(z = i, x) = q_{g(\tilde{\theta})}(x)$. Using standard identities, we have

$$\begin{aligned} \nabla_{\delta} \left(\text{KL} \left[\tilde{q}_{\tilde{\theta}+\delta}(z, x) \parallel \tilde{q}_{\tilde{\theta}}(z, x) \right] \right) &= \nabla_{\delta} \left(\text{KL} \left[\tilde{q}_{\tilde{\theta}+\delta}(x) \parallel \tilde{q}_{\tilde{\theta}}(x) \right] \right. \\ &\quad \left. + \mathbb{E}_{\tilde{q}_{\tilde{\theta}+\delta}(x)} \left[\text{KL} \left[\tilde{q}_{\tilde{\theta}+\delta}(z|x) \parallel \tilde{q}_{\tilde{\theta}}(z|x) \right] \right] \right) \\ &= \nabla_{\delta} \left(\text{KL} \left[q_{g(\tilde{\theta}+\delta)}(x) \parallel q_{g(\tilde{\theta})}(x) \right] \right. \\ &\quad \left. + \mathbb{E}_{q_{\tilde{\theta}+\delta}(x)} \left[\text{KL} \left[\tilde{q}_{\tilde{\theta}+\delta}(z|x) \parallel \tilde{q}_{\tilde{\theta}}(z|x) \right] \right] \right) \end{aligned} \quad (5.12)$$

We can see that (5.9) will be true when the red term in (5.12) has gradient (with respect to δ) equal to zero. In general however, this will not be true, and using a mixture model joint, as a surrogate for its marginal, can be viewed as using an approximate metric which includes an additional term. This term imposes an additional penalty on directions in parameter space that affect the expected *responsibility* distributions.² In the experiments of Section 5.4.4 we take this approximation one step further, and use EF mixture models as surrogates for mixtures with components that are *not* EF distributions.

5.3.2 Equivalence with optimisation of f

In Appendix C.1 we prove a number of results regarding the validity of SNGD, which we summarise here. Let Θ and $\tilde{\Theta}$ be open subsets of $\mathbb{R}^i$ and $\mathbb{R}^j$ respectively, with $i \leq j$. Let $g : \tilde{\Theta} \rightarrow \Theta$ be twice differentiable, with Jacobian of rank i everywhere, and with $g(\tilde{\Theta}) = \Theta$. Finally, let $f : \Theta \rightarrow \mathbb{R}$ be twice continuously differentiable, and define $\tilde{f} = f \circ g$. Under these conditions, optimising $\tilde{f}$ is equivalent to optimising f in the following sense: finding a local minimiser of $\tilde{f}$ also gives us a local minimiser for f , and all local minima of f are attainable through $\tilde{f}$. Furthermore, $\tilde{f}$ does not have any non-strict saddle points that are not also present at the corresponding points in f , and so $\tilde{f}$ does not introduce any additional spurious attractors (Lee et al., 2016).

5.3.3 Algorithms

By choosing $\tilde{q}$ from the set of exponential family (EF) distributions described in Chapter 2, we can perform update (5.8) efficiently without explicitly instantiating or

²The responsibility of component i for x is the probability that x was generated by component i having observed x .

inverting $\tilde{F}(\tilde{\theta})$. Algorithm 1 provides pseudocode for an implementation of SNGD when $\tilde{\Theta}$ is either the mean or natural domain of an EF. It assumes the existence of an autodiff operator **grad**, and an overloaded function **dualparams**, which maps from natural to mean parameters, or vice-versa, depending on the type of its argument. We use $\tilde{f}_{\text{dual}}$ to denote the reparameterisation of $\tilde{f}$ in terms of the dual parameters – that is, if $\tilde{\theta}$ are natural parameters, $\tilde{f}_{\text{dual}}$ is a function of the corresponding mean parameters, and vice-versa. A full explanation of Algorithm 1 can be found in Appendix C.2.

Algorithm 1 SNGD with EF surrogate

Require: objective $f : \Theta \rightarrow \mathbb{R}$
Require: parameter mapping $g : \tilde{\Theta} \rightarrow \Theta$
Require: initial surrogate parameters $\tilde{\theta} \in \tilde{\Theta}$
Require: step size schedule $\{\epsilon_t \in \mathbb{R}_+ : t = 0, 1, \dots\}$
 $\tilde{f}_{\text{dual}}(\cdot) := f(g(\text{dualparams}(\cdot)))$
 $t \leftarrow 0$
while not converged **do**
 $\tilde{\nabla} \leftarrow \text{grad}[\tilde{f}_{\text{dual}}](\text{dualparams}(\tilde{\theta}))$
 $\tilde{\theta} \leftarrow \tilde{\theta} - \epsilon_t \tilde{\nabla}$
 $t \leftarrow t + 1$
return $g(\tilde{\theta})$

The domains of valid natural and mean parameters of an EF are open convex sets. It is possible that finite-length NGD steps in these parameters can move outside of the valid domain. These steps can be corrected by using a backtracking linesearch, an approach taken in previous work on natural gradients. In our experiments, we found that this typically occurs so infrequently as to have a relatively small impact on performance. We do not expect this to always be the case however, and can be highly problem dependent. We note that specific efforts have been made to address this issue without backtracking, such as that of Lin et al. (2020).

In some cases, a chosen surrogate’s parameters, $\tilde{\theta}$, may not be sufficient to fully specify θ . To handle such cases, we can generalise SNGD using a partitioning of the parameters $\tilde{\theta} = (\tilde{\theta}_{\text{ng}}, \tilde{\theta}_{\text{fg}})$, where $\tilde{\theta}_{\text{ng}} \in \tilde{\Theta}_{\text{ng}}$ will be optimised using natural gradients, and $\tilde{\theta}_{\text{fg}} \in \tilde{\Theta}_{\text{fg}}$ with flat gradients. For example, we can apply the sequence of updates

$$\tilde{\theta}_{\text{ng}} \leftarrow \tilde{\theta}_{\text{ng}} - \epsilon_t [\tilde{F}(\tilde{\theta}_{\text{ng}})]^{-1} \nabla_{\tilde{\theta}_{\text{ng}}} \tilde{f}(\tilde{\theta}), \quad (5.13)$$

$$\tilde{\theta}_{\text{fg}} \leftarrow \tilde{\theta}_{\text{fg}} - \varepsilon_t \nabla_{\tilde{\theta}_{\text{fg}}} \tilde{f}(\tilde{\theta}), \quad (5.14)$$

where $\epsilon_t, \varepsilon_t$ are the step sizes for $\tilde{\theta}_{\text{ng}}, \tilde{\theta}_{\text{fg}}$, respectively at time t . Update (5.14) corresponds to GD in $\tilde{\theta}_{\text{fg}}$, but we may equally use any other first-order optimiser.

Revisiting our earlier example, if q is a multivariate Student's- t distribution, but now with *unknown* degrees of freedom ν , we can again directly map from $\tilde{\theta}_{\text{ng}}$, which contains the location and scale matrix parameters of $\tilde{q}$, to the mean and covariance of q . ν has no analogue in the multivariate normal distribution however, and so we can capture this with $\tilde{\theta}_{\text{fg}} = (\nu)$. $g(\cdot)$ can then simply be the identity map on $\tilde{\Theta} = \tilde{\Theta}_{\text{ng}} \times \tilde{\Theta}_{\text{fg}}$.³

Algorithm 2 SNGD with EF surrogate and parameter separation

Require: objective $f : \Theta \rightarrow \mathbb{R}$
Require: parameter mapping $g : \tilde{\Theta} \rightarrow \Theta$
Require: initial surrogate parameters $\tilde{\theta}_{\text{ng}} \in \tilde{\Theta}_{\text{ng}}$
Require: initial auxiliary parameters $\tilde{\theta}_{\text{fg}} \in \tilde{\Theta}_{\text{fg}}$
Require: $\tilde{\theta}_{\text{ng}}$ step size schedule $\{\epsilon_t \in \mathbb{R}_+ : t = 0, 1, \dots\}$
Require: $\tilde{\theta}_{\text{fg}}$ step size schedule $\{\varepsilon_t \in \mathbb{R}_+ : t = 0, 1, \dots\}$
1: $\tilde{f}_{\text{dual}}(_1, _2) := f(g(\text{dualparams}(_1), _2))$
2: $t \leftarrow 0$
3: **while** not converged **do**
4: $(\tilde{\nabla}_{\text{ng}}, \nabla_{\text{fg}}) \leftarrow \text{grad}[\tilde{f}_{\text{dual}}](\text{dualparams}(\theta_{\text{ng}}, \tilde{\theta}_{\text{fg}})$
5: $\tilde{\theta}_{\text{ng}} \leftarrow \tilde{\theta}_{\text{ng}} - \epsilon_t \tilde{\nabla}_{\text{ng}}$
6: $\tilde{\theta}_{\text{fg}} \leftarrow \tilde{\theta}_{\text{fg}} - \varepsilon_t \nabla_{\text{fg}}$
7: $t \leftarrow t + 1$
8: **return** $g((\tilde{\theta}_{\text{ng}}, \tilde{\theta}_{\text{fg}}))$

Pseudocode for this generalisation is given in Algorithm 2. The structure is largely the same as that of Algorithm 1. One notable difference is that $\tilde{f}_{\text{dual}}$ now has 2 arguments, and so the call to **grad** on line 4 returns a function that returns a 2-tuple of gradients, one for each argument. Algorithm 2 uses GD with a fixed learning rate schedule to optimise $\tilde{\theta}_{\text{fg}}$, but the extension to any first-order optimiser is straightforward. This generalisation expands the set of distributions that can be targeted by SNGD, and we make use of it in the experiments of Sections 5.4.2 and 5.4.3.

5.4 Evaluation

In this section we present a number of experiments demonstrating the utility of SNGD. We used a variety of MLE and VI tasks as test cases.

In MLE the goal is to find parameters θ of distribution q that maximise the (log) likelihood of observed data $\{x_i\}_{i=1}^n$ under q . That is, the objective function for MLE

³In practice it would be preferable to choose $\tilde{\theta}_{\text{fg}} = (\log \nu)$, with $g(\cdot)$ adjusted accordingly, so that optimisation of $\tilde{\theta}_{\text{fg}}$ is unconstrained.

Table 5.1 Summary of the surrogate-target pairs appearing in this chapter. For the sake of readability we have omitted the conventional multivariate prefix from distribution names, as this information is conveyed by the support. Parameter mappings for these examples are found in Appendix C.4.

Target		Surrogate	
Distribution	Support	Distribution	Support
Negative binomial	$\mathbb{N}$	Gamma	$\mathbb{R}_+$
Negative binomial mixture	$\mathbb{N}$	Gamma mixture model	$\mathbb{R}_+ \times \{1, \dots, k\}$
Skew-normal	$\mathbb{R}^d$	Normal	$\mathbb{R}^d$
Skew-normal mixture	$\mathbb{R}^d$	Normal mixture model	$\mathbb{R}^d \times \{1, \dots, k\}$
Skew- t	$\mathbb{R}^d$	Normal	$\mathbb{R}^d$
Elliptical copula	$[0, 1]^d$	Zero-mean normal	$\mathbb{R}^d$

tasks is of the form

$$f(\theta) = - \sum_{i=1}^n \log q_{\theta}(x_i). \quad (5.15)$$

In VI we are given a generative model $p(z)p(\mathcal{D} \mid z)$, and observed data $\mathcal{D}$, and the goal is to find parameters θ of distribution q that minimise the Kullback-Leibler (KL) divergence from q to the posterior $p(z|\mathcal{D})$. The objective function for VI tasks is of the form

$$f(\theta) = -\mathbb{E}_{q_{\theta}(z)} \left[\log \frac{p(\mathcal{D}, z)}{q_{\theta}(z)} \right]. \quad (5.16)$$

It can be shown that minimising (5.16) is equivalent to maximising a lower bound on the log marginal likelihood: $\log p(\mathcal{D})$. In the VI experiments of this section, gradients of (5.16) were estimated by applying the reparameterisation trick as described in Section 4.3.1 (Kingma and Welling, 2014).

In each experiment, we compared SNGD with a number of baselines. For tasks that were small scale, and had objectives that could be computed deterministically, we compared SNGD with two baselines: GD and BFGS (Nocedal and Wright, 2006). In these cases, we used exact line search to determine step sizes for each method, in order to compare search direction independent of hyperparameter settings. For larger tasks, or where only stochastic estimates of f were available (minibatched or VI experiments), we used Adam (Kingma and Ba, 2014) as the standard baseline, and used grid search to determine the best hyperparameters for each method. Further

details of all experiments are given in Appendix C.3. A summary of surrogate-target pairs used for SNGD in these experiments can be found in Table 5.1, and details of parameter mappings can be found in Appendix C.4.

We display average training curves for each experiment, with error bars corresponding to 2 standard errors, computed across 10 random seeds. In Appendix C.5 we provide Pareto frontiers, showing the best performance achieved by *any* hyperparameter setting, as a function of both iteration count and wall-clock time. In this section we use n to denote the number of training data points, and d for the dimensionality of the random variable under q . For brevity, we omit the conventional multivariate prefix from distribution names in this section, as this is implicit for $d > 1$.

5.4.1 Negative binomial distribution

The negative binomial distribution is widely used for modelling discrete data (Fisher, 1941, Lloyd-Smith et al., 2005, Lloyd-Smith, 2007, Orooji et al., 2021, Kendall et al., 2023). Maximum likelihood estimates of the negative binomial parameters are not available in closed form, and must be found numerically.

Let $\theta = (r, s)$ be the parameters of the negative binomial distribution with probability mass function

$$q(x) = \binom{x+r-1}{x} (1-s)^x s^r. \quad (5.17)$$

The gamma distribution can be seen as a continuous analogue of the negative binomial, and as it is a EF, it is a natural choice of surrogate when targeting the negative binomial with SNGD. Let $\tilde{q}$, therefore, be the gamma distribution with parameters α , β and probability density function

$$\tilde{q}(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp^{-\beta x}. \quad (5.18)$$

Guenther (1972) used the gamma distribution to approximate the CDF of the negative binomial, using the mapping $(r, s) = (\alpha/(1-\beta), \beta)$. Let g be the equivalent mapping, but defined in terms of *mean* parameters of $\tilde{q}$. To ensure that r and s correspond to valid negative binomial parameters, we restrict Θ to the subset of gamma mean parameters for which $\beta < 1$.⁴ Finally, let f be defined as (5.15), where $\{x_i\}_{i=1}^n$ is the

⁴It can be shown that Θ remains an open convex set.

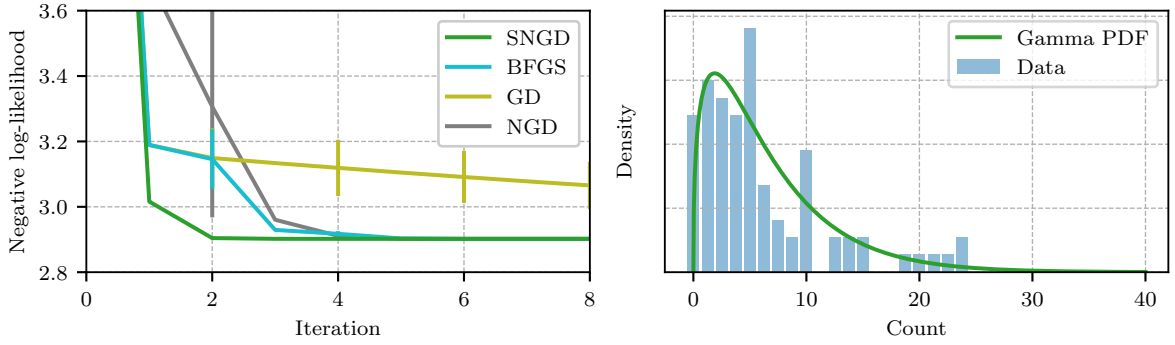


Fig. 5.2 Negative binomial MLE on the sheep dataset. (left) Training curves. (right) Histogram of the observed counts, overlaid with the PDF of the gamma surrogate of SNGD at convergence.

dataset of Fisher (1941), consisting of counts of ticks observed on a population of sheep ($n=82$, $d=1$).

Using these definitions, we applied Algorithm 1, modified to use a line search as discussed at the start of this section. We compared SNGD with our standard baselines: GD and BFGS. Additionally, d was sufficiently small in this task that we were able to compare with NGD under q by estimating and inverting the Fisher directly. Figure 5.2 shows that SNGD outperformed all of the baselines. Remarkably, this included NGD under q . This surprising result is explored in detail in Appendix C.6. There, we show that this was true even when NGD was performed with respect to the reparameterisation defined by $g(\cdot)$, implying that this is not simply an artefact of SNGD using a more effective parameterisation, rather, it is the change of parameterisation *and* distribution that results in such rapid convergence. We stress however that we do not expect this to be a general feature of SNGD, but just highlight it to show that the approximations involved do not necessarily result in a less efficient optimisation procedure when compared to exact NGD.

5.4.2 Skew-elliptical distributions

The skew-elliptical distributions are a flexible family of unimodal multivariate distributions, allowing for features such as asymmetry and heavy tails (Azzalini, 2013). In this section we consider in particular the skew-normal and skew- t distributions. The skew-normal can be viewed as an asymmetric generalisation of the normal distribution. Similarly, the skew- t can be viewed as an asymmetric generalisation of the Student's t distribution.

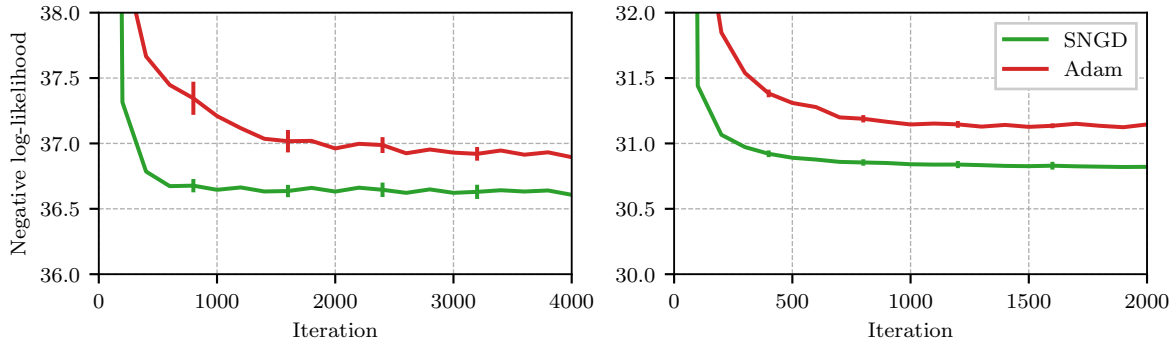


Fig. 5.3 Training curves for MLE on the minibooone dataset ($n=32,840$, $d=43$) using (left) skew-normal, and (right) skew- t distributions.

The normal distribution is a natural surrogate for these distributions. We used it as such in a number of experiments, mapping its mean and covariance to parameters playing similar roles in the target distributions, with any additional parameters optimised using Adam. We performed two tasks on real data in our experiments. In the first, the skew-elliptical distributions were used for density estimation on the UCI minibooone dataset ($n=32,840$, $d=43$) (Roe, 2010), fitting their parameters via MLE. In the second, they were used as approximate posteriors in VI for a Bayesian logistic regression model on the UCI coverytype dataset (Blackard, 1998). In this task we used a small subsample of observations in order to retain some uncertainty in the posterior ($n=500$, $d=53$). We also performed a further MLE experiment using high-dimensional synthetic data ($n=10,000$, $d=1,000$).

Note that the numbers of free parameters in these optimisations were $O(d^2)$ due to the covariance-matrix-like parameters of the target distributions. The skew-normal distribution had 1,032, 1,537, and 502,500 parameters in the minibooone, coverytype, and synthetic experiments, respectively. For the skew- t distribution those numbers were 1,033, 1,538, and 502,501.

Figures 5.3, 5.4 and 5.5 show training curves from the minibooone MLE, coverytype VI, and synthetic MLE experiments, respectively. In all cases SNGD significantly outperformed Adam. For the skew-normal VI experiment, it was also possible to apply the minimal conditional exponential family (MCEF) natural gradient VI method of Lin et al. (2019) – introduced in Section 4.3.4 – and so we included this as an additional baseline; its performance was virtually identical to that of SNGD, however, we note that this method can itself be viewed as applying SNGD, a point we discuss further in Section 5.5. We note that it is possible that other distributions in this chapter could also be optimised using the method of Lin et al. (2019) if they could be expressed as the marginal of a MCEF distribution, but finding such cases is not trivial, and the

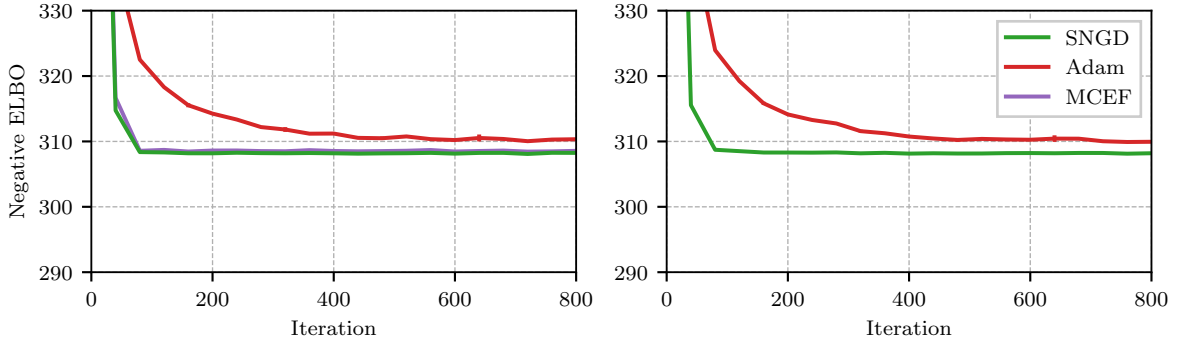


Fig. 5.4 Training curves for Bayesian logistic regression VI on the coverytype dataset ($n=500$, $d=53$) using (left) skew-normal, and (right) skew- t approximations. MCEF corresponds to the natural gradient VI method of Lin et al. (2019).

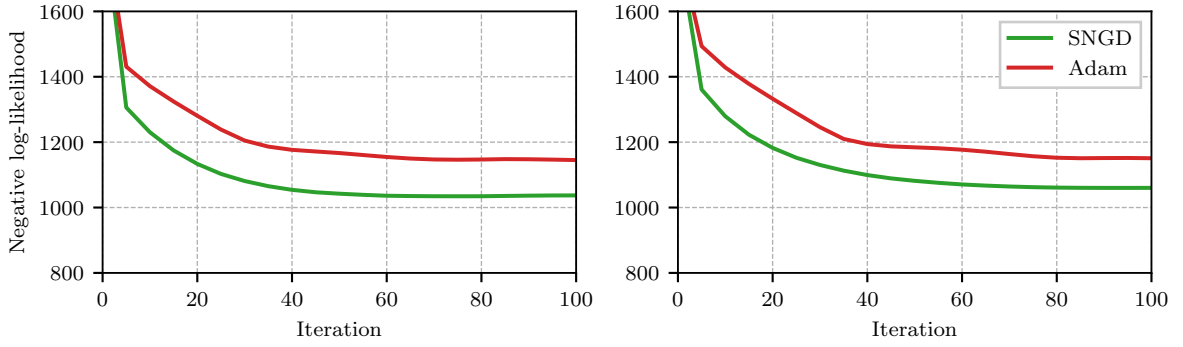


Fig. 5.5 Training curves for MLE on a synthetic dataset ($n=10,000$, $d=1,000$) using (left) skew-normal, and (right) skew- t distributions.

other distributions considered here are not addressed by the examples given in that work.

5.4.3 Elliptical copulas

Copula models define a distribution on the unit hypercube $[0, 1]^d$, for which each of the marginals is uniform on $[0, 1]$. If $x \in [0, 1]^d$ is distributed according to copula q , and $y_i = P_i^{-1}(x_i)$, where P_i^{-1} is the inverse CDF of a distribution q_i , then y_i will have marginal distribution q_i . The y_i will be dependent in general, with dependence structure determined by q . Copula models are widely used in finance for modelling high-dimensional variables, in part because they allow marginal distributions to be modelled separately from the dependence structure (Dewick and Liu, 2022).

The class of elliptical copulas are defined as those copulas which can be used to generate elliptical distributions (Frahm et al., 2003). Common examples are the Gaussian and t copulas. Elliptical copulas are parameterised by a correlation matrix

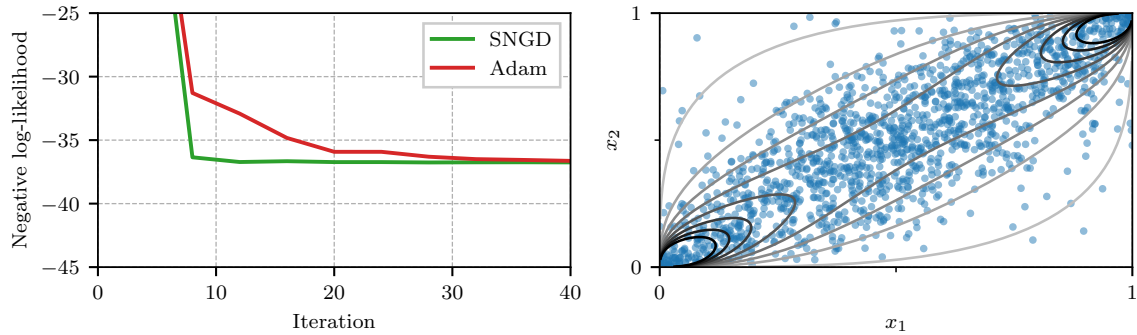


Fig. 5.6 t -copula MLE using 5 years of daily stock return data ($n=1,515$, $d=93$). (left) Training curves. (right) Contours of a 2D marginal density from the fitted copula, overlaid with the training data.

R , as well as a (possibly empty) set of additional parameters. A simple way to target elliptical copulas with SNGD is to use a zero-mean normal surrogate, mapping its correlation matrix to R , and using standard first-order methods to optimise any additional parameters. We used this approach to perform MLE of the t -copula on 5 years of daily stock returns from the FTSE 100 universe ($n=1,515$, $d=93$), with marginals estimated as (univariate) Student's t distributions. In this experiment q had 4,279 free parameters. Figure 5.6 shows that SNGD converged significantly faster than Adam.

Interestingly, when applied in this way, the SNGD update for R is equivalent to that of a fixed-point iteration scheme from the quantitative finance literature (Hernández et al., 2014), although that method was derived using an entirely different approach. Hernández et al. (2014) were unable to prove convergence of their method, but convergence of SNGD follows as a consequence of the usual stochastic optimisation guarantees (Robbins and Monro, 1951).

5.4.4 Mixture distributions

A mixture distribution expresses a complicated density as a convex combination of k simpler densities:

$$q(x) = \sum_{i=1}^k \pi_i q_{\theta_i}(x), \quad (5.19)$$

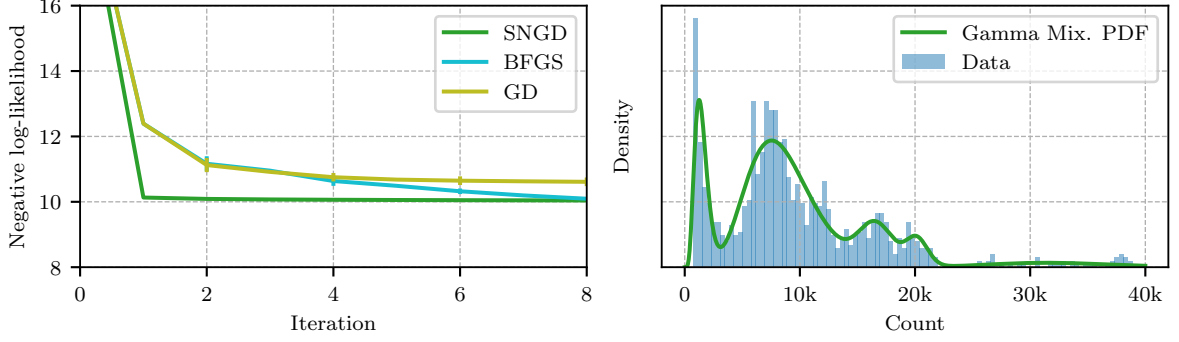


Fig. 5.7 Negative binomial mixture MLE ($k=5$) using the number of daily COVID hospital admissions in the UK over a 3 year period ($n=1,120$, $d=1$). (left) Training curves. (right) Histogram of the observed counts, overlaid with the PDF of (a marginal of) the gamma mixture model surrogate of SNGD at convergence.

where $\pi \in \Delta^{k-1}$, and q_{θ_i} is known as the i -th *component* distribution. For each mixture, we can also define a corresponding mixture *model*, a joint distribution, with density

$$q(z, x) = \pi_z q_{\theta_z}(x), \quad (5.20)$$

where the mixture component identity, $z \in \{1, \dots, k\}$, is treated as a random variable. Note then, that $q(x) = \sum_{i=1}^k q(z = i, x)$. That is, a mixture model has a mixture as its *marginal*. A mixture of EFs is not in general itself an EF. However, it can be shown that the corresponding mixture model (joint distribution) *is* an EF (see Appendix B.3).

When an EF is an appropriate surrogate for applying SNGD to a given target family, it is therefore straightforward to consider the extension to mixtures of that family by using an EF mixture *model* as a surrogate. We emphasise that in doing so, the surrogate distribution has support over more variables (z and x) than the target distribution (just x).

In Section 5.4.1 we demonstrated the use of a gamma surrogate for optimising negative binomial distribution parameters with SNGD. It is therefore straightforward to use a gamma mixture model to target a negative binomial mixture. As an experiment, we performed MLE of a 5 component negative binomial mixture, using a dataset consisting of the number of daily COVID-19 hospital admissions in the UK over a 3 year period ($n=1,120$, $d=1$). Figure 5.7 shows that SNGD significantly outperformed the baselines.

As a further example, similarly, we can use a normal mixture model as the surrogate for a skew-normal mixture. Figure 5.8 shows the qualitative improvement a skew-normal mixture can offer over a normal mixture on the Bayesian logistic regression VI

task of Murphy (2012) ($n=60$, $d=2$). Figure 5.9 shows training curves for this task, as well as for the UCI covertype VI task of Section 5.4.2. In the latter, q had 4,613 and 7,689 free parameters for $k=3$, 5, respectively.

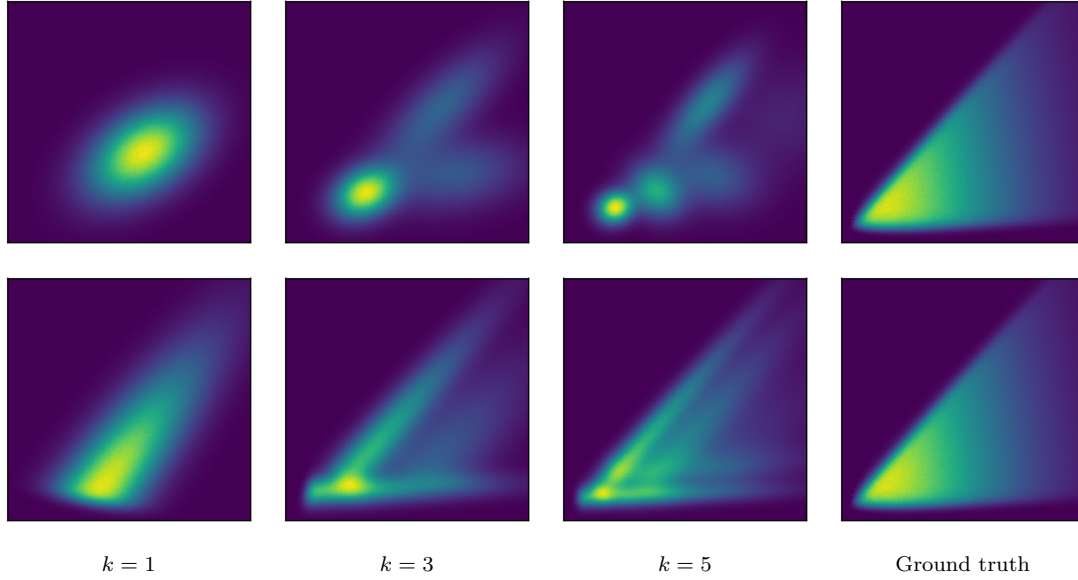


Fig. 5.8 Comparison of (top) normal and (bottom) skew-normal mixture posterior approximations in the Bayesian logistic regression VI task of Murphy (2012). Ground truth generated using MCMC samples.

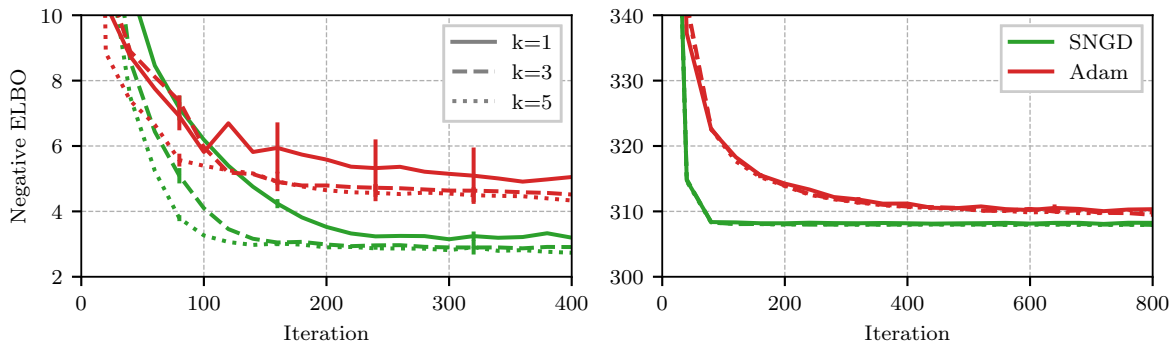


Fig. 5.9 Training curves for Bayesian logistic regression VI with a skew-normal mixture approximation. (left) the toy dataset of Murphy (2012) ($n=60$, $d=2$), and (right) UCI covertype ($n=500$, $d=53$). k is the number of mixture components.

5.5 Related work

Several existing methods can be interpreted as examples of the general technique presented in this chapter.

Lin et al. (2019) introduced the class of MCEF distributions, and showed that natural gradients with respect to a particular parameterisation of MCEF distributions can be efficiently computed using an identity analogous to (3.10). They also showed that several standard distributions can be expressed as *marginals* of MCEF distributions, including the multivariate Student's- t and multivariate skew-normal distributions. They then used NGD with respect to the MCEF joint as a surrogate for its marginal in a number of VI tasks. SNGD is more general in that it makes no such assumption about the relationship between surrogate and target distributions. In particular, it is not restricted to targets that can be expressed as the marginal of a MCEF (or any other) distribution.

Another closely related method is that of Lin et al. (2021). In that work, the authors develop a method for performing NGD in *structured* parameter spaces, a setting which would ordinarily be challenging. Their method performs NGD with respect to *local co-ordinates* at each step of the algorithm, which are carefully chosen in such a way that the updates are both tractable, and obey the desired structural constraints. While the base method in their work is one of reparametrisation, as opposed to change of distribution, they also show applications that can also be viewed as combining the latter. For example, they extend their approach to mixture distributions by using the same MCEF surrogates as described above. They also target matrix-Gaussian distributions by using a block-diagonal approximation of the Fisher information matrix, which could be viewed as performing NGD with respect to a factorised distribution in which groups of parameters do not interact.

The stochastic natural gradient expectation propagation (SNEP) algorithm of Hasenclever et al. (2017) solves a saddle-point optimisation problem, with the inner optimisation being over a set of parameters $\{\lambda_i\}$ known as the *site parameters*. The site parameters jointly parameterise a set of distributions for which computing natural gradients is not straightforward. SNEP instead treats each λ_i as if it were the natural parameter vector of a EF distribution in its own right, and performs NGD in the dual (mean parameter) space. These (pseudo)distributions acted as surrogates for the more problematic set of distributions that were the ultimate targets of the optimisation.

Hernández et al. (2014) devised a fixed-point iteration scheme for optimising the correlation matrix parameter of an elliptical copula. Interestingly, although their procedure was motivated in an entirely different way, the resulting updates for R are

identical to those performed by SNGD when applied in the manner described in Section 5.4.3.

In supervised learning, the goal is to model the conditional density $p(y|x)$ given training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$. In this setting the Fisher matrix is usually defined with respect to the distribution $p_{\mathcal{D}}(x)q_{\theta}(y|x)$, where $p_{\mathcal{D}}(x)$ is the empirical distribution of x under $\mathcal{D}$. In practice, when the data set is large, the Fisher must be computed with respect to some subset of the training inputs $\mathcal{S}$, such as the current minibatch. This is equivalent to using $p_{\mathcal{S}}(x)q_{\theta}(y|x)$ as a surrogate for $p_{\mathcal{D}}(x)q_{\theta}(y|x)$, where $p_{\mathcal{S}}(x)$ is the empirical distribution of x in $\mathcal{S}$. In the context of supervised deep learning, Ren and Goldfarb (2019) showed that exact natural gradients with respect to $p_{\mathcal{S}}(x)q_{\theta}(y|x)$ can be computed efficiently when $|\mathcal{S}|$ is small.

Several works have attempted to handle intractability of the natural gradient in other ways. Some methods are based on structured approximations to the Fisher, that allow inverse-vector products to be computed efficiently (Heskes, 2000, Martens and Grosse, 2015, George et al., 2018). Garcia et al. (2023) instead approximated the inverse Fisher matrix directly by expressing it in terms of the solution to an optimisation problem. The widely used Adam optimiser employs a moving average of squared gradients in a diagonal approximation to the Fisher (Kingma and Ba, 2014), although it has been cautioned that this is more accurately viewed as an approximation to the *empirical* Fisher (Kunstner et al., 2019).

Another work which extends the use of natural gradients beyond exponential families is that of Kiral et al. (2023), which can be seen as an extension of the Bayesian learning rule of Khan and Rue (2023), performing NGD in a Lie group of probability distributions. Among other benefits, this can in principle be applied to arbitrary probability distributions. In practice, however, computation of the Fisher, and the group’s exponential map, are not straightforward in all cases.

SNGD is also related to the methods of stochastic relaxation (Malagò et al., 2009, Staines and Barber, 2012), evolution strategies (Rechenberg and Eigen, 1973), and estimation of distribution algorithms (Baluja and Caruana, 1995, Larrañaga and Lozano, 2002, Pelikan et al., 2002). What these methods have in common is that they perform optimisation by sampling candidate solutions from a *search distribution* – a probability distribution of promising solutions – that is updated over time as more information about the objective is obtained. They can then be viewed as transforming the original optimisation problem into one over probability distributions, with a solution to the former being obtained from the solution to the latter. Furthermore, methods have been proposed in each of these areas that update the search distribution using

natural gradients (Malagò and Pistone, 2015, Malagò et al., 2011, Wierstra et al., 2014, Ollivier et al., 2017, Khan et al., 2018). In that way, the similarity with our method becomes clear – an optimisation in one space is performed by using natural gradient updates in another. The crucial difference however is the way in which the distributions are used. The search distributions of stochastic optimisation methods are used to maintain distributions over *solutions* to the optimisation problem, which are then used to sample promising candidates. In SNGD, the surrogate distributions could be viewed as existing solely to provide a convenient preconditioner for the original optimisation problem. This also implies that the two types of distributions operate on very different spaces. For an optimisation objective $f(\theta)$, where θ are the parameters of a probability distribution $q(x)$, the stochastic search methods would employ a search distribution over Θ , whereas in SNGD, the surrogate distribution can be over any space, but is likely to be more closely related to $\mathcal{X}$ than Θ . As an example, note that if the $q_\theta(x)$ is a skew-elliptical distribution over $x \in \mathbb{R}^d$, then θ will contain covariance-matrix-like parameters. A search distribution over Θ would then be (in part) a distribution over the space of positive-definite matrices, whereas in the experiments of Section 5.4, our surrogates for the elliptical distributions are simply Gaussians on $\mathbb{R}^d$. Furthermore, whereas search distributions are used to sample candidate solutions, doing so with the surrogate distributions of SNGD would make little sense, as there would be no obvious connection between such samples and candidate solutions.

5.6 Future work

The main limitation of SNGD is the need to find a suitable surrogate and reparameterisation for a given target distribution. In some cases there is an obvious candidate, such as when an EF distribution could serve as an approximation for the target, or when the target can be viewed as an EF that has been warped or transformed in some way. However, for most of the examples in this chapter, the target and surrogate distributions do not even have support over the same space (see Table 5.1). Our choices of surrogates were largely guided by the intuition that $\tilde{\theta}$ should have similar local effects on KL divergences in q than it does in $\tilde{q}$. However, finding more prescriptive or systematic methods for choosing surrogates would be beneficial, and may be an interesting direction for future work.

5.7 Summary

In this chapter we proposed a novel technique – *surrogate natural gradient descent* (SNGD) – for optimising the parameters of probability distributions: reframing the objective as an optimisation with respect to a *surrogate* distribution for which computing natural gradients is easy, and performing optimisation in that space. SNGD extends the set of distributions that can be targeted with natural gradients, going beyond just exponential families and minimal conditional exponential families. We found several existing methods that can be interpreted as applying this technique, and proposed a new method based on EF surrogates. We demonstrated that our method is able to converge rapidly on a variety of MLE and VI tasks. SNGD can be readily applied to distributions outside of the set of examples given here, but we also expect that new methods can be found by applying the more general technique that motivated our method.

Chapter 6

Fearless stochasticity in expectation propagation

Expectation propagation (EP) is a popular family of algorithms for performing approximate Bayesian inference (Opper and Winther, 2000, Minka, 2001). When it can be applied, it often converges rapidly, yielding results far faster than MCMC. The classic EP of Minka (2001) is often found to have better calibrated uncertainties than VI. Furthermore, its more general variant, power EP, allows for smooth interpolation between the characteristics of (classic) EP and VI, with intermediate choices typically yielding better results on downstream tasks (Bui et al., 2017).

In this chapter, we develop new variants of EP using natural-gradient-based methods, which have significant advantages over existing variants in settings where the moments are estimated with noise, including *black-box* settings, which potentially enables EP to be used on a much wider class of problems than was previously possible. This chapter is largely based on the paper “*Fearless Stochasticity in Expectation Propagation*” (So and Turner, 2024b), which appeared with a spotlight award at NeurIPS in 2024.

6.1 Introduction

Expectation propagation (EP) (Minka, 2001, Opper and Winther, 2000) is a family of algorithms that is primarily used for performing approximate inference in probabilistic models (Bui et al., 2016, 2017, Chu and Ghahramani, 2005, Cunningham et al., 2008, Gonzalez et al., 2016, Hennig, 2009, Hennig et al., 2010, Hennig and Schuler, 2012, Herbrich et al., 2006, Hernandez-Lobato and Hernandez-Lobato, 2016, Hernández-Lobato and Adams, 2015, Heskes and Zoeter, 2002, Højen-Sørensen et al., 2002, Jitkrittum et al., 2015, Jylänki et al., 2011, Li et al., 2015, Minka and Lafferty, 2002,

Minka et al., 2018, Naish-guzman and Holden, 2007, Nodelman et al., 2005, Opper et al., 2008, Seeger and Nickisch, 2011, Turner and Sahani, 2011, Vehtari et al., 2020, Welling and Lim, 2007, Xu et al., 2014, Zoeter and Heskes, 2005), although it can be used more generally for approximating certain kinds of functions and their integrals (Cunningham et al., 2013).

EP involves the evaluation of moments—expectations of certain functions—under distributions that are derived from the model of interest. EP is usually applied to models for which these moments have convenient closed-form expressions or can be accurately estimated using deterministic methods. Moments can also be estimated using Monte Carlo (MC) samples, significantly expanding the set of models EP can be applied to. However, the updates of EP are not robust to MC noise when performed naively, and various prior works have attempted to address this issue in different ways (Hasenclever et al., 2017, Vehtari et al., 2020, Xu et al., 2014).

In this chapter we provide a novel perspective on the moment-matching updates of EP; namely, that they perform natural-gradient-based optimization of a variational objective (Section 6.3). We use this insight to motivate two new EP variants, EP- η (Section 6.3.2) and EP- μ (Section 6.3.3), with updates that are particularly well-suited to MC estimation, remaining stable and being most sample-efficient when estimated with just a single sample. These new variants combine the benefits of their predecessors and address key weaknesses. In particular, they are easier to tune, offer an improved speed-accuracy trade-off, and do not rely on the use of debiasing estimators. We demonstrate their efficacy on a variety of probabilistic inference tasks (Section 6.4).

6.2 Background

In this section, we first introduce the problem setting. We then give an overview of EP, followed by a discussion of issues related to sampled estimation of EP updates.

Let $\mathcal{F}$ be the tractable, minimal exponential family of distributions (see Chapter 2), defined by the statistic function $s(\cdot)$ with respect to base measure $\nu(\cdot)$. Let Ω , $\mathcal{M}^\circ$, $A(\cdot)$ denote the natural domain, mean domain and log-partition function of $\mathcal{F}$, respectively, and $A^*(\cdot)$ the convex dual of $A(\cdot)$. Assume that a distribution of interest p^* , the *target distribution*, has a density, with respect to $\nu(\cdot)$, of the form

$$p^*(z) \propto p_0(z) \prod_{i=1}^m \exp(\ell_i(z)), \quad (6.1)$$

where p_0 is the member of $\mathcal{F}$ with natural parameter η_0 . In Bayesian inference settings, p^* would be the posterior distribution over parameters z given some observed data $\mathcal{D}$, in which case p_0 can be interpreted as a prior distribution, and $\ell_1(z), \dots, \ell_m(z)$ to log-likelihood terms given some partition of $\mathcal{D}$.¹ However, we consider the more general setting in which the density $p^*(.)$ is simply a product of factors. Note that we can assume form (6.1) without loss of generality by allowing p_0 to be improper. The inference problem typically amounts to computing quantities derived from the normalised density $p^*(.)$, such as samples, summary statistics, or expectations of given functions. In some special cases, these quantities can be computed exactly, but this is not feasible in general, and approximations must be employed.

6.2.1 Expectation propagation (EP)

Given a target distribution with density of the form (6.1), EP aims to find an approximation $q \in \mathcal{F}$ such that

$$q(z) \propto p_0(z) \prod_i \exp(\lambda_i^\top s(z)) \approx p^*(z). \quad (6.2)$$

By assumption $\mathcal{F}$ is tractable, and so provided that $(\eta_0 + \sum_i \lambda_i) \in \Omega$, q is a tractable member of $\mathcal{F}$. Each factor $\exp(\lambda_i^\top s(z))$ is known as a *site potential*, and can be roughly interpreted as a $\mathcal{F}$ -approximation to the i -th *target factor*, $\exp(\ell_i(z))$.² λ_i is known as the i -th *site parameter*.

Variational problem EP, and several of its variants (Hasenclever et al., 2017, Heskes and Zoeter, 2002, Minka, 2001), can be viewed as solving a variational problem – described in Section 4.4 – which we review briefly now, following the exposition of Hasenclever et al. (2017).

Let the i -th *locally extended family*, denoted $\mathcal{F}_i$, be the exponential family defined by the statistic function $s_i(z) = (s(z), \ell_i(z))$ with respect to base measure $\nu(.)$. Let $\Omega_i, \mathcal{M}_i^\circ$ and $A_i(.)$ denote the natural domain, mean domain, and log partition function of $\mathcal{F}_i$, respectively. Note that a member of $\mathcal{F}_i$ roughly corresponds to a distribution whose density is the (normalised) product of a member of $\mathcal{F}$ with the i -th target factor raised to a power. Unlike $\mathcal{F}$, we do not assume $\mathcal{F}_i$ is minimal.

¹We assume i and j to be taken from the index set $\{1, \dots, m\}$ unless otherwise specified.

²This interpretation is not precise, since it is not necessarily the case that $\lambda_i \in \Omega$.

Fixed points of EP correspond to the solutions of the saddle-point problem

$$\max_{\eta \in \Omega} \min_{[\lambda_i]_i} L(\eta, [\lambda_i]_i), \text{ where} \\ L(\eta, [\lambda_i]_i) = A\left(\eta_0 + \sum_i \lambda_i\right) + \sum_i \beta_i \left[A_i((\eta - \beta_i^{-1} \lambda_i, \beta_i^{-1})) - A(\eta)\right]. \quad (6.3)$$

At a solution to (6.3), the EP approximation is given by (6.2). The hyperparameters $[\beta_i]_i$ control the characteristics of the approximation, and correspond to the power parameters of power EP.

EP updates EP (Minka, 2001, Opper and Winther, 2000), power EP (Minka, 2004) and double-loop EP (Heskes and Zoeter, 2002), can all be viewed as alternating between some number (≥ 1) of *inner updates* to decrease L with respect to $[\lambda_i]_i$, with an *outer update* to increase L with respect to η (Hasenclever et al., 2017, Heskes and Zoeter, 2003). EP is not typically presented in this way, but by doing so we will be able to present a unified algorithm that succinctly illustrates the relationship between the different variants. We show equivalence with the conventional presentation of EP in Appendix D.1. The inner and outer updates, which we derived in Section 4.4.4, are given by

$$\textbf{Inner update: } \lambda_i \leftarrow \lambda_i - \alpha \left(\eta_0 + \sum_j \lambda_j - \nabla A^*(\mathbb{E}_{q_i(z)}[s(z)]) \right), \quad (6.4)$$

$$\textbf{Outer update: } \eta \leftarrow \eta_0 + \sum_j \lambda_j, \quad (6.5)$$

where $q_i \in \mathcal{F}_i$ denotes the i -th *tilted distribution*, with density

$$q_i(z) \propto \exp \left((\eta - \beta_i^{-1} \lambda_i)^\top s(z) + \beta_i^{-1} \ell_i(z) \right). \quad (6.6)$$

The hyperparameter α controls the level of damping, which is used to aid convergence. An undamped inner update (with $\alpha = 1$) can be seen as performing *moment matching* between q and q_i . The expectation in (6.4) is most often computed analytically, or estimated using deterministic numerical methods. It can also be estimated by sampling from q_i , however, we will later show that the resulting stochasticity can lead to a biased and unstable procedure. The inner updates can be performed either serially or in parallel (over i). In this chapter we assume they are applied in parallel, but the ideas presented can easily be extended to the serial case. Heskes and Zoeter (2003) showed that (6.4) follows a decrease direction in L with respect to λ_i , and so is guaranteed

to decrease L when α is small enough. When $[\lambda_i]_i$ are at a partial minimum of L (for fixed η), the outer update performs an exact partial maximisation of the primal form of the EP variational problem (4.47). All of the EP variants considered in this thesis differ only in their handling of the inner minimisation, and this is our focus.

Unified EP algorithm The double-loop EP algorithm of Heskes and Zoeter (2002) repeats the inner update to convergence before performing each outer update, which ensures convergence of the overall procedure. The usual presentation of EP combines (6.4) and (6.5) into a single update in λ_i (see Appendix D.1), however, Jylänki et al. (2011) observed that standard EP can also be viewed as performing double-loop EP with just a single inner update per outer update (which is not guaranteed to converge in general). By taking this view, we are able to present EP, power EP, and their double-loop counterparts as a single algorithm, presented in Algorithm 3. We do so primarily to illustrate how these variants are related to one another, and to the new variants of Section 6.3.

Algorithm 3 EP ($\beta_i=1, n_{\text{inner}}=1$), power EP ($\beta_i \neq 1, n_{\text{inner}}=1$), and their double-loop variants ($n_{\text{inner}} > 1$)

Require: $\mathcal{F}$, η_0 , $[\beta_i]_i$, $[\ell_i(z)]_i$, $[\lambda_i]_i$, n_{inner} , α

while not converged **do**

$\eta \leftarrow \eta_0 + \sum_j \lambda_j$

for 1 to n_{inner} **do**

for $i = 1$ to m *in parallel* **do**

$\lambda_i \leftarrow \lambda_i - \alpha(\eta_0 + \sum_j \lambda_j - \nabla A^*(\mathbb{E}_{q_i(z)}[s(z)]))$

return $[\lambda_i]_i$

Stochastic estimation of this expectation can lead to biased and unstable updates.

Note that dependence on $[\beta_i]_i$ comes through the definition of $q_i(\cdot)$. We can think of “exact” double-loop EP as corresponding to $n_{\text{inner}} = \infty$, with the inner loop exiting once some convergence criterion has been satisfied. In practice, a truncated inner loop is often used by setting n_{inner} to some small number (Hasenclever et al., 2017, Jylänki et al., 2011). Upon convergence of Algorithm 3, the approximation $q(z) \approx p^*(z)$ is given by (6.2). Going forwards we will refer to the family of algorithms encompassed by Algorithm 3 simply as EP.

Stochastic moment estimation EP is typically applied to models for which the updates either have closed-form expressions, or can be accurately estimated using deterministic numerical methods. However, as update (6.4) only depends on the target distribution through expectations under the tilted distributions, this suggests that

updates could also be performed using *sampled* estimates of those expectations. By using Monte Carlo (MC) methods to estimate the tilted distribution moments, EP can be used in a black-box manner, dramatically expanding the set of models it can be applied to. Instead of performing a single large sampling task—as would be required by applying MC methods directly—EP can instead solve several simpler ones (Hasenclever et al., 2017, Vehtari et al., 2020, Xu et al., 2014), which has significant computational benefits, as we discussed in Section 4.4.6. Unfortunately, when performed naively, update (6.4) is not robust to MC noise. This is because the moment estimates are converted to the natural (site) parameter space by mapping through $\nabla A^*(\cdot)$, which is not linear in general, and so MC noise in the estimates leads to biased updates of λ_i .

6.3 Fearlessly stochastic EP algorithms

In this section we show an equivalence between the moment-matching updates of EP and natural gradient descent (NGD). We use this view to motivate two new EP variants which have several advantages when updates are estimated using samples. We conclude with a review of related work.

6.3.1 Natural gradient view of EP

Several EP variants—EP (Opper and Winther, 2000, Minka, 2001), power EP, double loop EP (Minka, 2004), stochastic natural gradient EP (Hasenclever et al., 2017), and the new ones to follow—differ only in how they perform the *inner* optimisation in (6.3) with respect to $[\lambda_i]_i$. The optimisation can be viewed as one with respect to the $m + 1$ distributions $q, q_1, \dots, q_m$, which are jointly parameterised by $[\lambda_i]_i$ (Hasenclever et al., 2017). The natural gradient descent (NGD) method of Amari (1998), which we introduced in Chapter 3, involves preconditioning the gradient with the inverse Fisher information matrix (FIM) of a distribution. When it can be applied, it is often an effective method for optimising parameters of probability distributions. It would therefore seem desirable to apply NGD to the inner optimisation, yet doing so is not straightforward. While the FIM can be naturally extended to the product of statistical manifolds that is the solution space, computing and inverting it will not generally be tractable, making (the natural extension of) NGD in this space infeasible. Consider, instead, the distribution $\tilde{q}_i^{(t)} \in \mathcal{F}$, with natural parameter

$$\eta_i^{(t)}(\lambda_i) = \eta_0 + \sum_j \lambda_j^{(t)} + \alpha^{-1}(\lambda_i - \lambda_i^{(t)}). \quad (6.7)$$

We will later show that the update of EP is equivalent to an NGD step in L with respect to $\tilde{q}_i^{(t)}$. The (t) superscript on the site parameters above indicates that they are fixed for the current iteration, and so $\tilde{q}_i^{(t)}$ is fully parameterised by λ_i . Note that when $\lambda_j = \lambda_j^{(t)} \forall j$, we have $\tilde{q}_i^{(t)} = q$. To offer some intuition behind this distribution, note that when $\alpha = 1$ (the undamped case), we have

$$\eta_i^{(t)}(\lambda_i) = \eta_0 + \sum_{j \neq i} \lambda_j^{(t)} + \lambda_i, \quad (6.8)$$

and so $\tilde{q}_i^{(t)}$ can be viewed as the current approximation, but *with all site parameters except those for site i held fixed*.

Proposition 3, which we prove in Appendix D.2, states that the moment-matching updates of EP can be viewed as performing NGD in L with respect to *mean parameters* of $\tilde{q}_i^{(t)}$. We will again make use of the following property from Chapter 2 : for an exponential family with log partition function $A(\cdot)$, the gradient $\nabla A(\cdot)$ provides the map from natural to mean parameters, and this mapping is one-to-one for minimal families, with the inverse given by $\nabla A^*(\cdot)$.

Proposition 3. *For $\alpha > 0$, the moment-matching update of EP (6.4) is equivalent to performing an NGD step in L with respect to the mean parameters of $\tilde{q}_i^{(t)}$ with step size α^{-1} . That is, for $\mu_i = \mathbb{E}_{\tilde{q}_i^{(t)}(z)}[s(z)]$, and $\tilde{F}_i^{(t)}(\mu_i)$ the FIM of $\tilde{q}_i^{(t)}$ with respect to μ_i , we have*

$$\mu_i - \alpha^{-1} \left[\tilde{F}_i^{(t)}(\mu_i) \right]^{-1} \frac{\partial L}{\partial \mu_i} = \left(\nabla A \circ \eta_i^{(t)} \right) \left(\lambda_i - \alpha \left(\eta_0 + \sum_j \lambda_j - \nabla A^*(\mathbb{E}_{q_i(z)}[s(z)]) \right) \right). \quad (6.9)$$

Note that $\circ$ denotes function composition, so that, e.g., $(f \circ g)(x) = f(g(x))$. The right hand side of (6.9) just maps update (6.4) to mean parameters of $\tilde{q}_i^{(t)}$.³ We discussed in Section 6.2 that noise in the moment estimates results in bias in the update to λ_i of EP, due to the noise being passed through the nonlinear map $\nabla A^*(\cdot)$. This alone would not be a problem if the updates followed unbiased descent direction estimates in some other fixed parameterisation. Let $\eta_i = \nabla A^*(\mu_i)$ be natural parameters of $\tilde{q}_i^{(t)}$, then, from the definition of L and μ_i , we have

$$\frac{\partial L}{\partial \mu_i} = \frac{\partial \eta_i}{\partial \mu_i} \frac{\partial \lambda_i}{\partial \eta_i} \left(\nabla A \left(\eta_0 + \sum_j \lambda_j \right) - \mathbb{E}_{q_i(z)}[s(z)] \right). \quad (6.10)$$

³Note the apparent contradiction that decreasing α (increasing the damping) actually *increases* the NGD step size. This is resolved by observing that the definition of $\tilde{q}_i^{(t)}$ also changes with α .

This partial derivative appears on the left-hand side of (6.9), which can then be seen as being linear in $\mathbb{E}_{q_i(z)}[s(z)]$, and so when the expectation is estimated with unbiased samples, the NGD update in μ_i (6.9) is also unbiased. However, a *sequence* of updates introduces bias. This is because the parameterisation changes from one update to the next, with the mean parameters of $\tilde{q}_i^{(t)}$ mapped to those of $\tilde{q}_i^{(t+1)}$ by

$$\mu_i^{(t+1)} = \left(\nabla A \circ \eta_i^{(t+1)} \circ \left(\eta_i^{(t)} \right)^{-1} \circ \nabla A^* \right) \left(\mu_i^{(t)} \right), \quad (6.11)$$

where

$$\left(\eta_i^{(t)} \right)^{-1}(\eta) = \alpha \left(\eta - \eta_0 - \sum_j \lambda_j^{(t)} \right) + \lambda_i^{(t)} \quad (6.12)$$

is the inverse of $\eta_i^{(t)}(\cdot)$. $\eta_i^{(t+1)}(\cdot)$ is defined using the site parameters *after* the update at time t . That is, let $\lambda_j^{(t+1)}$ equal the value of λ_j after a parallel update at time t , then,

$$\eta_i^{(t+1)}(\lambda_i) = \eta_0 + \sum_j \lambda_j^{(t+1)} + \alpha^{-1}(\lambda_i - \lambda_i^{(t+1)}). \quad (6.13)$$

$\mu_i^{(t)}$ and $\mu_i^{(t+1)}$ denote mean parameters of $\tilde{q}_i^{(t)}$ and $\tilde{q}_i^{(t+1)}$, respectively. These are equivalent to μ_i in Proposition 3, but we use the superscript to make explicit the distribution that each parameterises.

The map between parameterisations defined by (6.11) is not linear in general, and so an unbiased, noisy update of μ_i at one time step, results in bias when mapped to the next. This bias can make the EP updates unstable in the presence of MC noise, leading previous work to use methods, specific to $\mathcal{F}$, for obtaining approximately unbiased *natural* parameter estimates from samples. However, even with these debiasing methods, a relatively large number of samples is still needed in practice (Vehtari et al., 2020, Xu et al., 2014). Here we have assumed parallel application of (6.4), but the bias at site i is induced whenever the parameters for other sites change from one update to the next, which occurs in both parallel and serial settings.

We now use the natural gradient interpretation of the EP updates to motivate two new variants that are far more robust to MC noise. In particular, they are stable, and most sample-efficient, when updates are estimated with just a single sample. Furthermore, unlike methods that rely on debiasing estimators, they do not require sample thinning (see Section 6.3.4). This largely eliminates two hyperparameters for the practitioner. While practical considerations, such as ensuring efficient use of parallel

hardware, may justify using more than one sample per update, this can in principle be tuned *a priori*, much like the batch size hyperparameter of stochastic gradient descent.

6.3.2 EP- η

We showed in the previous section that bias is introduced into the sequence of NGD updates due to a nonlinear map from one parameterisation to the next. If the map were affine, the sequence would remain unbiased. This can be achieved by performing NGD with respect to η_i , the *natural* parameters of $\tilde{q}_i^{(t)}$. Then, the map from one parameterisation to the next, by equating λ_i , is given by

$$\begin{aligned} \left(\eta_i^{(t+1)} \circ \left(\eta_i^{(t)}\right)^{-1}\right)(\eta_i) &= \eta_0 + \sum_j \lambda_j^{(t+1)} + \alpha^{-1} \left(\left(\eta_i^{(t)}\right)^{-1}(\eta_i) - \lambda_i^{(t+1)} \right) \\ &= \eta_i + \sum_j \left(\lambda_j^{(t+1)} - \lambda_j^{(t)} \right) + \alpha^{-1} \left(\lambda_i^{(t)} - \lambda_i^{(t+1)} \right), \end{aligned} \quad (6.14)$$

which is affine. The NGD direction in L with respect to $\tilde{q}_i^{(t)}$ and η_i is given by

$$-\left[\tilde{F}_i(\eta_i)\right]^{-1} \frac{\partial L}{\partial \eta_i} = -\alpha \frac{\partial \eta_i}{\partial \mu_i} \left(\nabla A\left(\eta_0 + \sum_j \lambda_j\right) - \mathbb{E}_{q_i(z)}[s(z)] \right). \quad (6.15)$$

See Appendix D.3 for a derivation. Note that α plays a similar role as the NGD step size in this parameterisation, but it also affects $\tilde{q}_i^{(t)}$, and so to obtain a more faithful NGD interpretation we fix $\alpha = 1$ and introduce an explicit NGD step size ϵ to decouple from the effect on $\tilde{q}_i^{(t)}$. The resulting update can be expressed directly as an update in λ_i , by applying $(\eta_i^{(t)})^{-1}$ to the updated η_i , giving

$$\lambda_i \leftarrow \lambda_i - \epsilon \frac{\partial \eta_i}{\partial \mu_i} \left(\nabla A\left(\eta_0 + \sum_j \lambda_j\right) - \mathbb{E}_{q_i(z)}[s(z)] \right). \quad (6.16)$$

Once we have an unbiased estimate of the bracketed expression in (6.16), we can compute (an unbiased estimate of) the second term efficiently. This is because the second term is simply a Jacobian-vector product (JVP) with respect to $\eta_i(\mu_i) = \nabla A^*(\mu_i)$. JVP computations can be efficiently performed by automatic differentiation packages – without explicitly instantiating the Jacobian matrix – with a computational cost that is comparable to that of computing the function itself ($\eta_i(\cdot)$, in this case).⁴

⁴This can actually be computed using forward *or* reverse mode automatic differentiation, because $\frac{\partial \eta_i}{\partial \mu_i}$ is symmetric, and so Jacobian-vector and vector-Jacobian products yield the same result.

In summary, by performing NGD with respect to *natural* parameters of $\tilde{q}_i^{(t)}$, instead of mean parameters, the resulting sequence of updates is unbiased in λ_i . We call the resulting procedure—which is given in Algorithm 4—EP- η , to emphasise that we have simply changed the NGD update of EP to use a *natural* parameterisation. The unbiased updates of EP- η allow it to be more sample-efficient than EP, by using a smaller number of samples per iteration. The left panel of Figure 6.1 demonstrates this effect in a stochastic version of the clutter problem (Minka, 2001).

6.3.3 EP- μ

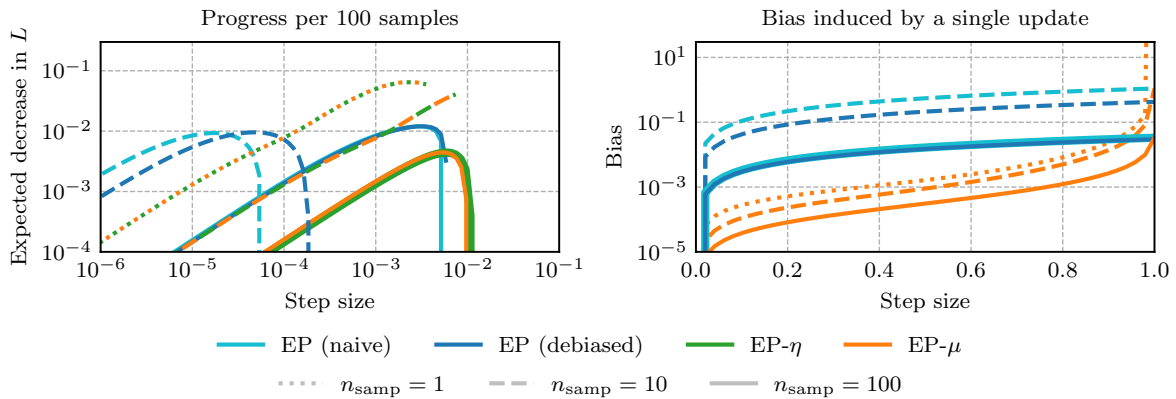


Fig. 6.1 The effect of varying the step size and number of MC samples (n_{samp}) on different EP variants in a stochastic version of the clutter problem of Minka (2001). EP (naive) uses maximum likelihood estimation to derive natural parameter updates from samples, and EP (debiased) uses the estimator of Xu et al. (2014). Step size corresponds to α for EP, and ϵ for EP- μ and EP- η . Only EP- η and EP- μ can perform 1-sample updates, hence the other traces are not visible. The left panel shows the expected decrease in L after $100/n_{\text{samp}}$ steps. Performing e.g. 100×1 -sample steps, or 10×10 -sample steps, achieves a much larger decrease in L than a single 100-sample step. The right panel shows the magnitude of the bias in λ_i after a single parallel update, averaged over all sites and dimensions. The bias of EP- μ shrinks far faster as the step size decreases than that of EP. EP- η is always unbiased and so is not visible.

Proposition 4 below shows that the bias introduced by the updates of EP cannot be mitigated by reducing α (increasing the damping) without also proportionally sacrificing the amount of progress made by each update. More concretely, let the bias in dimension d of $\mu_i^{(t+1)}$, after an update at time t , be defined as $\mathbb{E}[\mu_i^{(t+1)} - \bar{\mu}_i^{(t+1)}]_d$, where $\bar{\mu}_i^{(t+1)}$ is the value of $\mu_i^{(t+1)}$ after a *noise-free* update, and expectation is taken over the sampling distributions of all parallel updates at time t . Then, Proposition 4 below, which we prove in Appendix D.5, summarises the effect of decreasing α on EP.

Proposition 4. *After update (6.4) is executed in parallel over i , as $\alpha \rightarrow 0^+$, both the expected decrease in L , and the bias $\mathbb{E}[\mu_i^{(t+1)} - \bar{\mu}_i^{(t+1)}]_d$, are $O(\alpha)$ for all d .*

Proposition 3 at the beginning of this section states that update (6.4) can be viewed as performing an NGD step with respect to μ_i with a step size of α^{-1} . However, changing α also has the effect of changing $\tilde{q}_i^{(t)}$. It is then natural to wonder what happens if we fix $\alpha = 1$ and introduce an explicit step size for NGD, ϵ , resulting in the update

$$\mu_i \leftarrow \mu_i - \epsilon \left(\nabla A \left(\eta_0 + \sum_j \lambda_j \right) - \mathbb{E}_{q_i(z)}[s(z)] \right). \quad (6.17)$$

See Appendix D.4 for a derivation. It turns out that in doing so, when we decrease ϵ , the bias shrinks far faster than the expected decrease in L .

Proposition 5. *After update (6.17) is executed in parallel over i , as $\epsilon \rightarrow 0^+$, the expected decrease in L is $O(\epsilon)$, and the bias $\mathbb{E}[\mu_i^{(t+1)} - \bar{\mu}_i^{(t+1)}]_d$ is $O(\epsilon^2)$, for all d .*

Proposition 5, which we prove in Appendix D.5, tells us that by using update (6.17), we can reduce the bias to arbitrarily small levels while still making progress in decreasing L . Update (6.17) can also be expressed directly as an update in λ_i by applying $(\eta^{(t)})^{-1} \circ \nabla A^*$, giving

$$\lambda_i \leftarrow \nabla A^* \left((1 - \epsilon) \nabla A \left(\eta_0 + \sum_j \lambda_j \right) + \epsilon \mathbb{E}_{q_i(z)}[s(z)] \right) - \eta_0 - \sum_{j \neq i} \lambda_j, \quad (6.18)$$

which has the simple interpretation of performing EP, but with damping of the moments instead of the site (natural) parameters. In summary, if we perform NGD with respect to the same mean parameterisation as EP, but treat the NGD step size as a free parameter, ϵ , we can obtain updates that are *approximately* unbiased (by reducing the step size, ϵ , as shown by Proposition 5), while still making progress. We call this variant EP- μ , again to indicate that we are simply performing the NGD update of EP in the mean parameter space.⁵ The resulting procedure is also given by Algorithm 4. In Table 6.1 we show the updates for EP- η , EP- μ , and several other related algorithms.

The computational cost of EP- μ is lower than that of EP- η because it does not require JVPs through $\nabla A^*(.)$.⁶ The drawback is that the updates of EP- μ do still retain some bias, however, we find that the bias of EP- μ typically has negligible impact on its performance relative to EP- η . This is evident in the clutter problem of Minka

⁵EP too is performing NGD in mean parameters, but using a “step size” that also affects $\tilde{q}_i^{(t)}$.

⁶See Appendix D.7 for a detailed discussion of the computational costs of EP- η and EP- μ .

Algorithm 4 EP- η and EP- μ (differences with Algorithm 3 are highlighted in green)

Require: $\mathcal{F}$, η_0 , $[\beta_i]_i$, $[\ell_i(z)]_i$, $[\lambda_i]_i$, n_{inner} , ϵ

while not converged **do**

$\theta \leftarrow \eta_0 + \sum_j \lambda_j$

for 1 to n_{inner} **do**

for $i = 1$ to m *in parallel* **do**

update λ_i using (6.16) for EP- η , or (6.18) for EP- μ

return $[\lambda_i]_i$

(2001), as demonstrated in the left panel of Figure 6.1, as well as in the larger scale experiments of Section 6.4. The right panel of Figure 6.1 illustrates how quickly the bias in λ_i shrinks as the step size of EP- μ is decreased. Note that the results of this subsection are stated in terms of bias in μ_i , but it is straightforward to show that equivalent results also hold for λ_i using Taylor series arguments.

Algorithm	Update
EP	$\lambda_i \leftarrow \lambda_i - \alpha \left(\eta_0 + \sum_j \lambda_j - \nabla A^*(\mathbb{E}_{q_i(z)}[s(z)]) \right)$
EP- η	$\lambda_i \leftarrow \lambda_i - \epsilon \frac{\partial \eta_i}{\partial \mu_i} \left(\nabla A \left(\eta_0 + \sum_j \lambda_j \right) - \mathbb{E}_{q_i(z)}[s(z)] \right)$
EP- μ	$\mu_i^{(t)} \leftarrow \mu_i^{(t)} - \epsilon \left(\nabla A \left(\eta_0 + \sum_j \lambda_j \right) - \mathbb{E}_{q_i(z)}[s(z)] \right)$
SNEP	$\gamma_i \leftarrow \gamma_i - \epsilon \left(\nabla A \left(\eta_0 + \sum_j \lambda_j \right) - \mathbb{E}_{q_i(z)}[s(z)] \right)$
CVI	$\lambda \leftarrow \lambda - \epsilon \left(\lambda - \nabla_\mu \left(\mathbb{E}_{q(z)}[\ell(z)] \right) \right)$
PVI+NG	$\lambda_i \leftarrow \lambda_i - \epsilon \left(\lambda_i - \nabla_\mu \left(\mathbb{E}_{q(z)}[\ell_i(z)] \right) \right)$

Table 6.1 The update performed by various algorithms. EP- μ and SNEP follow the same update direction, but their state variables map to (and from) λ_i differently. The (t) superscript in the EP- μ update is to emphasise that the parameterisation is time dependent. Note that ϵ denotes an NGD step size, and EP uses α rather than ϵ , since α does not have a strict NGD step size interpretation, as discussed in the main text. CVI can be interpreted as a special case of PVI+NG with $m = 1$. PVI+NG (and therefore CVI) is also the limiting case of EP as $\alpha \rightarrow \infty$ and $\beta_i = \alpha/\epsilon \forall i$.

6.3.4 Related work

The link between natural gradients and moment matching in an exponential family is well known (Bouchard-Côté et al., 2024), but the connection with EP shown here – which relies on identification of the distribution $\tilde{q}_i^{(t)}$ and parameterisation $\eta_i^{(t)}$ – is new, to the best of our knowledge. Bui et al. (2018) showed that the updates of (power) EP

are equivalent to performing NGD in a local variational free energy objective when taking the limit of the power parameter as $\beta_i \rightarrow \infty$, and Wilkinson et al. (2023) showed that this extends to the (global) variational free energy objective for models with a certain structure. Our result is different in that it relates to NGD of the EP variational objective, and applies for all values of β_i .

EP was combined with Markov chain Monte Carlo (MCMC) moment estimates by Xu et al. (2014) in a method called *sampling via moment sharing* (SMS), and later by Vehtari et al. (2020) in the context of hierarchical models. In both works, when $\mathcal{F}$ was multivariate normal (MVN) —arguably the most useful case—the authors used an estimator for the updates that is unbiased when $\tilde{q}_i$ is also MVN. Although this estimator is only *approximately* unbiased in general, it can help to stabilise the updates significantly. Even so, it is necessary to use a relatively large number of samples for the updates, with several hundred or more being typical. Using many samples per update is inefficient, as the update direction can change from one iteration to the next. Furthermore, such estimators typically rely on a known number of *independent* samples. Samples drawn using MCMC methods are generally autocorrelated, necessitating the use of sample thinning before the estimators can be applied. This adds another hyperparameter to the procedure—the thinning ratio—and it is also inefficient due to the discarding of samples. The practitioner is required to choose the number of samples, the thinning ratio, and the amount of damping, all of which affect the accuracy, stability, and computational efficiency of the procedure. There is no easy way to choose these *a priori*, forcing the practitioner to either set them conservatively (favouring accuracy and stability), or to find appropriate settings by trial and error, both of which are likely to expend time and computation unnecessarily. We demonstrate the relative difficulty of tuning EP hyperparameters in Section 6.5. The black-box alpha method of Hernandez-Lobato et al. (2016) solves a variational problem that is derived from that of EP, but with the additional constraint that the site parameters are tied (so that $\lambda_i = \lambda_j \forall i, j$). In black-box alpha, the intractable integrals are estimated using simple Monte Carlo, and the resulting gradient estimates are biased due to the estimated quantities appearing non-linearly in the objective.

The stochastic natural gradient EP (SNEP) method of Hasenclever et al. (2017), which is closely related to our work, also optimises the EP variational objective, using a form of NGD. The SNEP updates are unbiased in the presence of MC noise, allowing them to be performed with as little as one sample, without relying on $\mathcal{F}$ -specific debiasing estimators, or the sample thinning that would entail. In SNEP, NGD is performed with respect to mean parameters of the *site potentials*, which are treated as

bona fide distributions in $\mathcal{F}$. In contrast, we showed that EP can already be viewed as performing NGD, but with respect to the distributions $\tilde{q}_1^{(t)}, \dots, \tilde{q}_m^{(t)}$, and our new variants, EP- η and EP- μ , are able to gain the same advantages as SNEP, but using the same distributions for NGD as EP. Hasenclever et al. (2017) showed that in some settings, SNEP can obtain accurate point estimates fairly rapidly. However, we find that it typically converges far slower than both EP and our new variants, consistent with findings in Vehtari et al. (2020). We argue that this is because the site potentials (when considered as distributions) bear little resemblance with the distributions that are ultimately being optimised, and so their geometry is largely irrelevant for the optimisation of L . In contrast, the geometry of $q_i^{(t)}$ is closely related to that of L , as we show in Appendix D.6.

We note that Xu et al. (2014) and Hasenclever et al. (2017) also proposed methods for performing updates in an asynchronous fashion, significantly reducing the frequency and cost of communication between nodes in distributed settings. In principle, these methods could be combined with those presented in this chapter, but we do not consider them further here.

6.4 Evaluation

In this section we compare the efficacy of EP- η and EP- μ with other EP variants on a variety of probabilistic inference tasks. In each experiment, the task was to perform approximate Bayesian inference of unobserved parameters z , given some observed data $\mathcal{D}$. All of the models in these experiments followed the same general structure, consisting of a minimal exponential family prior over z , p_0 , and a partition of the data $[\mathcal{D}_i]_i^m$, where each block $\mathcal{D}_i$ depends on both z and an additional vector of *local* latent variables w_i that also depend on z . That is, the joint density has the form

$$p(z) \propto p_0(z) \prod_i p(w_i | z) p(\mathcal{D}_i | w_i, z), \quad (6.19)$$

where $p_0 \in \mathcal{F}$ has natural parameters $\eta_0 \in \Omega$. Inference in models with this kind of hierarchical structure can often be challenging (Gelman et al., 2013), but EP is particularly well suited to them, for reasons we discussed in Section 4.4.6. This structure is shown graphically in Figure 6.2.

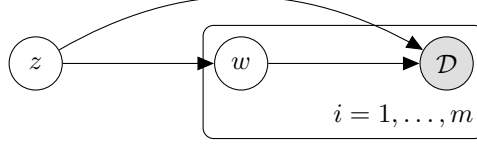


Fig. 6.2 Directed graphical model for the experiments of Section 6.4.

To perform approximate inference of z , we first define $\ell_i(z)$ as the log likelihood of $\mathcal{D}_i$ given z , with w_i integrated out. That is,

$$\ell_i(z) = \log \int p(\mathcal{D}_i, w_i | z) dw_i. \quad (6.20)$$

Then, given η_0 and $[\ell_i(z)]_i$, we define $p^*(z)$ as (6.1), and proceed to find an approximation $q(z) \approx p^*(z)$ using the methods described in earlier sections. Note that sampling from the tilted distribution $q_i(z)$ requires jointly sampling over z and w_i and taking the marginal. EP is a particularly appealing framework for performing inference in this setting, as the dimensionality of the sampled distributions is constant with respect to m , mitigating the curse of dimensionality that would be suffered by a direct application of MCMC (Vehtari et al., 2020). In our experiments we used NUTS (Hoffman and Gelman, 2014) to perform the sampling, consistent with prior work (Vehtari et al., 2020, Xu et al., 2014). In each experiment we compared EP- η and EP- μ with EP, in the manner used by Xu et al. (2014) and Vehtari et al. (2020). When $\mathcal{F}$ was MVN we also compared to SNEP (Hasenclever et al., 2017). For the models with normal-inverse Wishart (NIW) $\mathcal{F}$, we were unable to find an initialisation for SNEP that would allow us to perform a meaningful comparison. We discuss this point further in Appendix D.8.

To evaluate the performance of the different variants, we monitored KL divergence to an *estimate* of the optimum, obtained by running EP with a large number of samples. We used 500 different hyperparameter settings for each variant, chosen using random search, and repeated each run 5 times using different random seeds for NUTS. All runs of EP- η and EP- μ were performed with $n_{\text{samp}} = 1$ and $n_{\text{inner}} = 1$. The results of our evaluation are summarised by the Pareto frontiers in Figure 6.3.

We see that EP- η and EP- μ can typically reach a given level of accuracy (distance from an estimate of the optimum) faster than EP, often significantly so. We stress that the frontiers for EP- η and EP- μ are traced out with n_{samp} and n_{inner} each fixed to 1, with no sample thinning, varying only the step size hyperparameter ϵ , with higher accuracy/cost regions of the frontier corresponding to lower values of ϵ . In contrast, to trace out the frontier of EP we must jointly vary α , n_{samp} , and the thinning ratio. See

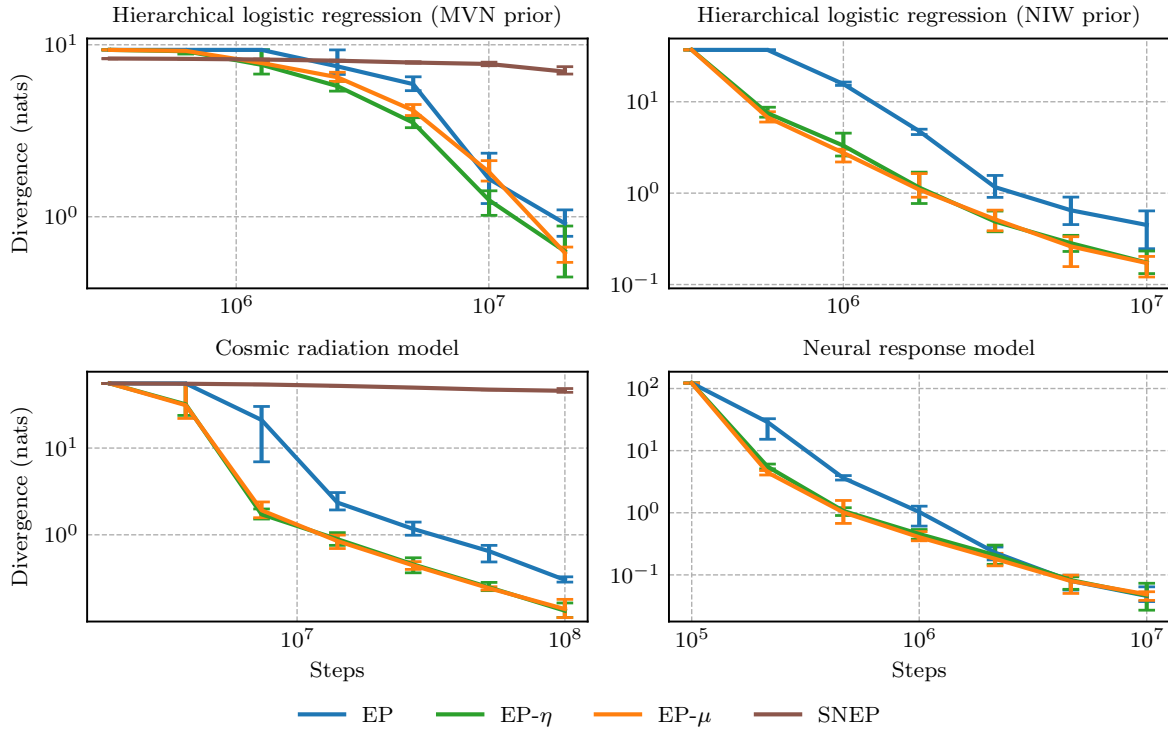


Fig. 6.3 Pareto frontiers showing the number of NUTS steps (x -axis) against the KL divergence from q to an estimate of the optimum (y -axis). Each point on the plot marks the lowest average KL divergence attained by *any* hyperparameter setting by that step count. Error bars mark the full range of values for the marked hyperparameter setting across 5 random seeds.

Section 6.5 for examples of this effect. The performances of EP- η and EP- μ are very similar. We found SNEP to be significantly slower than the other methods, consistent with findings by Vehtari et al. (2020).

We have largely assumed that the drawing of samples is the dominant cost. In practice, other costs become relevant, which may shift the balance in favour of using more samples to estimate updates. We discuss this in Appendix D.7, along with strategies for reducing computational overheads. We note, however, that wall-clock time results for our experiments (in Appendix D.9) are in line with the results of this section, indicating that the drawing of samples was indeed the dominant cost.

Our experiment code can be found at <https://github.com/cambridge-mlg/fearless-ep>. We now provide a brief overview of individual experiments, with further details given in Appendix D.8.

Hierarchical logistic regression with MVN prior Hierarchical logistic regression (HLR) is used to perform binary classification in a number of groups when the extent

to which data should be pooled across groups is unknown (Gelman et al., 2013). We performed approximate Bayesian inference in a HLR model using a MVN prior over the group-level parameters. We applied this model to synthetic data, generated using the same procedure as Vehtari et al. (2020), which was designed to be challenging for EP. In this experiment $z \in \mathbb{R}^8$, $w_i \in \mathbb{R}^4 \forall i$, and each of the $m = 16$ groups had $n = 20$ observations. In the upper-left panel of Figure 6.3 we see that EP- η and EP- μ typically reach a given level of accuracy faster than EP and SNEP.

Hierarchical logistic regression with NIW prior We also performed approximate Bayesian inference in a similar HLR model, but using a normal-inverse Wishart (NIW) prior over group-level parameters, allowing for correlation of regression coefficients within groups. We applied this model to political survey data, where each of $m = 50$ groups corresponded to responses from a particular US state, with 7 predictor variables corresponding to characteristics of a given survey respondent, so that $w_i \in \mathbb{R}^7 \forall i$. There were $n = 97$ survey respondents per state. The data, taken from the 2018 Cooperative Congressional Election Study, related to support for allowing employers to decline coverage of abortions in insurance plans (Lopez-Martin et al., 2022, Schaffner et al., 2019). The upper-right panel of Figure 6.3 shows that EP- η and EP- μ consistently reach a given level of accuracy significantly faster than EP.

Cosmic radiation model A hierarchical Bayesian model was used by Vehtari et al. (2020) to capture the nonlinear relationship between diffuse galactic far ultraviolet radiation and 100- μm infrared emission in various sectors of the observable universe, using data obtained from the Galaxy Evolution Explorer telescope. In this model each $w_i \in \mathbb{R}^9$ parameterised a nonlinear regression model using data obtained from one of m sections of the observable universe. The m regression problems were related through hyperparameters $z \in \mathbb{R}^{18}$, which parameterised the section-level parameter densities. The prior, $p_0 \in \mathcal{F}$, was MVN. The specifics of the nonlinear regression model are quite involved and we refer the reader to Vehtari et al. (2020) for details. We were unable to obtain the dataset, and so we generated synthetic data using parameters that were tuned by hand to try and match the qualitative properties of the original data – see Appendix D.11 for examples. We used a reduced number of $m = 36$ sites and $n = 200$ observations per site to allow us to perform a comprehensive hyperparameter search. The lower-left panel of Figure 6.3 shows once again that EP- η and EP- μ consistently reach a given level of accuracy significantly faster than EP and SNEP.

Neural response model A common task in neuroscience is to model the firing rates of neurons under various conditions. We performed inference in a hierarchical Bayesian neural response model, using recordings of V1 complex cells in an anaesthetised adult cat (Blanche, 2009). In this dataset, 10 neurons in a specific area of cat V1 were simultaneously recorded under the presentation of 18 different visual stimuli, each repeated 8 times, for a total of 144 trials. We modelled the observed spike counts of the 10 neurons in each trial as Poisson, with latent (log) intensities being MVN, with mean and covariance (collectively $z \in \mathbb{R}^{65}$) drawn from a NIW prior $p_0 \in \mathcal{F}$. Inference of z amounted to inferring the means, and inter-neuron covariances, of latent log firing rates across stimuli. We grouped the data into $m = 8$ batches of $n = 18$ trials each, so that the latent log firing rates for all trials in batch i were jointly captured by $w_i \in \mathbb{R}^{180}$. The lower-right panel of Figure 6.3 shows that EP- η and EP- μ either match or beat the speed of EP for reaching any given level of accuracy. For high levels of accuracy the performances of the methods appear to converge with one another, but note that the asymptote is not exactly zero because the optimal solution was estimated by running EP with a large (but finite) number of samples.

6.5 Hyperparameter sensitivity analysis

In order to assess hyperparameter sensitivity, we examined the effect of varying the hyperparameters of EP and EP- η in the synthetic hierarchical logistic regression experiment described in Section 6.4. The results for EP- μ are similar to those of EP- η and so we do not present them separately here.

EP In the left panel of Figure 6.4 we show the effect of varying the number of samples used for estimating updates in EP – more accurate regions of the frontier generally require more samples. The middle panel, similarly, shows the effect of varying the step size, and the right panel shows the effect of varying the thinning ratio τ . Together these plots illustrate the difficulty of tuning EP in stochastic settings – tracing out the frontier is a three-dimensional problem. For example, to achieve the best accuracy within a compute budget of 10^7 steps, the practitioner would need to set $400 < n_{\text{samp}} \leq 500$, $0.1 < \alpha \leq 0.3$, and $\tau = 2$, and any deviation from this would seemingly result in suboptimal accuracy.

EP- η In the left panel of Figure 6.5 we show the effect of varying the number of samples used for estimating updates in EP- η – the frontier is traced out with $n_{\text{samp}} = 1$.

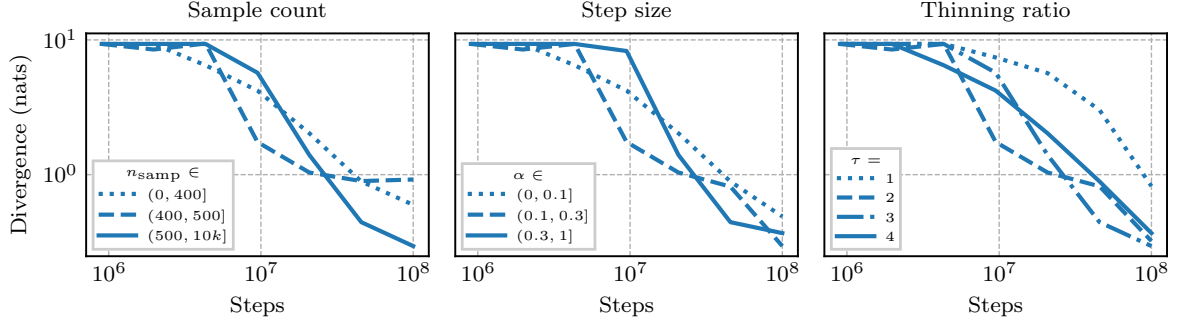


Fig. 6.4 The effect of varying EP hyperparameters. Partial Pareto frontiers show the number of NUTS steps (x -axis) against the KL divergence from q to an estimate of the optimum (y -axis).

We note however that the difference between $n_{\text{samp}} = 1$ and $n_{\text{samp}} = 10$ is relatively small, and so it may be sensible to choose $n_{\text{samp}} > 1$ to make efficient use of parallel hardware, or to minimise per-iteration overheads (see Appendix D.7). The middle panel, similarly, shows the effect of varying the step size, with more accurate regions of the frontier corresponding to a smaller step size. This also suggests that in practice it may make sense to set ϵ relatively large at first and then gradually decay it to improve the accuracy. Finally, the right panel shows the effect of varying n_{inner} , the number of inner steps performed per outer update. $n_{\text{inner}} > 1$ corresponds to using EP- η in “double-loop” mode. We did not find it necessary to increase n_{inner} above one to obtain convergence in our experiments, but Figure 6.5 (right) demonstrates that doing so would have a relatively small impact on performance. Together these plots illustrate that hyperparameter tuning for EP- η is relatively straightforward. The frontier can be largely be traced out by varying ϵ , and is relatively insensitive to n_{samp} and n_{inner} .

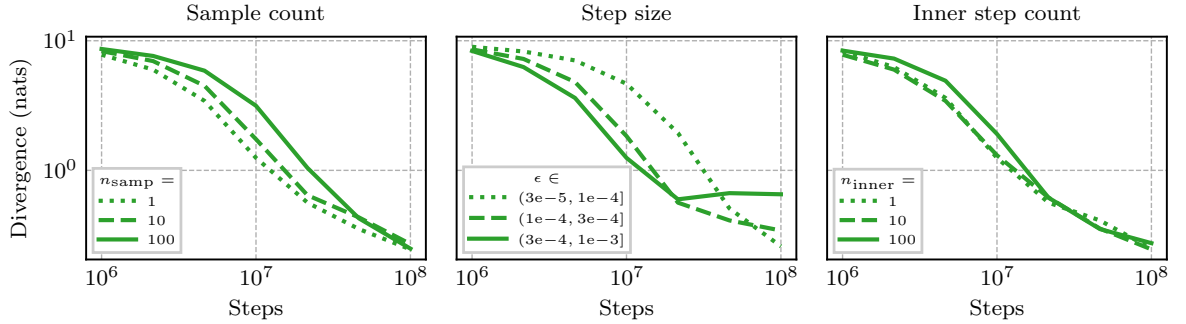


Fig. 6.5 The effect of varying EP- η hyperparameters. Partial Pareto frontiers show the number of NUTS steps (x -axis) against the KL divergence from q to an estimate of the optimum (y -axis).

6.6 Comparison with CVI

Our focus in this chapter has been on developing improved methods for performing EP when the required moments must be estimated using samples. This potentially enables the use of EP as a black-box inference method, with the motivation for this kind of setting having been discussed by prior work (Xu et al., 2014, Hasenclever et al., 2017, Vehtari et al., 2020). EP and its variants are agnostic to the choice of sampling kernel, and as such we have not yet discussed issues relating to the performance of the underlying samplers, instead we have so far only focused on the relative performance of different EP variants.

In order to understand the trade-offs involved in using EP variants over variational inference (VI) – which too can be used in black-box settings, and has similar aims – we performed experiments to explore the relative strengths and weaknesses of EP- η and conjugate-computation variational inference (CVI) (Khan and Lin, 2017), an efficient natural gradient VI method that is closely related to EP, as discussed in Section 4.5.

Experimental setup We used EP- η and CVI to perform approximate Bayesian inference using the same hierarchical logistic regression model (with MVN prior) and synthetic dataset as Section 6.4. For our evaluation metrics, we used both forward and reverse KL divergences between the current (MVN) approximation, and a MVN “target” distribution. The target distribution was a MVN distribution fitted using 500,000 samples from the posterior, obtained by running NUTS for 1 million iterations and then discarding the first half. For CVI, we used a structured MVN approximation that had the same conditional independence structure as the true posterior – that is, it modelled all pairwise dependencies within z , within each w_i , and between z and each w_i , but did not model direct dependence between w_i, w_j for $i \neq j$. We used 500 random hyperparameter settings for each method, using the search spaces described below. Unless stated otherwise (below), all other details were as described in Appendix D.8.

Hyperparameter search spaces For EP- η we used the same hyperparameter search space as described in Appendix D.8. For CVI, the step size was drawn log-uniformly in the range $(10^{-5}, 1)$, and the number of samples used to estimate the update was drawn log-uniformly in the range $[.5, 100.5)$ and then rounded to the nearest integer. The site parameters for CVI were initialised to be the parameters of a zero-mean MVN distribution with scaled identity covariance matrix, with the scale drawn log-uniformly from $(10^{-5}, 10^5)$.

Wall-clock time performance comparison We compared the wall-clock time performance of EP- η and CVI, with the former using NUTS as the underlying sampler. For both methods, we chose the hyperparameters that gave the lowest *reverse* KL divergence between the approximation and the MCMC target distribution after 100 seconds. The results of this experiment are shown in the left panel of Figure 6.6. We can see that EP- η converges at least one order of magnitude slower than CVI on this problem, when measured in wall-clock time. This difference is largely due to the relative cost of drawing samples for the two methods. While CVI also uses samples to estimate its updates, it uses samples from the current MVN approximation, which are cheap to generate using standard methods. EP- η on the other hand (as with other EP variants), requires samples from the *tilted* distributions to estimate the updates, and in this experiment we used NUTS to draw those samples. Each NUTS sample can involve many evaluations of the energy function gradient, and furthermore, because MCMC samples are generally autocorrelated, the cost of drawing approximately *independent* samples can be even higher. We used NUTS to be consistent with prior work, and because our focus in this work has primarily been the relative performance of different EP variants, we did not make particular efforts to improve the performance of the underlying samplers, beyond tuning their hyperparameters in a fairly standard manner. However, all of the EP variants considered in this chapter, including EP- η , are agnostic to the choice of sampling kernel used, and we believe there is scope to significantly improve the efficiency of the underlying samplers, for reasons we briefly discuss in Section 6.7.

Sample efficiency In order to decouple the performance of EP- η from that of the underlying sampler, we repeated the experiment using an “oracle” sampling kernel for EP- η . This oracle kernel was simply NUTS but with a thinning ratio of 100, so that each sample was an approximately independent sample from the tilted distribution. We then compared EP- η and CVI using the same metrics as the previous experiment, but with “time” measured in samples. We chose the hyperparameter setting for each method that achieved the lowest reverse KL divergence after 1,500 samples. When measured on this basis we see that the convergence speeds of CVI and EP- η are roughly equivalent, demonstrating that the sample efficiency of the two methods are similar. This should not be too surprising, as CVI is (mathematically, but not algorithmically) equivalent to the limiting case of power EP (as $\beta_i \rightarrow \infty$) (Bui et al., 2018, Wilkinson et al., 2023).

Approximation quality Figure 6.6 demonstrates that EP- η is able to achieve a more faithful approximation of the target distribution when measured by either the forward *or* reverse KL. At first it may seem surprising that EP is able to obtain a lower *reverse* KL divergence than CVI, but there are two reasons for this apparent anomaly. First, with EP, we have a MVN approximation for z only, with the approximate posterior over local variables (w_i for $i = 1, \dots, m$) represented only implicitly with samples. CVI on the other hand necessarily optimises a joint MVN over *all* variables, and the z marginal of the optimal VI posterior over all variables is not the same as the optimal VI posterior over z . The second reason is that our target distribution is essentially a moment-matched MVN approximation of the true posterior, which would naturally tend to favour the forward KL divergence that is (approximately) targeted by EP. Nevertheless, a qualitative assessment of the pairwise marginals, as seen in the two rightmost panels of Figure 6.6, shows that the *true* posterior is more faithfully approximated by EP- η than CVI, with the latter significantly underestimating uncertainty in the inter-group variability parameters, $[\log \sigma_i]_i$. The full set of pairwise posterior marginals can be seen in Figure D.2. Note that the CVI approximation appears more constrained than might be expected based on the marginal plots alone. This is due to the zero-avoiding nature of the reverse KL divergence – non-Gaussianity in one dimension can constrain the marginal distributions of others, particularly when there is strong dependence. The local variables, $[w_i]$, can have both significant non-Gaussian posterior structure, and strong dependence on $[\log \sigma_i]_i$.

We see from the results of these experiments that EP- η is able to obtain much more faithful posterior approximations than CVI, but it is significantly slower when measured in wall-clock time, despite the gains made over its predecessors. This is due to the relative cost of drawing independent samples with NUTS, as the sample efficiency of EP- η is roughly equivalent to that of CVI when an (illustrative) “oracle” sampling kernel is used. EP and its variants are agnostic to the choice of sampling kernel however. Although we have not done so here, we believe there are significant gains to be had from exploiting knowledge of the tilted distributions in order to accelerate sampling performance, which may allow us to close the computational performance gap between EP and CVI. We discuss some examples of how this could be achieved in Section 6.7 below.

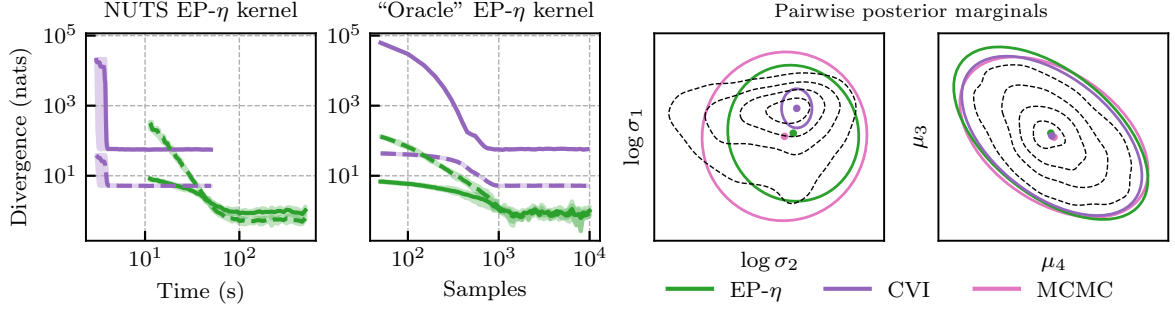


Fig. 6.6 Comparison of EP- η with conjugate-computation variational inference (CVI) on a hierarchical logistic regression model. The two leftmost plots show forward (solid) and reverse (dashed) KL divergences between the approximation of each method and a MVN distribution estimated directly from MCMC samples. The left panel shows a comparison with respect to wall-clock time, when NUTS is used as the underlying sampling kernel for EP- η . The left-middle panel shows a similar comparison, but with respect to the number of samples drawn, and using an “oracle” sampling kernel for EP- η . Hyperparameters were tuned for each method and shaded regions show the range of trajectories across 5 random seeds. The right and right-middle panels show pairwise marginals of the various MVN approximations, overlaid on contours of the true posterior. Coloured dots and ellipses correspond to means and 2-standard-deviation contours, respectively. See Section 6.6 for further details.

6.7 Future work

In order for EP variants to be widely adopted as a framework for performing black-box inference, we believe there are some challenges that still need to be addressed. We showed in Section 6.6 that the performance of EP variants is constrained by the performance of the underlying samplers. Furthermore, the samplers typically have hyperparameters of their own, which are often adapted using so-called warmup phases. While EP- η and EP- μ significantly reduce the complexity of tuning hyperparameters specific to EP, they do not help with those of the underlying samplers. This means that the complexity of hyperparameter tuning (for all EP variants) is still greater than for direct MCMC methods.

We believe that both of these issues can be addressed by exploiting the close relationship between the approximation $q \in \mathcal{F}$ and the tilted distributions. Note that at any given time, q is our best $\mathcal{F}$ approximation of each tilted distribution $q_i \forall i$, and in fact, after an “outer” update, they differ by only one of $m + 1$ factors. This close similarity could be exploited in several ways.

Automatic setting of hyperparameters Knowledge of q can be used to set MCMC hyperparameters directly. For example, in Hamiltonian Monte Carlo (HMC) sampling schemes, it is common to use a mass matrix to improve the conditioning of the energy landscape. This mass matrix is typically derived from the estimated covariance matrix of the sampled distribution (or its diagonal entries), which is estimated using samples obtained during a warm-up phase. For many choices of $\mathcal{F}$, the covariance of q will be available in closed form, and so this can be used to derive the mass matrix directly, avoiding the need to estimate the covariance matrix during warm-up. Furthermore, when $\mathcal{F}$ is Gaussian, the tilted distributions will also be “close to” Gaussian, and so in using the covariance of q to derive the mass matrix, we will obtain an energy landscape that is close to *isotropic* Gaussian. Not only is this the ideal landscape for HMC, but it also allows us to leverage known properties in this setting, such as optimal Metropolis-Hastings acceptance probabilities (Neal, 1996) and maximal stable step sizes (Leimkuhler and Reich, 2005).

Reinitialisation of sampling chains MCMC chains could be reinitialised using samples from q for each update (or sample). This would provide approximate independence between updates (or samples) for very little extra cost, assuming sampling from q is cheap. More generally, we could consider a sequence of distributions from q to q_i , and perform annealed importance sampling along that sequence.

Metropolis-Hastings transitions Transitions of the underlying MCMC kernel could be interleaved with trivial Metropolis-Hastings (MH) transitions using q as a proposal density. When q and q_i are very similar, transitions will be accepted with high probability, reducing autocorrelations in the MCMC chain toward 0. We could of course use these simple MH transitions alone for sampling, but by interleaving with a more advanced sampler like HMC, we have an automatic fallback if the acceptance rate of the trivial MH transitions is too low.

Control variates In Table 6.1 we can see that the updates of EP- η , EP- μ , and SNEP, rely on the moment difference

$$\nabla A \left(\eta_0 + \sum_j \lambda_j \right) - \mathbb{E}_{q_i(z)}[s(z)] = \mathbb{E}_{q(z)}[s(z)] - \mathbb{E}_{q_i(z)}[s(z)]. \quad (6.21)$$

The first term in this difference is available in closed form ($\mathcal{F}$ is tractable by assumption), but we have assumed that the second term is only available up to sampled estimates in

general. In our evaluation we computed the first term in closed form. However, because (6.21) is a difference in expectations between two similar distributions, the variance of updates could be significantly reduced by using a control variate – a stochastic estimate of $\mathbb{E}_{q(z)}[s(z)]$ that is correlated with that of $\mathbb{E}_{q_i(z)}[s(z)]$. For example, if each underlying MCMC transition is followed by a simple MH transition, as described above, the same sample from q could be used as both a transition proposal *and* a sample for estimating $\mathbb{E}_{q(z)}[s(z)]$. In this case, when q and q_i are very similar the MH acceptance probability would approach 1, and the variance in the update would vanish.

In our evaluation we used NUTS as the underlying sampler for the different EP variants, consistent with prior work (Xu et al., 2014, Vehtari et al., 2020). Ideally, we would run at least m parallel chains (at least one per site) for the different EP variants. However, NUTS is not well suited for running multiple chains on modern parallel hardware such as GPUs (Sountsov et al., 2024). To get the best practical performance from an EP-based black-box inference framework, it may be beneficial to consider alternative MCMC methods that are better suited to efficient implementation on modern parallel hardware (Hoffman et al., 2021). Performance may also be further improved by using asynchronous EP extensions in order to minimise the impact of communication-related overheads and delays (Hasenclever et al., 2017, Xu et al., 2014). Finally, the applications in this Chapter are all of a hierarchical Bayesian modelling flavour, and we would like to see our methods applied in more modern machine learning settings, such as Gaussian process regression / classification, or Bayesian deep learning. We leave investigation of all these ideas to future work.

6.8 Summary

In this chapter, we used a novel, natural-gradient interpretation of the moment-matching updates of EP to motivate two new EP variants that are far more robust to MC noise. We demonstrated that these variants can offer an improved speed-accuracy trade-off compared to their predecessors and are easier to tune. We also saw that EP is able to yield a more faithful posterior approximation than VI on challenging problems.

Despite our progress, some challenges remain. In particular, the burden of tuning hyperparameters with our new variants is still greater than that for direct MCMC methods. Furthermore, the practical performance of our EP variants is hindered by that of the underlying sampling kernels, leaving a large speed differential with efficient VI methods such as CVI. We believe that both of these challenges can be addressed

by exploiting similarities between the approximation, q , and the tilted distributions, $q_1, \dots, q_m$, and we suggested several ideas along these lines in Section 6.7. We are excited about the potential of these ideas to enable more widespread adoption of EP in black-box settings, offering both a more scalable alternative to direct MCMC and a more accurate alternative to VI.

Chapter 7

Conclusions and future work

In this thesis, we set out to advance methods for performing computationally efficient inference. Our developments drew heavily on natural gradient methods (Amari, 1998). Although the efficient application of (exact) natural gradients is largely confined to exponential family distributions (MCEF distributions being a notable exception), a common theme in our developments was the idea that the distribution used to compute the natural gradient direction is to some extent a free choice of the optimisation procedure.

In **Chapter 5**, we exploited this idea directly as a method for optimizing parameters of probability distributions where a direct application of natural gradient descent (NGD) would not otherwise be feasible. In particular, our method reframed an optimisation problem as one with respect to the parameters of an exponential family *surrogate* distribution, and performed NGD in that space. We demonstrated that this method was able to outperform alternative approaches on a number of maximum likelihood estimation and variational inference tasks, and in one case, we even found it to outperform *exact* natural gradient descent. The main limitation of this method is the need to find an appropriate distribution and (re)parameterisation with which to perform NGD. For the examples given in Chapter 5, these choices were largely guided by intuition, and there will not always be an *obvious* choice of surrogate distribution and parameterisation for any given problem. However, one interesting direction for future work would be to investigate *learned* surrogate parameterisations, which could broaden the potential use cases for this kind of approach. We believe the method presented in Chapter 5 is sufficiently easy to implement and understand, that it can be used effectively by practitioners in its current form, but we are excited to see whether the more general idea of using NGD surrogate distributions can be leveraged in other ways in the future.

In **Chapter 6**, we showed that expectation propagation (EP) – a popular Bayesian inference algorithm – can also be viewed as applying this same idea, but using a different NGD distribution at each step of the algorithm. We then used this insight to motivate two new EP variants whose updates are particularly well suited to Monte Carlo (MC) estimation. One of these variants, EP- η , has updates that are completely unbiased in the presence of MC noise, which is not true of (standard) EP. The other, EP- μ , has bias that shrinks at a faster rate than the expected progress, as the step size decreases, which again is not true of EP. We are most excited by the potential for these new variants to enable EP’s use as a framework for performing black-box inference at scale. We discussed the computational advantages of EP compared to a direct application of MCMC – the predominant approach to black-box inference – in Section 4.4.6. We also demonstrated the improved approximation quality obtained by EP (EP- η , more specifically) over a similarly structured variational inference solution in Section 6.6, and furthermore, the power parameter of (power) EP variants allows the practitioner to control the characteristics of the approximation, with intermediate values often providing superior results than either of the (standard) EP or VI extremes (Bui et al., 2017). Unfortunately, we also showed that, when applied naively, our new EP variants converge significantly slower than conjugate-computation variational inference (CVI) when measured in wall-clock time. However, we believe there is much scope to close this gap by exploiting the close relationship between the approximate posterior of EP and the tilted distributions – in Section 6.7 we gave several examples of ways this relationship could be exploited in order to accelerate performance of the sampling kernels, while also potentially obviating the need for hyperparameter tuning. With future developments such as these, we believe EP has the potential to be a more scalable alternative to direct MCMC, or a more accurate alternative to VI, in problems of black-box Bayesian inference.

In this thesis, we have presented advances in performing computationally efficient inference in a variety of settings, drawing heavily on natural-gradient-*based* methods. We hope these advances will be useful in their own right, but we are most excited to see future work that they may enable or inspire.

References

- S. Amari. *Differential-Geometrical Methods in Statistics*. Springer, 2012.
- S.-i. Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 1998.
- R. Anil, V. Gupta, T. Koren, K. Regan, and Y. Singer. Scalable second order optimization for deep learning. *arXiv:2002.09018*, 2021.
- A. Azzalini. *The Skew-Normal and Related Families*. Cambridge University Press, 2013.
- S. Baluja and R. Caruana. Removing the genetics from the standard genetic algorithm. *Machine Learning Proceedings 1995*, 1995.
- D. P. Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 1997.
- C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.
- J. Blackard. Coverttype. UCI Machine Learning Repository, 1998.
- T. Blanche. Multi-neuron recordings in primary visual cortex, 2009.
- M. Blondel, Q. Berthet, M. Cuturi, R. Frostig, S. Hoyer, F. Llinares-Lopez, F. Pedregosa, and J.-P. Vert. Efficient and modular implicit differentiation. *Advances in Neural Information Processing Systems*, 2022.
- A. Bouchard-Côté, T. Campbell, G. Pleiss, and N. Surjanovic. MCMC-driven learning. *arXiv:2402.09598*, 2024.
- J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018.
- S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2011.
- T. Bui, D. Hernandez-Lobato, J. Hernandez-Lobato, Y. Li, and R. Turner. Deep Gaussian processes for regression using approximate expectation propagation. *International Conference on Machine Learning*, 2016.

- T. D. Bui, J. Yan, and R. E. Turner. A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation. *Journal of Machine Learning Research*, 2017.
- T. D. Bui, C. V. Nguyen, S. Swaroop, and R. E. Turner. Partitioned variational inference: A unified framework encompassing federated and continual learning. *arXiv:1811.11206*, 2018.
- C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.
- B. Christianson. Reverse accumulation and attractive fixed points. *Optimization Methods & Software*, 1994.
- W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 2005.
- J. P. Cunningham, K. V. Shenoy, and M. Sahani. Fast Gaussian process methods for point process intensity estimation. *International Conference on Machine Learning*, 2008.
- J. P. Cunningham, P. Hennig, and S. Lacoste-Julien. Gaussian probabilities and expectation propagation. *arXiv:1111.6832*, 2013.
- Y. N. Dauphin, A. Agarwala, and H. Mobahi. Neglected hessian component explains mysteries in sharpness regularization. *Advances in Neural Information Processing Systems*, 2024.
- P. R. Dewick and S. Liu. Copula modelling to analyse financial data. *Journal of Risk and Financial Management*, 2022.
- R. A. Fisher. The negative binomial distribution. *Annals of Eugenics*, 1941.
- G. Frahm, M. Junker, and A. Szimayer. Elliptical copulas: applicability and limitations. *Statistics & Probability Letters*, 2003.
- J. R. Garcia, F. Freddi, S. Fotiadis, M. Li, S. Vakili, A. Bernacchia, and G. Hennequin. Fisher-legendre (fishleg) optimization of deep neural networks. *International Conference on Learning Representations*, 2023.
- F. García-Fernández, L. Svensson, and S. Särkkä. Iterated posterior linearization smoother. *IEEE Transactions on Automatic Control*, 2017.
- A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin. *Bayesian Data Analysis, Third Edition*. Chapman and Hall/CRC, 2013.
- T. George, C. Laurent, X. Bouthillier, N. Ballas, and P. Vincent. Fast approximate natural gradient descent in a kronecker-factored eigenbasis. *Advances in Neural Information Processing Systems*, 2018.
- J. Gonzalez, M. Osborne, and N. Lawrence. Glasses: Relieving the myopia of Bayesian optimisation. *International Conference on Artificial Intelligence and Statistics*, 2016.

- W. C. Guenther. A simple approximation to the negative binomial (and regular binomial). *Technometrics*, 1972.
- L. Hasenclever, S. Webb, T. Lienart, S. Vollmer, B. Lakshminarayanan, C. Blundell, and Y. W. Teh. Distributed Bayesian learning with stochastic natural gradient expectation propagation and the posterior server. *Journal of Machine Learning Research*, 2017.
- P. Hennig. Expectation propagation on the maximum of correlated normal variables. Technical report, University of Cambridge, 2009.
- P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 2012.
- P. Hennig, D. Stern, and T. Graepel. Coherent inference on optimal play in game trees. *International Conference on Artificial Intelligence and Statistics*, 2010.
- R. Herbrich, T. Minka, and T. Graepel. Trueskill™: A Bayesian skill rating system. *Advances in Neural Information Processing Systems*, 2006.
- D. Hernandez-Lobato and J. M. Hernandez-Lobato. Scalable Gaussian process classification via expectation propagation. *International Conference on Artificial Intelligence and Statistics*, 2016.
- J. Hernandez-Lobato, Y. Li, M. Rowland, T. Bui, D. Hernandez-Lobato, and R. Turner. Black-box alpha divergence minimization. *International Conference on Machine Learning*, 2016.
- J. M. Hernández-Lobato and R. P. Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. *International Conference on Machine Learning*, 2015.
- L. Hernández, J. Tejero, and J. Vinuesa. Maximum likelihood estimation of the correlation parameters for elliptical copulas. *arXiv:1412.6316*, 2014.
- T. Heskes. On “natural” learning and pruning in multilayered perceptrons. *Neural Computation*, 2000.
- T. Heskes and O. Zoeter. Expectation propagation for approximate inference in dynamic Bayesian networks. *Uncertainty in Artificial Intelligence*, 2002.
- T. Heskes and O. Zoeter. Extended version of “Expectation propagation for approximate inference in dynamic Bayesian networks”. Technical report, University of Nijmegen, 2003.
- M. Hoffman, A. Radul, and P. Sountsov. An adaptive-mcmc scheme for setting trajectory lengths in hamiltonian monte carlo. *International Conference on Artificial Intelligence and Statistics*, 2021.
- M. D. Hoffman and A. Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 2014.

- M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.
- P. A. d. F. R. Højén-Sørensen, O. Winther, and L. K. Hansen. Mean-field approaches to independent component analysis. *Neural Computation*, 2002.
- A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 2005.
- W. Jitkrittum, A. Gretton, N. M. O. Heess, S. M. A. Eslami, B. Lakshminarayanan, D. Sejdinovic, and Z. Szabó. Kernel-based just-in-time learning for passing expectation propagation messages. *Uncertainty in Artificial Intelligence*, 2015.
- P. Jylänki, J. Vanhatalo, and A. Vehtari. Robust Gaussian process regression with a student- t likelihood. *Journal of Machine Learning Research*, 2011.
- S. M. Kakade. A natural policy gradient. *Advances in Neural Information Processing Systems*, 2001.
- M. Kendall, D. Tsallis, C. Wymant, A. Di Francia, Y. Balogun, X. Didelot, L. Ferretti, and C. Fraser. Epidemiological impacts of the nhs covid-19 app in england and wales throughout its first year. *Nature Communications*, 2023.
- A. Kerekes, A. Mészáros, and F. Huszár. Depth without the magic: Inductive bias of natural gradient descent. *arXiv:2111.11542*, 2021.
- M. Khan and W. Lin. Conjugate-Computation Variational Inference : Converting Variational Inference in Non-Conjugate Models to Inferences in Conjugate Models. *International Conference on Artificial Intelligence and Statistics*, 2017.
- M. Khan, D. Nielsen, V. Tangkaratt, W. Lin, Y. Gal, and A. Srivastava. Fast and scalable Bayesian deep learning by weight-perturbation in Adam. *International Conference on Machine Learning*, 2018.
- M. E. Khan and D. Nielsen. Fast yet simple natural-gradient descent for variational inference in complex models. *International Symposium on Information Theory and Its Applications*, 2018.
- M. E. Khan and H. Rue. The bayesian learning rule. *Journal of Machine Learning Research*, 2023.
- M. E. Khan, W. Lin, V. Tangkaratt, Z. Liu, and D. Nielsen. Variational adaptive-newton method. *NeurIPS Workshop on Advances in Approximate Bayesian Inference*, 2017.
- D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2014.
- D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations*, 2014.
- E. M. Kiral, T. Moellenhoff, and M. E. Khan. The lie-group bayesian learning rule. *International Conference on Artificial Intelligence and Statistics*, 2023.

- D. Knowles and T. Minka. Non-conjugate variational message passing for multinomial and binary regression. *Advances in Neural Information Processing Systems*, 2011.
- F. Kunstner, P. Hennig, and L. Balles. Limitations of the empirical fisher approximation for natural gradient descent. *Advances in Neural Information Processing Systems*, 2019.
- P. Larrañaga and J. A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Springer, 2002.
- J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient descent only converges to minimizers. *Conference on Learning Theory*, 2016.
- B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics*. Cambridge University Press, 2005.
- Y. Li, J. M. Hernández-Lobato, and R. E. Turner. Stochastic expectation propagation. *Advances in Neural Information Processing Systems*, 2015.
- W. Lin, M. E. Khan, and M. Schmidt. Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. *International Conference on Machine Learning*, 2019.
- W. Lin, M. Schmidt, and M. E. Khan. Handling the positive-definite constraint in the Bayesian learning rule. *International Conference on Machine Learning*, 2020.
- W. Lin, F. Nielsen, K. M. Emtiyaz, and M. Schmidt. Tractable structured natural-gradient descent using local parameterizations. *International Conference on Machine Learning*, 2021.
- J. O. Lloyd-Smith. Maximum likelihood estimation of the negative binomial dispersion parameter for highly overdispersed data, with applications to infectious diseases. *PLOS ONE*, 2007.
- J. O. Lloyd-Smith, S. Schreiber, P. Kopp, and W. Getz. Superspreading and the effect of individual variation on disease emergence. *Nature*, 2005.
- J. Lopez-Martin, J. Phillips, and A. Gelman. Multilevel regression and poststratification case studies, 2022.
- L. Malagò and G. Pistone. Information geometry of the gaussian distribution in view of stochastic optimization. *ACM Conference on Foundations of Genetic Algorithms*, 2015.
- L. Malagò, M. Matteucci, and G. Pistone. Stochastic relaxation as a unifying approach in 0/1 programming. *NIPS 2009 Workshop on Discrete Optimization in Machine Learning: Submodularity, Sparsity & Polyhedra*, 2009.
- L. Malagò, M. Matteucci, and P. Giovanni. Stochastic natural gradient descent by estimation of empirical covariances. *IEEE Congress of Evolutionary Computation*, 2011.

- J. Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 2020.
- J. Martens and R. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. *International Conference on Machine Learning*, 2015.
- T. Minka. Estimating a dirichlet distribution. Technical report, Microsoft, September 2000.
- T. Minka. Estimating a gamma distribution. Technical report, Microsoft, 2002.
- T. Minka. Power ep. Technical report, Microsoft, 2004.
- T. Minka. The ep energy function and minimization schemes. August 2007.
- T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. *Uncertainty in Artificial Intelligence*, 2002.
- T. Minka, R. Cleven, and Y. Zaykov. Trueskill 2: An improved Bayesian skill rating system. Technical report, Microsoft, 2018.
- T. P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.
- K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.
- A. Naish-guzman and S. Holden. The generalized fitc approximation. *Advances in Neural Information Processing Systems*, 2007.
- R. M. Neal. *Monte Carlo Implementation*. 1996.
- A. Nemirovsky and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, 1983.
- J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2006.
- U. Nodelman, D. Koller, and C. Shelton. Expectation propagation for continuous time Bayesian networks. *Uncertainty in Artificial Intelligence*, 2005.
- Y. Ollivier, L. Arnold, A. Auger, and N. Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *Journal of Machine Learning Research*, 2017.
- M. Opper and O. Winther. Gaussian processes for classification: Mean-field algorithms. *Neural Computation*, 2000.
- M. Opper, U. Paquet, and O. Winther. Improving on expectation propagation. *Advances in Neural Information Processing Systems*, 2008.
- A. Orooji, E. Nazar, M. Sadeghi, A. Moradi, Z. Jafari, and H. a. Esmaily. Factors associated with length of stay in hospital among the elderly patients using count regression models. *Medical Journal of the Islamic Republic Of Iran*, 2021.

- K. Osawa, S. Swaroop, A. Jain, R. Eschenhagen, R. E. Turner, R. Yokota, and M. E. Khan. Practical deep learning with bayesian principles. *Advances in Neural Information Processing Systems*, 2019.
- G. Papamakarios, T. Pavlakou, and I. Murray. Masked autoregressive flow for density estimation. *Advances in Neural Information Processing Systems*, 2017.
- M. Pelikan, D. E. Goldberg, and F. G. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 2002.
- D. Phan, N. Pradhan, and M. Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro, 2019.
- G. Raskutti and S. Mukherjee. The information geometry of mirror descent. *IEEE Transactions on Information Theory*, 2015.
- I. Rechenberg and M. Eigen. *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog, 1973.
- Y. Ren and D. Goldfarb. Efficient subsampled Gauss-Newton and natural gradient methods for training neural networks. *arXiv:1906.02353*, 2019.
- H. Robbins and S. Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 1951.
- B. Roe. MiniBooNE particle identification. UCI Machine Learning Repository, 2010.
- F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 1958.
- T. Salimans and D. Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 2012.
- H. Salimbeni, S. Eleftheriadis, and J. Hensman. Natural gradients in practice: Non-conjugate variational inference in Gaussian process models. *International Conference on Artificial Intelligence and Statistics*, 2018.
- M.-A. Sato. Online model selection based on the variational Bayes. *Neural Computation*, 2001.
- B. Schaffner, S. Ansolabehere, and S. Luks. CCES common content 2018, 2019.
- M. Seeger. Expectation propagation for exponential families. 2008.
- M. Seeger and H. Nickisch. Fast convergent algorithms for expectation propagation approximate Bayesian inference. *International Conference on Artificial Intelligence and Statistics*, 2011.
- Y. Shen, N. Daheim, B. Cong, P. Nickl, G. M. Marconi, B. C. E. M. Raoul, R. Yokota, I. Gurevych, D. Cremers, M. E. Khan, and T. Möllenhoff. Variational learning is effective for large deep networks. *International Conference on Machine Learning*, 2024.

- J. So. Estimating the normal-inverse-wishart distribution. *arXiv:2405.16088*, 2024.
- J. So and R. E. Turner. Optimising distributions with natural gradient surrogates. *International Conference on Artificial Intelligence and Statistics*, 2024a.
- J. So and R. E. Turner. Fearless stochasticity in expectation propagation. *Advances in Neural Information Processing Systems*, 2024b.
- P. Sountsov, C. Carroll, and M. D. Hoffman. Running markov chain monte carlo on modern hardware and software, 2024.
- J. Staines and D. Barber. Variational optimization. *arXiv:1906.02353*, 2012.
- D. Tang and R. Ranganath. The variational predictive natural gradient. *International Conference on Machine Learning*, 2019.
- L. Tatzel, P. Hennig, and F. Schneider. Late-phase second-order training. *Has it Trained Yet? NeurIPS Workshop*, 2022.
- M. Trottini and F. Spezzaferri. A generalized predictive criterion for model selection. *The Canadian Journal of Statistics*, 2002.
- R. E. Turner and M. Sahani. Probabilistic amplitude and frequency demodulation. *Advances in Neural Information Processing Systems*, 2011.
- A. Vehtari, A. Gelman, T. Sivula, P. Jylänki, D. Tran, S. Sahai, P. Blomstedt, J. P. Cunningham, D. Schiminovich, and C. P. Robert. Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data. *Journal of Machine Learning Research*, 2020.
- M. Wainwright and M. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 2008.
- M. Welling and J. J. Lim. A distributed message passing algorithm for sensor localization. *International Conference on Artificial Neural Networks*, 2007.
- W. Wiegand and T. Heskes. Fractional belief propagation. *Advances in Neural Information Processing Systems*, 2002.
- D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 2014.
- Wikipedia contributors. Normal-inverse-wishart distribution — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Normal-inverse-Wishart_distribution&oldid=1184912425, 2023a.
- Wikipedia contributors. Polygamma function — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Polygamma_function&oldid=1171994095, 2023b.
- W. J. Wilkinson, S. Särkkä, and A. Solin. Bayes-Newton methods for approximate Bayesian inference with PSD guarantees. *Journal of Machine Learning Research*, 2023.

-
- J. Winn and C. M. Bishop. Variational message passing. *Journal of Machine Learning Research*, 2005.
- M. Xu, B. Lakshminarayanan, Y. W. Teh, J. Zhu, and B. Zhang. Distributed Bayesian posterior sampling via moment sharing. *Advances in Neural Information Processing Systems*, 2014.
- O. Zoeter and T. Heskes. Gaussian quadrature based expectation propagation. *International Workshop on Artificial Intelligence and Statistics*, 2005.

Appendix A

Gradient notation

In this section we introduce our notation for gradients and related quantities. We largely follow the notation of Bertsekas (1997), with one addition. For function $f : \mathbb{R}^m \rightarrow \mathbb{R}$, the gradient at x , assuming all partial derivatives exist, is given by

$$\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_m} \right]^\top. \quad (\text{A.1})$$

Note, in particular, that this implies $\nabla f(g(y))$ is the gradient of f evaluated at $x = g(y)$. The Hessian of f , denoted $\nabla^2 f(x)$, is the matrix with entries given by

$$[\nabla^2 f(x)]_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}. \quad (\text{A.2})$$

If $f : \mathbb{R}^{m+n}$ is a function of $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$, then

$$\nabla_x f(x, y) = \left[\frac{\partial f(x, y)}{\partial x_1}, \frac{\partial f(x, y)}{\partial x_2}, \dots, \frac{\partial f(x, y)}{\partial x_m} \right]^\top \quad (\text{A.3})$$

with $\nabla_y f(x, y)$ defined similarly.

When $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a vector-valued function, the *gradient matrix* of f , denoted $\nabla f(x)$, is the transpose of the Jacobian of f . That is, the matrix with i -th column equal to the gradient of f_i , the i -th component of f :

$$\nabla f(x) = [\nabla f_1(x) \dots \nabla f_n(x)]. \quad (\text{A.4})$$

Where ∇ is immediately followed by a bracketed expression, we use this to denote the gradient of an anonymous function, with definition given by the bracketed expression,

and gradient taken with respect to the subscript, e.g.

$$\nabla_x(2f(x)) = 2\nabla f(x), \quad (\text{A.5})$$

and in these cases (only), evaluation of the gradient at, e.g. $x = g(y)$, is denoted

$$\nabla_x(2f(x))\Big|_{g(y)}. \quad (\text{A.6})$$

Finally, in some cases, where e.g. θ are parameters of a probability q distribution, we will use $\tilde{\nabla}$ in place of ∇ above to denote the *natural gradient* with respect to q and θ . That is, e.g.

$$\tilde{\nabla} f(\theta) = [F(\theta)]^{-1} \nabla f(\theta), \quad (\text{A.7})$$

where $F(\theta)$ is the Fisher information matrix of q with respect to parameters θ . The distribution q should be clear from context.

Appendix B

Exponential family examples

In this appendix we provide examples of the exponential family formulation of the Gamma distribution, the normal-inverse Wishart distribution, and exponential family mixture models.

B.1 Gamma distribution

The gamma distribution is a common univariate distribution for modelling positive-valued continuous random variables. It is conventionally parameterised in terms of parameters $\alpha > 0$ and $\beta > 0$, known as the shape and rate, respectively, with density function given by

$$p(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \quad (\text{B.1})$$

$$=: \text{Gamma}(x; \alpha, \beta), \quad (\text{B.2})$$

where $\Gamma(\cdot)$ is the gamma function. (B.2) can be rewritten in the form (2.1) by defining the sufficient statistics function $s(x) = (\log x, x)$, base measure $\nu(x) = 1$, natural parameters

$$\begin{aligned} \eta &= (\alpha - 1, -\beta) \\ &=: (\eta_1, \eta_2), \end{aligned} \quad (\text{B.3})$$

and log partition function

$$A(\eta) = \log \Gamma(\eta_1 + 1) - (\eta_1 + 1) \log(-\eta_2). \quad (\text{B.4})$$

The mean parameters of the Gamma distribution can be found by evaluating the forward mapping, giving

$$\begin{aligned}\mu &= \left(\mathbb{E}_{p(x)}[\log x], \mathbb{E}_{p(x)}[x] \right) \\ &= \left(\psi(\eta_1 + 1) - \log(-\eta_2), -(\eta_1 + 1)/\eta_2 \right).\end{aligned}\tag{B.5}$$

The backwards mapping, $\nabla A^*(\mu)$, is not available in closed form, but can be computed efficiently by solving a 1-dimensional optimisation problem (Minka, 2002).

B.2 Normal-inverse Wishart distribution

The normal-inverse-Wishart (NIW) distribution is commonly used as a prior distribution for the mean and covariance parameters of a d -dimensional multivariate normal distribution. In this section we describe a new procedure for performing the backwards mapping in the NIW family, which is based on a short technical note that we posted on arXiv in 2024 entitled “*Estimating the normal-inverse Wishart distribution*” (So, 2024). The normal-inverse-Wishart distribution with parameters ξ_0 , λ , Ψ and κ , has density (Wikipedia contributors, 2023a)

$$p(\xi, \Sigma) = \frac{\lambda^{\frac{d}{2}} |\Psi|^{\frac{\kappa}{2}} |\Sigma|^{-\frac{\kappa+d+2}{2}}}{(2\pi)^{\frac{d}{2}} 2^{\frac{\kappa d}{2}} \Gamma_d(\frac{\kappa}{2})} \exp \left(-\frac{1}{2} \text{Tr}(\Psi \Sigma^{-1}) - \frac{\lambda}{2} (\xi - \xi_0) \Sigma^{-1} (\xi - \xi_0) \right). \tag{B.6}$$

We can recognise (B.6) as having the form of an exponential family distribution density, with statistic function

$$s(\xi, \Sigma) = \left(-\frac{1}{2} \Sigma^{-1}, \Sigma^{-1} \xi, -\frac{1}{2} \xi^\top \Sigma^{-1} \xi, -\frac{1}{2} \log |\Sigma| \right) \tag{B.7}$$

base measure $\nu(\xi, \Sigma) = |\Sigma|^{-\frac{d+2}{2}}$, natural parameters

$$\begin{aligned}\eta &= \left(\Psi + \lambda \xi_0 \xi_0^\top, \lambda \xi_0, \lambda, \kappa \right) \\ &=: (\eta_1, \eta_2, \eta_3, \eta_4),\end{aligned}\tag{B.8}$$

and log partition function

$$A(\eta) = -\frac{d}{2} \log \eta_3 - \frac{\eta_4}{2} \log \left| \eta_1 - \frac{\eta_2 \eta_2^\top}{\eta_3} \right| + \frac{d}{2} \log(2\pi) + \frac{\eta_4 d}{2} \log 2 + \log \Gamma_d(\eta_4/2), \tag{B.9}$$

where $\Gamma_d(\cdot)$ is the d -dimensional multivariate Gamma function. Note that here we have a symmetry constraint on Ψ , which is equivalent to working with an unconstrained minimal representation (see the remark about symmetry constraints in Section 2.3).

We now describe how to perform the map from natural to mean parameters, a.k.a. the *forward mapping*. Let p be the density of the NIW distribution with natural parameters η . The log partition function is given by

$$A(\eta) = -\frac{d}{2} \log \eta_3 - \frac{\eta_4}{2} \log \left| \eta_1 - \frac{\eta_2 \eta_2^\top}{\eta_3} \right| + \frac{d}{2} \log(2\pi) + \frac{\eta_4 d}{2} \log 2 + \log \Gamma_d(\eta_4/2). \quad (\text{B.10})$$

Using the standard result for exponential families, $E_{p(\xi, \Sigma)}[s(\xi, \Sigma)] = \nabla A(\eta)$ (Wainwright and Jordan, 2008), we have

$$\begin{aligned} \frac{\partial A}{\partial \eta_1} &= \text{vec} \left(-\frac{\eta_4}{2} \left(\eta_1 - \frac{\eta_2 \eta_2^\top}{\eta_3} \right)^{-1} \right) \\ &= -\frac{\kappa}{2} \text{vec}(\Psi^{-1}) \\ &=: \mu_1, \end{aligned} \quad (\text{B.11})$$

$$\begin{aligned} \frac{\partial A}{\partial \eta_2} &= \frac{\eta_4}{\eta_3} \left(\eta_1 - \frac{\eta_2 \eta_2^\top}{\eta_3} \right)^{-1} \eta_2 \\ &= \kappa \Psi^{-1} \xi_0 \\ &=: \mu_2, \end{aligned} \quad (\text{B.12})$$

$$\begin{aligned} \frac{\partial A}{\partial \eta_3} &= -\frac{d}{2\eta_3} - \frac{\eta_4}{2\eta_3^2} \text{Tr} \left(\left(\eta_1 - \frac{\eta_2 \eta_2^\top}{\eta_3} \right)^{-1} \eta_2 \eta_2^\top \right) \\ &= -\frac{d}{2\lambda} - \frac{\kappa}{2} \text{Tr}(\Psi^{-1} \xi_0 \xi_0^\top) \\ &=: \mu_3, \end{aligned} \quad (\text{B.13})$$

$$\begin{aligned} \frac{\partial A}{\partial \eta_4} &= -\frac{1}{2} \log \left| \eta_1 - \frac{\eta_2 \eta_2^\top}{\eta_3} \right| + \frac{d}{2} \log 2 + \frac{1}{2} \sum_{i=0}^{d-1} \psi \left(\frac{\eta_4 - i}{2} \right) \\ &= -\frac{1}{2} \log |\Psi| + \frac{d}{2} \log 2 + \frac{1}{2} \sum_{i=0}^{d-1} \psi \left(\frac{\kappa - i}{2} \right) \\ &=: \mu_4, \end{aligned} \quad (\text{B.14})$$

so that $\mu = (\mu_1, \mu_2, \mu_3, \mu_4)$.

We now describe a novel procedure for efficiently performing the map from mean to natural parameters, a.k.a. the *backward mapping*. Let $M_1 = \text{vec}^{-1}(\mu_1)$; then, we have

$$\xi_0 = (-2M_1)^{-1}\mu_2 \quad (\text{B.15})$$

$$\lambda = -d(2\mu_3 + \mu_2^\top \xi_0)^{-1} \quad (\text{B.16})$$

$$\Psi = \left(-\frac{2}{\kappa} M_1 \right)^{-1}. \quad (\text{B.17})$$

Unfortunately κ cannot be solved for in closed form, but we can find it by performing a 1-dimensional optimisation. We want to find κ such that

$$-\frac{1}{2} \log |\Psi| + \frac{d}{2} \log 2 + \frac{1}{2} \sum_{i=0}^{d-1} \psi\left(\frac{\kappa - i}{2}\right) = \mu_4, \quad (\text{B.18})$$

or equivalently,

$$\begin{aligned} 0 &= -\log |\Psi| + d \log 2 + \sum_{i=0}^{d-1} \psi\left(\frac{\kappa - i}{2}\right) - 2\mu_4 \\ &= -\log \left| \left(-\frac{2}{\kappa} M_1 \right)^{-1} \right| + d \log 2 + \sum_{i=0}^{d-1} \psi\left(\frac{\kappa - i}{2}\right) - 2\mu_4 \\ &= \log |-2M_1| - d \log \frac{\kappa}{2} + \sum_{i=0}^{d-1} \psi\left(\frac{\kappa - i}{2}\right) - 2\mu_4 \\ &=: f(\kappa). \end{aligned} \quad (\text{B.19})$$

Taking first derivatives, we have

$$f'(\kappa) = -\frac{d}{\kappa} + \frac{1}{2} \sum_{i=0}^{d-1} \psi^{(1)}\left(\frac{\kappa - i}{2}\right). \quad (\text{B.20})$$

Note that the density p is only defined for $\kappa > d - 1$. Using the inequality (Wikipedia contributors, 2023b)

$$\frac{x + \frac{1}{2}}{x^2} \leq \psi^{(1)}(x), \quad (\text{B.21})$$

we have

$$\begin{aligned}
 f'(\kappa) &\geq -\frac{d}{\kappa} + \sum_{i=0}^{d-1} \frac{(\kappa - i) + 1}{(\kappa - i)^2} \\
 &\geq -\frac{d}{\kappa} + \frac{d(\kappa + 1)}{\kappa^2} \\
 &= \frac{d}{\kappa^2} > 0,
 \end{aligned} \tag{B.22}$$

hence $f'(\kappa) > 0$, and $f(\kappa)$ is strictly increasing for $\kappa > d - 1$. Taking second derivatives, we have

$$f''(\kappa) = \frac{d}{\kappa^2} + \frac{1}{4} \sum_{i=0}^{d-1} \psi^{(2)}\left(\frac{\kappa - i}{2}\right), \tag{B.23}$$

and using the inequality (Wikipedia contributors, 2023b)

$$\psi^{(2)}(x) \leq -\frac{1}{x^2} - \frac{1}{x^3}, \tag{B.24}$$

we have the bound

$$\begin{aligned}
 f''(\kappa) &\leq \frac{d}{\kappa^2} - \frac{1}{4} \sum_{i=0}^{d-1} \frac{1}{\left(\frac{\kappa - i}{2}\right)^2} + \frac{1}{\left(\frac{\kappa - i}{2}\right)^3} \\
 &= \frac{d}{\kappa^2} - \sum_{i=0}^{d-1} \frac{1}{(\kappa - i)^2} + \frac{2}{(\kappa - i)^3} \\
 &\leq \frac{d}{\kappa^2} - \frac{d}{\kappa^2} - \frac{2d}{\kappa^3} \\
 &= -\frac{2d}{\kappa^3} < 0,
 \end{aligned} \tag{B.25}$$

and so $f(\kappa)$ is also strictly concave for $\kappa > d - 1$. The exponential family of NIW distributions is minimal, and so we know that there is a *unique* κ for which $f(\kappa) = 0$ (Wainwright and Jordan, 2008). Taken together, we have that $f(\kappa)$ for $\kappa > d - 1$ is strictly increasing, strictly concave, and has a single root; this tells us that if we find a starting point κ for which $f(\kappa) < 0$, the Newton-Raphson method is guaranteed to converge to the unique root. We can easily find such a point by taking any initial guess $\kappa > d - 1$ and halving the distance to between κ and $d - 1$ until $f(\kappa) < 0$. Once we have found ξ_0 , λ , Ψ and κ , the natural parameters are given by (B.8).

Pseudocode for the resulting procedure is given by Algorithm 5. κ_0 is some initial guess for κ which could default to e.g. $\kappa_0 = d$. ϵ is some small tolerance for detecting convergence.

Algorithm 5 Normal-inverse-Wishart mean to natural parameter map

Require: $(\mu_1, \mu_2, \mu_3, \mu_4)$

Require: $\kappa_0 > d - 1$

Require: $\epsilon > 0$

$\kappa \leftarrow \kappa_0$

$M_1 \leftarrow \text{vec}^{-1}(\mu_1)$

$f(\kappa) := \log|-2M_1| - d \log \frac{\kappa}{2} + \sum_{i=0}^{d-1} \psi\left(\frac{\kappa-i}{2}\right) - 2\mu_4$

while $f(\kappa) > 0$ **do**

$\kappa \leftarrow (\kappa - d + 1)/2$

while $|f(\kappa)| > \epsilon$ **do**

$\kappa \leftarrow \kappa - f(\kappa)/f'(\kappa)$

$\xi_0 \leftarrow (-2M_1)^{-1}\mu_2$

$\lambda \leftarrow -d(2\mu_3 + \mu_2^\top \xi_0)^{-1}$

$\Psi \leftarrow (-\frac{2}{\kappa}M_1)^{-1}$

return $(\text{vec}(\Psi + \lambda\xi_0\xi_0^\top), \lambda\xi_0, \lambda, \kappa)$

B.3 Mixture models

A mixture distribution expresses a density as a convex combination of k simpler densities:

$$p(x) = \sum_{i=1}^k \pi_i p_i(x), \quad (\text{B.26})$$

where $\pi \in \Delta^{k-1}$, and p_i is known as the i -th *component* distribution. For each mixture, we can also define a corresponding mixture *model*, a joint distribution, with density

$$p(z, x) = \pi_z p_z(x), \quad (\text{B.27})$$

where the mixture component identity, $z \in \{1, \dots, k\}$, is treated as a random variable. Note then, that $p(x) = \sum_{i=1}^k p(z = i, x)$. That is, a mixture model has a mixture as its *marginal*. A mixture of exponential family distributions is not in general itself in an exponential family. However, it can be shown that the corresponding mixture

model (joint distribution) is an exponential family distribution¹ An exponential family mixture with k components from the same family has density

$$p(x) = \sum_{i=1}^k \pi_i \nu(x) \exp \left(s(x)^\top \eta_i - A(\eta_i) \right) \quad (\text{B.28})$$

where $\pi_i > 0$ are the mixture probabilities, with $\pi \in \Delta^{k-1}$, and η_i are the natural parameters of mixture component i . The corresponding mixture *model* then has density

$$\begin{aligned} p(z, x) &= \pi_z \nu(x) \exp \left(s(x)^\top \eta_z - A(\eta_z) \right) \\ &= \sum_{i=1}^k \mathbb{I}_i(z) \pi_i \nu(x) \exp \left(s(x)^\top \eta_i - A(\eta_i) \right) \\ &= \nu(x) \prod_{i=1}^k \left[\pi_i \exp \left(s(x)^\top \eta_i - A(\eta_i) \right) \right]^{\mathbb{I}_i(z)} \\ &= \nu(x) \exp \left(\sum_{i=1}^k \mathbb{I}_i(z) \log \pi_i + \mathbb{I}_i(z) s(x)^\top \eta_i - \mathbb{I}_i(z) A(\eta_i) \right) \\ &= \nu(x) \exp \left[\left(\sum_{i=1}^{k-1} \mathbb{I}_i(z) \left(\log \frac{\pi_i}{\pi_k} - A(\eta_i) + A(\eta_k) \right) \right) \right. \\ &\quad \left. + \left(\sum_{i=1}^k \mathbb{I}_i(z) s(x)^\top \eta_i \right) + \log \pi_k - A(\eta_k) \right], \end{aligned} \quad (\text{B.29})$$

where $\mathbb{I}_i(z)$ is an indicator function for the singleton set $\{i\}$. We can recognise (B.29) as having the form of the exponential family with base measure $\tilde{\nu}(z, x) = \nu(x)$, and statistics function

$$\tilde{s}(z, x) = \left(\mathbb{I}_1(z), \dots, \mathbb{I}_{k-1}(z), \mathbb{I}_1(z)s(x), \dots, \mathbb{I}_k(z)s(x) \right), \quad (\text{B.30})$$

where $\mathbb{I}_i(z)$ is an indicator function for the singleton set $\{i\}$, and we have used the tilde notation to distinguish between the component and mixture model functions. The natural parameters are given by $\eta = (\alpha_1, \dots, \alpha_{k-1}, \eta_1, \dots, \eta_k)$, where $\alpha_i :=$

¹This was shown for a univariate normal mixture model by Wainwright and Jordan (2008). While the general result must surely be well known, we have been unable to find a reference. The closest result we are aware of is one by Lin et al. (2019), which states that a mixture model lies in the more general class of *minimal conditional exponential family* distributions.

$\log \frac{\pi_i}{\pi_k} - A(\eta_i) + A(\eta_k)$ for $i = 1, \dots, k-1$, and the log-partition function is given by

$$\tilde{A}(\eta) = \log \left(e^{A(\eta_k)} + \sum_{i=1}^{k-1} e^{\alpha_i + A(\eta_i)} \right). \quad (\text{B.31})$$

The forward mapping is given by $\nabla \tilde{A}(\eta) = (\pi_1, \dots, \pi_{k-1}, \tilde{\mu}_1, \dots, \tilde{\mu}_k)$ where $\tilde{\mu}_i = \pi_i \nabla A(\eta_i)$ for $i = 1, \dots, k$, and

$$\pi_i = \begin{cases} \frac{e^{\alpha_i + A(\eta_i)}}{e^{A(\eta_k)} + \sum_{j=1}^{k-1} e^{\alpha_j + A(\eta_j)}} & \text{for } i = 1, \dots, k-1 \\ \frac{e^{A(\eta_k)}}{e^{A(\eta_k)} + \sum_{j=1}^{k-1} e^{\alpha_j + A(\eta_j)}} & \text{for } i = k \end{cases}. \quad (\text{B.32})$$

Provided the component distributions are from a minimal family, the statistics $\tilde{s}(z, x)$ are linearly independent, and so the mixture model family is also minimal. In that case, the backwards mapping is obtained using $\eta_i = \nabla A^*(\tilde{\mu}_i/\pi_i)$ and $\pi_k = 1 - \sum_{i=1}^{k-1} \pi_i$.

B.4 Exponential family natural gradients

When NGD is applied to objectives involving a KL divergence (or related quantity), it has special properties when it is performed with respect to either the natural or mean parameters of an EF distribution, as the following propositions show.

Proposition 1. *For EF q with mean parameters $\mu \in \mathcal{M}^\circ$, and mean domain $\mathcal{M}^\circ$, let*

$$f(\mu) = -\mathbb{E}_{p^*(x)}[\log q_\mu(x)] + c \quad (3.12)$$

where p^* is any distribution with $\text{supp}[p^*] \subseteq \text{supp}[q]$, and c is constant with respect to μ . Then, $\forall \mu \in \mathcal{M}^\circ$,

$$\mu - \tilde{\nabla} f(\mu) = \arg \min_{\mu' \in \mathcal{M}^\circ} f(\mu'). \quad (3.13)$$

Proof. First, let η be dually coupled with μ , so that $\mu = \mathbb{E}_{q_\eta(x)}[s(x)]$. From (3.11), we have that

$$\begin{aligned} \mu - \tilde{\nabla} f(\mu) &= \mu + \nabla_\eta \left(\mathbb{E}_{p^*(x)} [\log q_\eta(x)] \right) \\ &= \mu + \nabla_\eta \left(\mathbb{E}_{p^*(x)} [s(x)^\top \eta - A(\eta) + \log \nu(x)] \right) \\ &= \mu - \nabla A(\eta) + \mathbb{E}_{p^*(x)} [s(x)] \\ &= \mathbb{E}_{p^*(x)} [s(x)] \end{aligned} \quad (B.33)$$

where we have used the standard EF identity $\nabla A(\eta) = \mu(\eta)$. It remains to show that $\mathbb{E}_{p^*(x)}[s(x)] = \arg \min_{\mu' \in \mathcal{M}^\circ} f(\mu')$. Let us reparameterise $f(\mu)$ as $f_\eta(\eta) = f(\mu(\eta))$, then

$$\begin{aligned} \nabla f_\eta(\eta) &= -\nabla_\eta \left(\mathbb{E}_{p^*(x)} [s(x)^\top \eta - A(\eta) + \log \nu(x)] \right) \\ &= \mu - \mathbb{E}_{p^*(x)} [s(x)], \end{aligned} \quad (B.34)$$

and equating to zero, we find the unique stationary point at $\mu = \mathbb{E}_{p^*(x)}[s(x)]$. To show that this is a minimum, note that

$$\nabla^2 f_\eta(\eta) = \nabla^2 A(\eta). \quad (B.35)$$

Given that A is strictly convex and twice-differentiable (Wainwright and Jordan, 2008), then f_η is also strictly convex, and so must be minimised when $\mu = \mathbb{E}_{p^*(x)}[s(x)]$. $\square$

Proposition 2. For EF q with natural parameters $\eta \in \Omega$, and natural domain Ω , let

$$f(\eta) = \text{KL}[q_\eta \parallel p^*] + c \quad (3.14)$$

where p^* is from the same EF as q . Then, $\forall \eta \in \Omega$,

$$\eta - \tilde{\nabla} f(\eta) = \arg \min_{\eta' \in \Omega} f(\eta'). \quad (3.15)$$

Proof. First, let μ be dually coupled with η , so that $\mu = \mathbb{E}_{q_\eta(x)}[s(x)]$, and let η^* be the natural parameters of p^* . From (3.10), we have that

$$\begin{aligned} \eta - \tilde{\nabla} f(\eta) &= \eta - \nabla_\mu (\text{KL}[q_\mu \parallel p^*]) \\ &= \eta - \nabla_\mu (A^*(\mu) + A(\eta^*) - \mu^\top \eta^*) \\ &= \eta - \nabla A^*(\mu) + \eta^* \\ &= \eta^* \end{aligned} \quad (\text{B.36})$$

where q_μ denotes the EF distribution with mean parameter μ , and we have used the identity $\nabla A^*(\mu) = \eta$. It is clear from the properties of the KL divergence that η^* must be the unique minimiser for $f(\eta)$. $\square$

Appendix C

Appendices for Chapter 5

C.1 Equivalence with optimisation of f

In this appendix we show that under certain conditions, optimising $\tilde{f}$ is equivalent to optimising f in the following sense: finding a local minimiser of $\tilde{f}$ also gives us a local minimiser for f (Proposition 6), and all local minima of f are attainable through $\tilde{f}$ (Proposition 7). Furthermore, we show that $\tilde{f}$ does not have any non-strict saddle points that are not also present at the corresponding points in f (Proposition 10, with support from Propositions 8 and 9). Note that the results derived here are more general than those that are summarised in Chapter 5.

Let Θ and $\tilde{\Theta}$ be differentiable manifolds of dimension i and j respectively, where $i \leq j$. Let $g : \tilde{\Theta} \rightarrow \Theta$ be a twice differentiable submersion on $\tilde{\Theta}$, with $g(\tilde{\Theta}) = \Theta$. Let $f : \Theta \rightarrow \mathbb{R}$ be twice continuously differentiable, and define $\tilde{f} = f \circ g$.

Proposition 6. *$\tilde{f}$ has a local minimum at $\tilde{\theta}^*$ if and only if f has a local minimum at $\theta^* = g(\tilde{\theta}^*)$*

Proof. First we prove the statement: f has a local minimum at $\theta^* = g(\tilde{\theta}^*) \Rightarrow \tilde{f}$ has a local minimum at $\tilde{\theta}^*$. From the definition of a local minimum, there exists a neighbourhood of θ^* , $\mathcal{V}$ such that

$$f(\theta^*) \leq f(\theta) \quad \forall \theta \in \mathcal{V}. \quad (\text{C.1})$$

Because g is a continuous map, the preimage of $\mathcal{V}$, $\mathcal{U} = g^{-1}(\mathcal{V})$, is an open set which by construction contains $\tilde{\theta}^*$, and is therefore a neighbourhood of $\tilde{\theta}^*$. The result then

follows

$$\begin{aligned}
 f(\theta^*) &\leq f(\theta) \quad \forall \theta \in \mathcal{V} \\
 \Rightarrow f(g(\tilde{\theta}^*)) &\leq f(g(\tilde{\theta})) \quad \forall \tilde{\theta} \in \mathcal{U} \\
 \Rightarrow \tilde{f}(\tilde{\theta}^*) &\leq \tilde{f}(\tilde{\theta}) \quad \forall \tilde{\theta} \in \mathcal{U}.
 \end{aligned} \tag{C.2}$$

Finally, we prove the statement: $\tilde{f}$ has a local minimum at $\tilde{\theta}^* \Rightarrow f$ has a local minimum at $\theta^* = g(\tilde{\theta}^*)$. From the definition of a local minimum, there exists a neighbourhood of $\tilde{\theta}^*$, $\mathcal{U}$ such that

$$\tilde{f}(\tilde{\theta}^*) \leq \tilde{f}(\tilde{\theta}) \quad \forall \tilde{\theta} \in \mathcal{U}. \tag{C.3}$$

By assumption g is a submersion and therefore a continuous open map. Because $\mathcal{U}$ is an open set containing $\tilde{\theta}^*$, $\mathcal{V} = g(\mathcal{U})$ must also be an open set containing θ^* . Finally, then

$$\begin{aligned}
 \tilde{f}(\tilde{\theta}^*) &\leq \tilde{f}(\tilde{\theta}) \quad \forall \tilde{\theta} \in \mathcal{U} \\
 \Rightarrow f(g(\tilde{\theta}^*)) &\leq f(g(\tilde{\theta})) \quad \forall \tilde{\theta} \in \mathcal{U} \\
 \Rightarrow f(\theta^*) &\leq f(\theta) \quad \forall \theta \in \mathcal{V}
 \end{aligned} \tag{C.4}$$

□

Proposition 7. *For any local minimiser θ^* of f , $\exists \tilde{\theta}^* \in \tilde{\Theta}$ that is a local minimiser of $\tilde{f}$ s.t. $\theta^* = g(\tilde{\theta}^*)$*

Proof. $g(\tilde{\Theta}) = \Theta \Rightarrow \exists \tilde{\theta}^* \in \tilde{\Theta}$ s.t. $\theta^* = g(\tilde{\theta}^*)$. The result then follows from Proposition 6. □

Proposition 8. *$\tilde{f}$ has a local maximum at $\tilde{\theta}$ if and only if f has a local maximum at $\theta = g(\tilde{\theta})$*

Proof. This follows from Proposition 6 by symmetric arguments. □

Proposition 9. *$\tilde{f}$ has a saddle point at $\tilde{\theta}$ if and only if f has a saddle point at $\theta = g(\tilde{\theta})$*

Proof. First, we prove the following statement: $\tilde{f}$ has a critical point at $\tilde{\theta} \Leftrightarrow f$ has a critical point at $\theta = g(\tilde{\theta})$. Let $\tilde{\theta}$ and $\theta = g(\tilde{\theta})$ be represented as co-ordinates for some charts at those points, with $f, g, \tilde{f}$ defined similarly. Then the statement about critical points can be expressed as follows

$$\nabla \tilde{f}(\tilde{\theta}) = \mathbf{0} \Leftrightarrow \nabla f(g(\tilde{\theta})) = \mathbf{0} \tag{C.5}$$

where $\mathbf{0}$ is a vector of zeros. To show that $\nabla \tilde{f}(\tilde{\theta}) = \mathbf{0} \Rightarrow \nabla f(g(\tilde{\theta})) = \mathbf{0}$, we have

$$\nabla \tilde{f}(\tilde{\theta}) = \nabla g(\tilde{\theta}) \nabla f(g(\tilde{\theta})) \quad (\text{C.6})$$

where $\nabla g(\tilde{\theta})$ is the transposed Jacobian of g (see Appendix A for an explanation of this notation) and has full column rank due to g being a submersion, and so:

$$\nabla g(\tilde{\theta}) \nabla f(g(\tilde{\theta})) = \mathbf{0} \Rightarrow \nabla f(g(\tilde{\theta})) = \mathbf{0}. \quad (\text{C.7})$$

For the other direction, $\nabla f(g(\tilde{\theta})) = \mathbf{0} \Rightarrow \nabla \tilde{f}(\tilde{\theta}) = \mathbf{0}$, trivially,

$$\begin{aligned} \nabla \tilde{f}(\tilde{\theta}) &= \nabla g(\tilde{\theta}) \nabla f(g(\tilde{\theta})) \\ &= \nabla g(\tilde{\theta}) \mathbf{0} \\ &= \mathbf{0}. \end{aligned} \quad (\text{C.8})$$

Having proven correspondence of critical points, we can proceed to prove the proposition statement in two parts. First, if $\tilde{f}$ has a saddle point at $\tilde{\theta}$, then f must have a critical point at $\theta = g(\tilde{\theta})$. However, from Propositions 6 and 8, we know that this cannot be a local minimum or local maximum, and hence must be a saddle point. Second, in the other direction, if f has a saddle point at $\theta = g(\tilde{\theta})$, then $\tilde{f}$ has a critical point at $\tilde{\theta}$, which by similar reasoning must also be a saddle point. $\square$

Proposition 10. *If $\tilde{f}$ has a non-strict saddle point at $\tilde{\theta}$ then f has a non-strict saddle point at $\theta = g(\tilde{\theta})$*

Proof. From Proposition 9, we know that if $\tilde{f}$ has a non-strict saddle point at $\tilde{\theta}$, then f must have a saddle point at $\theta = g(\tilde{\theta})$. It remains to be proven that the saddle point at θ must be a non-strict saddle point. We do this by contradiction. Let us assume that the saddle point of f at $\theta = g(\tilde{\theta})$ is strict.

Let $\tilde{\theta}, \theta$ be represented as co-ordinates for some charts at those points, with $f, g, \tilde{f}$ defined similarly. Furthermore, let $H_f = \nabla^2 \tilde{f}(\tilde{\theta})$, $H_{\tilde{f}} = \nabla^2 f(g(\tilde{\theta}))$ be the Hessians of f and $\tilde{f}$, at $\tilde{\theta}$ and $\theta = g(\tilde{\theta})$, respectively. Then,

$$\begin{aligned} H_{\tilde{f}} &= \nabla g(\tilde{\theta}) H_f \nabla g(\tilde{\theta})^\top + \sum_{k=1}^j [\nabla f(g(\tilde{\theta}))]_k \nabla^2 g_k(\tilde{\theta}) \\ &= \nabla g(\tilde{\theta}) H_f \nabla g(\tilde{\theta})^\top \end{aligned} \quad (\text{C.9})$$

where $\nabla^2 g_k(\tilde{\theta})$ is the Hessian of the k -th component of g . The second equality follows from θ being a critical point of f .

A strict saddle point is a saddle point for which there is at least one direction of strictly negative curvature, and so given the assumption that $\theta = g(\tilde{\theta})$ is a strict saddle point of f , $\exists v \in \mathbb{R}^i$ such that $v^\top H_f v < 0$. Let $\tilde{v} = A^\top v$, where A is any left inverse of $\nabla g(\tilde{\theta})$, then

$$\begin{aligned}\tilde{v}^\top H_{\tilde{f}} \tilde{v} &= (A^\top v)^\top \nabla g(\tilde{\theta}) H_f \nabla g(\tilde{\theta})^\top (A^\top v) \\ &= v^\top H_f v\end{aligned}\tag{C.10}$$

$$< 0,\tag{C.11}$$

implying that $\tilde{f}(\tilde{\theta})$ is a strict saddle point, a contradiction, and so we conclude that the saddle point at $f(\theta)$ cannot be strict. $\square$

C.2 SNGD pseudocode details

In Algorithm 1, restated with line numbers below, we provide pseudocode for an implementation of SNGD when $\tilde{q}$ is an EF distribution, and $\tilde{\theta}$ are either natural or mean parameters of that family. We assume the existence of an autodiff operator **grad**, which takes as input a real-valued function, and returns another function for computing its gradient. Note that when f cannot be computed deterministically, such as in VI or minibatch settings, we assume **grad** returns a function that provides unbiased stochastic estimates of the gradient. In our VI experiments, where gradients had to be taken through samples, we used the reparameterisation trick (Kingma and Ba, 2014), applied to the target distribution.

We also assume the existence of an overloaded function **dualparams**, which converts from mean to natural parameters of the EF, or vice-versa, depending on the type of its argument.¹ That is, in the notation of Section 5.2.2, **dualparams** resolves to either $\mu(\cdot)$ or $\eta(\cdot)$ as appropriate. We note that each **dualparams** pair only needs to be defined once for each EF, and is not dependent on e.g. the target distribution or loss function, meaning that if these are supplied as part of a software library, the end user is only required to supply f , g and $\tilde{\theta}_0$. Algorithm 1 also assumes a given step size schedule, but can easily be extended to incorporate line search or other methods for choosing ϵ_t .

On line 1 of Algorithm 1 we define a reparameterisation of $\tilde{f}$ in terms of the dual parameters. That is, if $\tilde{\theta}$ are natural parameters, $\tilde{f}_{\text{dual}}$ is a function of mean parameters, and vice-versa. Line 4 computes the natural gradient, given by equation (3.10) or

¹This is purely for convenience, as it allow us to describe a single implementation handling both parameterisations.

Algorithm 1 SNGD with EF surrogate

Require: objective $f : \Theta \rightarrow \mathbb{R}$
Require: parameter mapping $g : \tilde{\Theta} \rightarrow \Theta$
Require: initial surrogate parameters $\tilde{\theta}_0 \in \tilde{\Theta}$
Require: step size schedule $\{\epsilon_t \in \mathbb{R}_+ : t = 0, 1, \dots\}$

```

1:  $\tilde{f}_{\text{dual}}(.) := f(g(\text{dualparams}(.)))$ 
2:  $t \leftarrow 0$ 
3: while not converged do
4:    $\tilde{\nabla} \leftarrow \text{grad}[\tilde{f}_{\text{dual}}](\text{dualparams}(\tilde{\theta}_t))$ 
5:    $\tilde{\theta}_{t+1} \leftarrow \tilde{\theta}_t - \epsilon_t \tilde{\nabla}$ 
6:    $t \leftarrow t + 1$ 
7: return  $g(\tilde{\theta}_t)$ 

```

(3.11), using automatic differentiation of the function $\tilde{f}_{\text{dual}}$. Note that *both* overloads of `dualparams` are called: one inside the auto-differentiated function $\tilde{f}_{\text{dual}}$, and the other outside (to compute its argument). It is often possible for the inner conversion to be elided; for example, the user can supply a function g_{dual} (instead of g) that can perform the map from dual parameters to θ directly, more efficiently than the composition $g \circ \text{dualparams}$. For example, this is often the case when f depends on covariance-like parameters, and the composition $g \circ \text{dualparams}$ would otherwise involve inverting a matrix twice (a no-op).

If $\tilde{\theta}$ are mean parameters of $\tilde{q}$, then the computational overhead of Algorithm 1 (relative to GD in f) is approximately $\text{cost}[\eta] + 3 \times \text{cost}[g \circ \mu]$, where `cost` returns the cost of its function argument. Similarly, if $\tilde{\theta}$ are natural parameters, then the overhead will be approximately $\text{cost}[\mu] + 3 \times \text{cost}[g \circ \eta]$. The factor of 3 in the second term in each case results from taking gradients through $g \circ \text{dualparams}$; however, as discussed above, it is often the case the composition is almost zero cost, in which case this term can be largely eliminated by implementing g_{dual} directly.

We have assumed here that we can take gradients through either $\mu(\cdot)$ or $\eta(\cdot)$ efficiently, depending on the choice of parameterisation. For $\mu(\cdot)$ this is true by assumption for tractable families. For some families the ‘reverse’ map $\eta(\cdot)$, is not available in closed form, but can be efficiently computed using an iterative optimisation procedure (Minka, 2000, 2002). In such cases, we can use implicit differentiation techniques to efficiently compute gradients (Christianson, 1994, Blondel et al., 2022).

C.3 Evaluation details

In this appendix we provide additional details about the experiments presented in Section 5.4. The code for these experiments is available at <https://github.com/cambridge-mlg/sngd>.

We repeated all experiments with 10 different random seeds. In all cases this led to different random parameter initialisations. For VI experiments, this also seeded randomness in the Monte Carlo samples, and for experiments with minibatching, it also seeded randomness in the minibatch sampling. The mean and standard errors as displayed in the training curves and Pareto frontier plots were computed over the 10 runs. For VI experiments we used the reparameterisation trick to estimate gradients for each method (Kingma and Welling, 2014).

In experiments that were small scale, and had objectives that could be computed deterministically, namely MLE experiments using the sheep and COVID datasets, we used exact line search to determine step sizes for each of the methods being tested. This allowed us to compare the search direction of each method without confounding results with choice of hyperparameter settings.

For all other experiments we chose hyperparameters using a grid search. The training curves in Section 5.4 correspond to the “best” hyperparameter settings for each method. The best setting was considered to be that which had the best average (across time steps) worst case (over random seeds) value of the evaluation metric (negative elbo or negative log-likelihood as appropriate). Although this choice is somewhat arbitrary, we found that it consistently chose settings with training curves that closely resembled those that we considered best for each method. In Appendix C.5 we provide Pareto frontier plots which incorporate *all* of the hyperparameter settings tried, which qualitatively are very similar to the training curves in Section 5.4.

For SNGD we chose $\tilde{\theta}$ to be mean parameters of $\tilde{q}$ for all MLE tasks, and natural parameters for all VI tasks. These choices were motivated by the results of Appendix B.4, and we found them to consistently perform better than alternatives.

All experiments were executed on a 76-core Dell PowerEdge C6520 server, with 256GiB RAM, and dual Intel Xeon Platinum 8368Q (Ice Lake) 2.60GHz processors. Each individual optimisation run was locked to a single dedicated core. Implementations were written in JAX (Bradbury et al., 2018).

Next we provide details specific to each task featured in the experiments. In the list below, n denotes the number of training observations, and d denotes the dimensionality of the distribution that is being optimised in the task.

Sheep ($n=82$, $d=1$) Taken from a seminal work on the negative binomial distribution by Fisher (1941), this dataset consists of the number of ticks observed on each member of a population of sheep. The task for this dataset was MLE of the negative binomial distribution.

UCI miniboone ($n=32,840$, $d=43$) Taken from the MiniBooNE experiment at Fermilab, this dataset consists of a number of readings that can be used to classify observations as either electron or muon neutrinos (Roe, 2010). We follow the pre-processing of Papamakarios et al. (2017). Using this dataset we performed MLE of skew-normal and skew- t distributions. We used 32,840 of the observations for training, and the remaining 3,648 observations for evaluation. We used a minibatch size of 256 for each method.

UCI covertype ($n=500$, $d=53$) This dataset classifies the forest cover type of 581,024 pixels, based on 53 cartographic variables (Blackard, 1998). We used the ‘binary scale’ preprocessing of Chang and Lin (2011). The task was to perform VI in a Bayesian logistic regression model, with regularisation parameter 1.0. We found that using anything close to the full number of observations resulted in degenerate posteriors with virtually zero uncertainty, obviating the need for variational inference, and so we used a randomly chosen subset of 500 observations for our experiments. We used skew-normal, skew- t and skew-normal mixture distributions as approximate posteriors. All methods used 20 Monte Carlo samples for training and 1000 for evaluation.

Synthetic skew-normal ($n=10,000$, $d=1,000$) We generated synthetic data from a d -dimensional skew-normal distribution with parameters (ξ, Ω, η) , where ξ and η were drawn from $\mathcal{N}(\mathbf{0}, I_d)$. We chose $\Omega = d^{-1}W^\top W + 10^{-4}I_d$, where the components of $W \in \mathbb{R}^{d \times d}$ were drawn from independent standard normal distributions. The task was to perform MLE of a skew-normal distribution on this dataset.

Synthetic skew- t ($n=10,000$, $d=1,000$) We generated synthetic data from a d -dimensional skew- t distribution with parameters (ξ, Ω, η, ν) . ξ , Ω and η were generated in the same manner as the synthetic skew-normal task as detailed above, with ν chosen to be 10. The task was to perform MLE of a skew- t distribution on this dataset.

FTSE 100 stock returns ($n=1,515$, $d=93$) This dataset consists of daily stock price returns from 2017/01/01 to 2022/12/31 for the subset of FTSE 100 stocks that were members of the index during the entire period. We first fitted univariate Student’s

t distributions to each dimension independently, and then transformed the data by converting each observed variable to its quantile value under the marginal distribution. The task was then MLE of a t copula on the quantile values.

COVID hospital admissions ($n=1,120$, $d=1$) This dataset consists of the number of daily COVID hospital admissions in the UK from 2020/4/1 to 2023/5/1. The task for this dataset was MLE of a 5-component negative binomial mixture.

Synthetic 2D logistic regression ($n=30$, $d=2$) This synthetic logistic regression dataset was generated using the same procedure as Murphy (2012). The task was VI in a Bayesian logistic regression model, with regularisation parameter 1. We used a skew-normal mixture approximate posterior, with 20 Monte Carlo samples for training and 1,000 for evaluation.

Now we provide details particular to the distributions being optimised in the tasks above. Note that the correspondence between tasks (above) and target distributions (below) is many to many: some target distributions were applied to more than one task, and some tasks were used for several target distributions. The parameter mappings used for SNGD are given in Table C.1 in Appendix C.4 – we do not repeat them here unless additional explanation is required. When the surrogate parameters are partitioned as $\tilde{\theta} = (\tilde{\theta}_{\text{fg}}, \tilde{\theta}_{\text{bg}})$, as described in Section 5.3.3, we used Adam to optimise $\tilde{\theta}_{\text{fg}}$. With our baseline methods, when a target distribution required a positive definite covariance matrix parameter, we tried two different parameterisations: covariance square-root (e.g. $\Sigma = W^\top W$), and precision square-root (e.g. $\Sigma^{-1} = W^\top W$) (Salimbeni et al., 2018). This parameterisation choice was determined by a hyperparameter which we included in our grid search. Further parameterisation details are given below.

Negative binomial The PMF of the negative binomial with parameters $\theta = (r, s)$ is given by (5.17). We initialised negative binomial parameters for all methods by drawing $s \sim \text{Uniform}(0.05, 0.95)$, $r \sim \text{Gamma}(6.25, 1.25)$. With GD, BFGS and NGD we used a log parameterisation of r , and a logit parameterisation of s . For SNGD we chose $\tilde{\theta}$ to be mean parameters of $\tilde{q}$. In order to ensure that $g(\tilde{\theta}) \in \Theta \forall \tilde{\theta} \in \tilde{\Theta}$, we require $\beta < 1 \forall \tilde{\theta} \in \tilde{\Theta}$, therefore we chose $\tilde{\Theta}$ as the subset of $\mathcal{M}$ (the mean domain of $\tilde{q}$) for which $\beta < 1$. It can be shown that this remains an open convex set.

Skew-normal The PDF of the multivariate skew-normal with parameters $\theta = (\xi, \Omega, \eta)$ is given by $q_\theta(x) = 2\mathcal{N}_d(x; \xi, \Omega)\Phi(\eta^\top(x - \xi))$, where $\mathcal{N}_d(\cdot)$ is the PDF of

the d -dimensional normal distribution, and $\Phi(\cdot)$ is the CDF of the standard normal distribution.² We used random initialisations of $\xi \sim \mathcal{N}(\mathbf{0}, 0.01^2 I_d)$, $\eta \sim \mathcal{N}(\mathbf{0}, 0.01^2 I_d)$, with Ω initialised to I_d .

Skew- t The PDF of the multivariate skew- t with parameters $\theta = (\xi, \Omega, \eta, \nu)$ is given by $q_\theta(x) = 2\mathcal{T}_d(x; \xi, \Omega, \nu)\Psi(\alpha^\top x \sqrt{(\nu + d)/(\nu + x^\top \bar{\Omega}^{-1}x)}; \nu + d)$, where $\mathcal{T}_d$ is the PDF of the d -dimensional Student's t , $\Psi(\cdot)$ is the CDF of the (univariate) Student's t , $\omega = (\Omega \odot I_d)^{1/2}$, $\bar{\Omega} = \omega^{-1}\Omega\omega^{-1}$, and $\alpha = \omega\eta$.³ We used random initialisations of $\xi \sim \mathcal{N}(\mathbf{0}, 0.01^2 I_d)$, $\eta \sim \mathcal{N}(\mathbf{0}, 0.01^2 I_d)$, with Ω and ν initialised to I_d and 50 respectively. For all methods we parameterised ν as $\log(\nu - 2)$.

t copula The d -dimensional t copula with parameters $\theta = (R, \nu)$ has PDF $q_\theta(x) = \mathcal{T}_d(z; 0, R) / \prod_i \mathcal{T}(z_i; 0, 1)$, where $\mathcal{T}$ is the PDF of the univariate Student's t . We initialised R to $\text{corr}(I + W^\top W)$, and ν to 50, where $W_{ij} \sim \mathcal{N}(0, .01^2)$, and $\text{corr}(\Sigma) = (\Sigma \odot I_d)^{-\frac{1}{2}}\Sigma(\Sigma \odot I_d)^{-\frac{1}{2}}$ projects a covariance matrix to its implied correlation matrix. For all methods we parameterised ν as $\log(\nu - 2)$.

Negative binomial mixture The negative binomial mixture with k components and parameters $\theta = (\pi_i, r_i, s_i)_{i=1}^k$ has PMF given by (B.26), where $q_{\theta_i}(x)$ is the negative binomial PMF given by (5.17) with parameters $\theta_i = (r_i, s_i)$. We initialised π_i to k^{-1} for $i = 1, \dots, k$, and the component negative binomials were initialised to have mean m_i and variance v_i , with $m_i \sim \text{Uniform}(0.1x_{\max}, 0.9x_{\max})$ and $v_i = m_i o_i$ where $o_i^{-1} \sim \text{Uniform}(0.001, 0.02)$ and $x_{\max}$ is the maximum value observed in the training data. With GD and BFGS, we used a softmax parameterisation of the mixture probabilities, a log parameterisation of r_i , and a logit parameterisation of s_i .

Skew-normal mixture The skew-normal mixture with k components and parameters $\theta = (\pi_i, \xi_i, \Omega_i, \nu_i)_{i=1}^k$ has PDF given by (B.26), where $q_{\theta_i}(x)$ is the skew-normal PDF with parameters $\theta_i = (\xi_i, \Omega_i, \nu_i)$. We initialised π_i to k^{-1} for $i = 1, \dots, k$, with the remaining parameters initialised equivalently to the (non-mixture) skew-normal case. With Adam, we used a softmax parameterisation of the mixture probabilities.

²Equation (5.1) of Azzalini (2013).

³Equation (6.24) of Azzalini (2013).

C.4 Parameter mappings

In Table C.1 we provide parameter mappings for all of the examples appearing in Chapter 5. For convenience, we express the mappings in terms of the standard parameterisation of $\tilde{q}$. For the algorithms described in Section 5.3.3, $\tilde{\theta}$ would be the mean or natural parameters corresponding to the parameters stated here, with g similarly adjusted. Note that in Table C.1 we follow convention by using μ to denote the mean of a normal distribution, whereas in the body of the thesis it refers to the mean *parameters* (expected sufficient statistics) of an EF distribution. See Appendix C.3 for target distribution definitions.

C.5 Additional results

In this appendix we provide additional results for the experiments presented in Section 5.4. In several of the experiments (those not using line-search), the competing methods were dependent on hyperparameters which were chosen by grid search. For the training curves in Section 5.4, we used the heuristic method described in Appendix C.3 in order to choose the “best” settings for each method.

In order to provide a more complete comparison, in Figures C.1 through C.8 we present Pareto frontier plots that show the best (average) value of the chosen metric (y axis) attained by *any* learning rate setting for each method, as a function of both iteration count and wall-clock time (x axis). The displayed error bars correspond to 2 standard errors, computed for the optimal setting at the corresponding point in time.

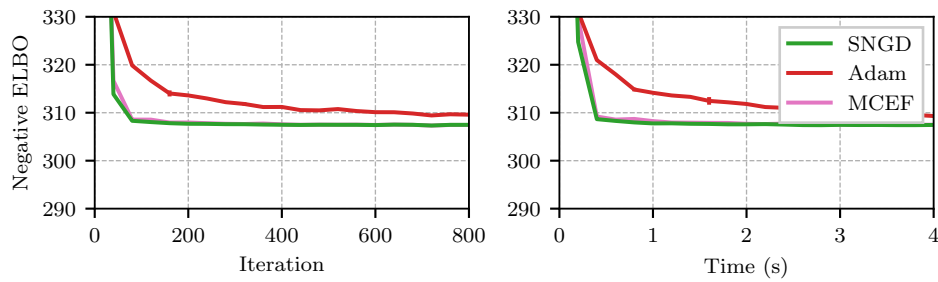


Fig. C.1 Pareto frontiers for Bayesian logistic regression VI on UCI covertime ($n=500$, $d=53$), with a skew-normal approximation.

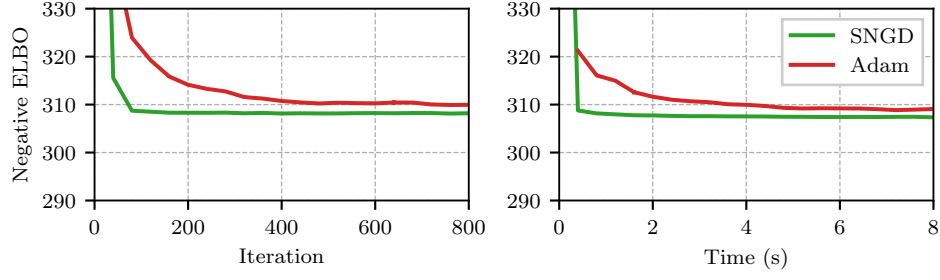


Fig. C.2 Pareto frontiers for Bayesian logistic regression VI on UCI coverytype ($n=500$, $d=53$), with a skew- t approximation.

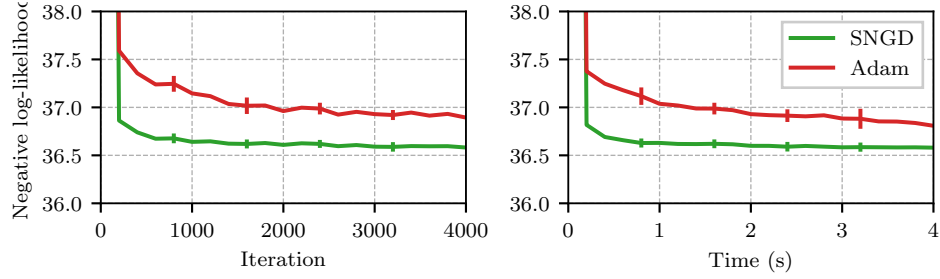


Fig. C.3 Pareto frontiers for skew-normal MLE on UCI miniboone ($n=32,840$, $d=43$).

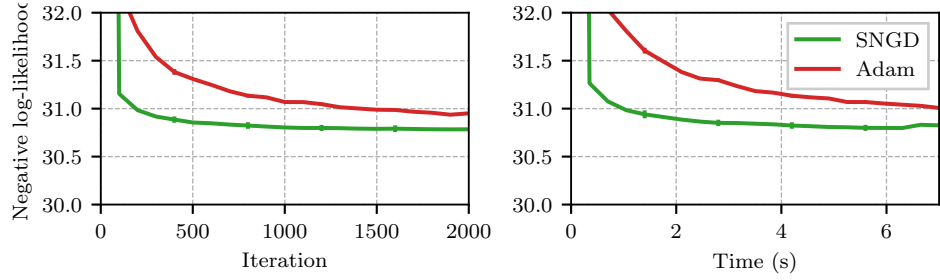


Fig. C.4 Pareto frontiers for skew- t MLE on UCI miniboone ($n=32,840$, $d=43$).

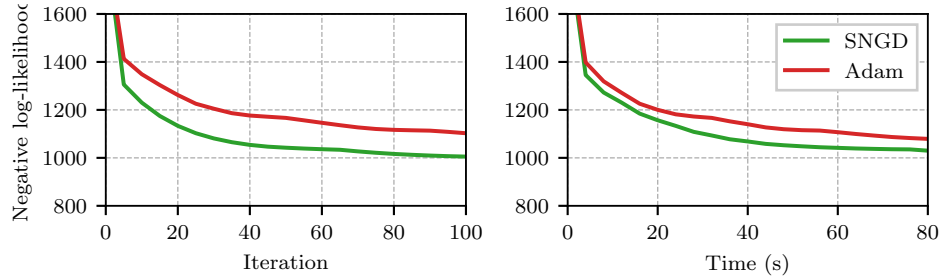


Fig. C.5 Pareto frontiers for skew-normal MLE on synthetic dataset ($n=10,000$, $d=1,000$).

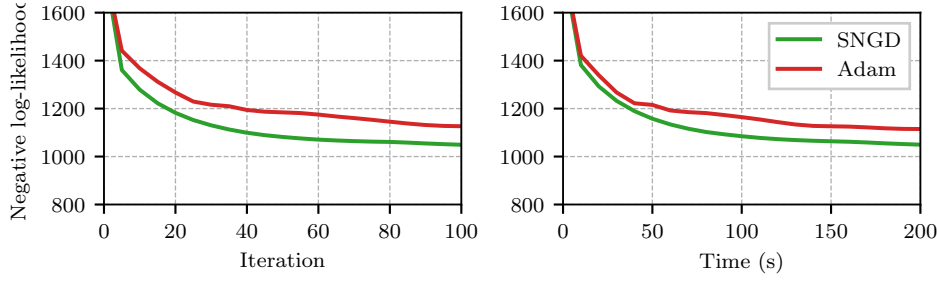


Fig. C.6 Pareto frontiers for skew- t MLE on synthetic dataset ($n=10,000$, $d=1,000$).

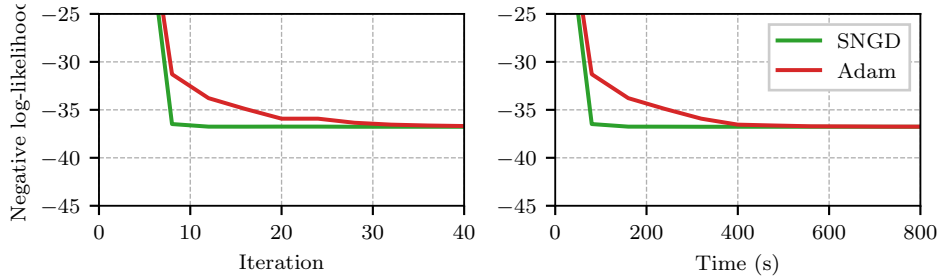


Fig. C.7 Pareto frontiers for t -copula MLE on 5 years of stock return data ($n=1,515$, $d=93$).

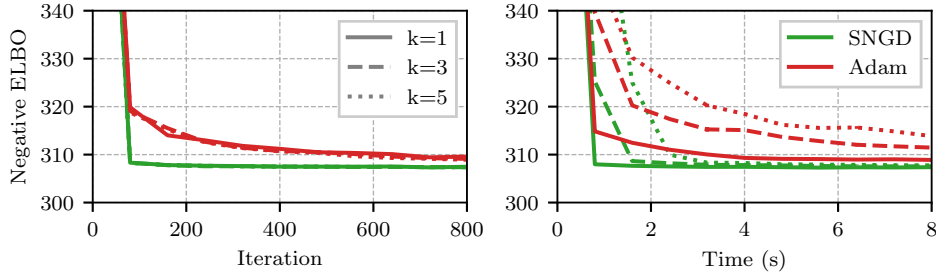


Fig. C.8 Pareto frontiers for Bayesian logistic regression VI on UCI covtype ($n=500$, $d=53$), with a skew-normal mixture approximation.

C.6 Comparison with NGD under q

In Section 5.3.1 we provided a motivation for SNGD that viewed it as approximating NGD under a reparameterisation of q . However, for the negative binomial MLE experiment of Section 5.4.1, we demonstrated that SNGD was actually able to *outperform* NGD under q . In Figure C.9, we show that this was true when NGD was performed with respect to both a standard parameterisation (also shown in Figure 5.2), and the reparameterisation defined by $g(\tilde{\theta})$. This implies that the outperformance of SNGD is

⁴ $\text{corr}(\cdot)$, defined in Appendix C.3, maps a covariance matrix to its implied correlation matrix.

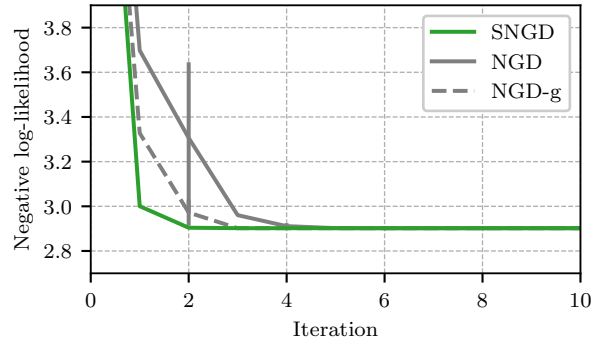


Fig. C.9 Training curves for negative binomial MLE on the sheep dataset ($n=82$, $d=1$), using surrogate natural gradient descent (SNGD) and natural gradient descent under q with respect to both a standard parameterisation (NGD), and the reparameterisation defined by g (NGD-g).

not simply an artefact of the reparameterisation. This surprising result deserves extra scrutiny.

We begin by highlighting that in fact, SNGD *is* NGD, but in the objective $\tilde{f}(\tilde{\theta}) = f(g(\tilde{\theta}))$ and with respect to distribution $\tilde{q}$. It is helpful therefore to consider the desirable properties of NGD:

1. It is locally invariant to parameterisation of the distribution being optimised.
2. It follows the direction of steepest descent in the objective on the statistical manifold of the distribution being optimised.⁵
3. For MLE objectives, it asymptotically approaches Newton's method near the optimum.

Properties 1 and 2 apply always, and so they are necessarily inherited by SNGD. However, when f is a MLE objective for q (as in the experiment of Section 5.4.1), $\tilde{f}$ is *not*, in general, an MLE objective for $\tilde{q}$, and so SNGD loses property 3. Contrast this with NGD in $\tilde{f}$ with respect to q – although the objective function is the same as that of SNGD, it is simply a reparameterised MLE objective for q , and so property 3 is retained. In other words, the asymptotic efficiency of NGD is retained under change of parameterisation, but not under change of distribution.

On the face of it, then, when f is a MLE objective for q , SNGD appears less desirable than NGD under q , inheriting only two of the properties listed above. However, none of these properties say anything about performance over large (or indeed, any *non-infinitesimal*) steps in parameter space. But there is one special case in which we can

⁵Where steepness is defined with respect to a KL divergence.

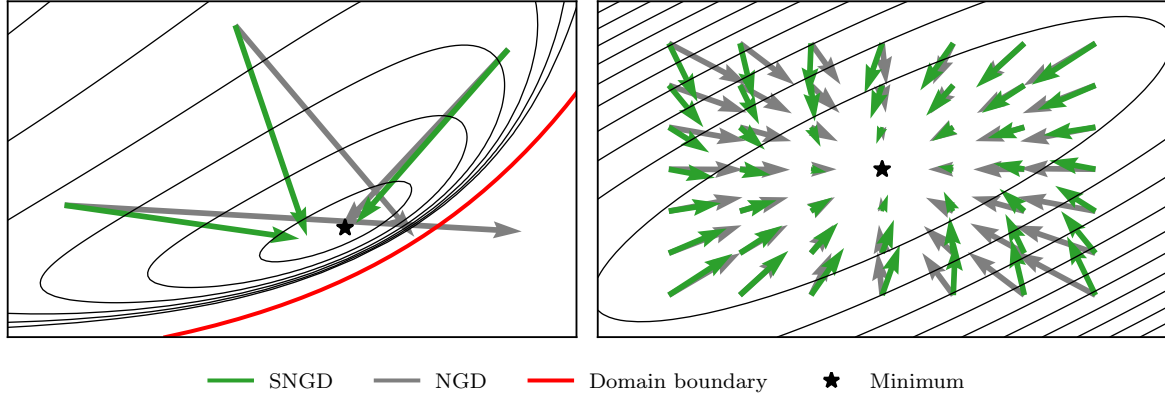


Fig. C.10 Comparison of SNGD and NGD in the sheep experiment of Section 5.4.1. Contours are of $\tilde{f}(\tilde{\theta})$, an MLE objective for q . (left) Vectors represent a full (undamped) step under each method. When far from the optimum, a full step of SNGD more consistently moves to a low value of the objective, compared to that of NGD under q . (right) The optimum magnified $\times 1000$. At this scale the loss is well approximated by a quadratic. The Fisher of q , $F(\tilde{\theta})$, approaches the Hessian. This is not true of $\tilde{F}(\tilde{\theta})$, because the loss is not a MLE objective for $\tilde{q}$. Here the gradient vector lengths have been scaled down by a factor of 3 for clarity.

say something about the performance of NGD over large steps in parameter space. When NGD is applied to a MLE objective of an EF distribution with respect to the mean parameters of that distribution, a single undamped step will converge straight to the optimum (see Proposition 1 in Section 3.4).

When we apply SNGD with EF $\tilde{q}$, and with respect to mean parameters $\tilde{\theta}$, we meet 2 of the criteria for single-step convergence. Although $\tilde{f}(\tilde{\theta})$ is not in general a MLE for $\tilde{q}$, if it behaves *approximately* like a MLE objective for $\tilde{q}$, then SNGD can make rapid progress over large distances in parameter space, and *approximately* converge in a single step. And so, while SNGD loses the asymptotic efficiency of NGD under q , it is possible to obtain improved performance in the early to mid stages. This is exactly what we observe in the experiment of Section 5.4.1, as is further illustrated in Figure C.10. The ability to make rapid progress early on is arguably more important for the overall performance of an optimiser in practice. However, if convergence to an exact (within machine precision range) optimum is required, it may be beneficial to switch to Newton-like methods during the late phase, an approach previously proposed by Tatzel et al. (2022).

We now attempt to characterise what it means for $\tilde{f}$ to behave *approximately* like a MLE for $\tilde{q}$. Let $\tilde{q}$ be an EF with mean parameters $\tilde{\theta}$. Furthermore, let $f(\tilde{\theta})$ be an

MLE objective for q so that

$$\begin{aligned}\tilde{f}(\tilde{\theta}) &= f(g(\tilde{\theta})) \\ &= -\mathbb{E}_{p_{\mathcal{D}}(x)} \left[\log q_{g(\tilde{\theta})}(x) \right]\end{aligned}\tag{C.12}$$

where $p_{\mathcal{D}}(x) = \frac{1}{n} \sum_{i=1}^n \delta(x - x_i)$ is the empirical density function. Using a trivial identity, we can rewrite this as

$$\begin{aligned}\tilde{f}(\tilde{\theta}) &= \text{KL}[\tilde{q}_{\tilde{\theta}^*} \parallel \tilde{q}_{\tilde{\theta}}] - \mathbb{E}_{p_{\mathcal{D}}(x)} \left[\log q_{g(\tilde{\theta})}(x) \right] - \text{KL}[\tilde{q}_{\tilde{\theta}^*} \parallel \tilde{q}_{\tilde{\theta}}] \\ &= \tilde{f}^*(\tilde{\theta}) + h(\tilde{\theta})\end{aligned}\tag{C.13}$$

where $\tilde{\theta}^*$ are optimal parameters of $\tilde{f}$, and we have defined

$$\tilde{f}^*(\tilde{\theta}) = \text{KL}[\tilde{q}_{\tilde{\theta}^*} \parallel \tilde{q}_{\tilde{\theta}}],\tag{C.14}$$

and

$$h(\tilde{\theta}) = -\mathbb{E}_{p_{\mathcal{D}}(x)} \left[\log q_{g(\tilde{\theta})}(x) \right] - \text{KL}[\tilde{q}_{\tilde{\theta}^*} \parallel \tilde{q}_{\tilde{\theta}}].\tag{C.15}$$

$\tilde{f}^*(\tilde{\theta})$ can be minimised by a single NGD step with respect to $\tilde{q}$ and $\tilde{\theta}$ (see Proposition 1). The second term, $h(\tilde{\theta})$, can then be seen as distorting the natural gradient step, acting to move it away from the optimum $\tilde{\theta}^*$. If $h(\tilde{\theta})$ is (approximately) constant with respect to $\tilde{\theta}$, then SNGD will (approximately) converge in a single step. In Figure C.11 we show that in the sheep experiment of Section 5.4.1, the effect of the distortion term is indeed small, even when far from the optimum, illustrating why SNGD is able to make rapid progress on this problem.

To add some intuition behind $h(\tilde{\theta})$, let us assume that the data were generated from the model distribution q with parameters θ^* . Furthermore, take the infinite data limit,⁶ so that

$$\tilde{f}(\tilde{\theta}) = -\mathbb{E}_{q_{\theta^*}(x)} \left[\log q_{g(\tilde{\theta})}(x) \right]\tag{C.16}$$

and $\theta^* = g(\tilde{\theta}^*)$. The distortion term $h(\tilde{\theta})$ can then be written as

$$\begin{aligned}h(\tilde{\theta}) &= -\mathbb{E}_{q_{\theta^*}(x)} \left[\log q_{g(\tilde{\theta})}(x) \right] - \text{KL}[\tilde{q}_{\tilde{\theta}^*} \parallel \tilde{q}_{\tilde{\theta}}] \\ &= \text{KL}[q_{g(\tilde{\theta}^*)} \parallel q_{g(\tilde{\theta})}] - \text{KL}[\tilde{q}_{\tilde{\theta}^*} \parallel \tilde{q}_{\tilde{\theta}}] + c\end{aligned}\tag{C.17}$$

⁶We do this for convenience; for finite samples, the resulting expressions hold in expectation (over random data sets).

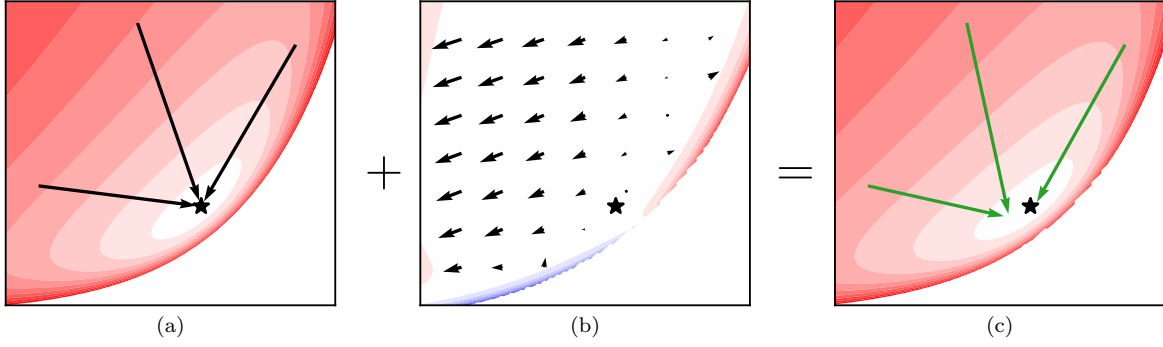


Fig. C.11 Decomposition of the objective function and SNGD step. (a) Contours of $\tilde{f}^*(\tilde{\theta})$, vectors show $\tilde{\nabla}\tilde{f}^*(\tilde{\theta})$ at 3 distinct locations, where $\tilde{\nabla}$ denotes the natural gradient with respect to mean parameters of $\tilde{q}$ – the natural gradient always move straight to the optimum of $\tilde{f}$. (b) Contours of the distortion term $h(\tilde{\theta})$, vectors show $\tilde{\nabla}h(\tilde{\theta})$ at a regular grid of points. (c) Contours of $\tilde{f}(\tilde{\theta})$, vectors now show the resulting SNGD step, $\tilde{\nabla}\tilde{f}(\tilde{\theta})$ – this is the sum of the previous two vector fields.

where c is constant with respect to $\tilde{\theta}$. For $h(\tilde{\theta})$ to have no effect on the NGD step, we then require

$$\nabla_{\tilde{\theta}}(\text{KL}[\tilde{q}_{\tilde{\theta}^*} \parallel \tilde{q}_{\tilde{\theta}}]) = \nabla_{\tilde{\theta}}(\text{KL}[q_{g(\tilde{\theta}^*)} \parallel q_{g(\tilde{\theta})}]). \quad (\text{C.18})$$

This is similar to the statement expressed by equation (5.9), concerning the effect of $\tilde{\theta}$ on *reverse* KL divergences between nearby points, whereas now we have a statement about the effect of $\tilde{\theta}$ on *forward* KL divergences from the (potentially distant) optimal point.

Table C.1 Example surrogate-target parameter mappings

Target	Surrogate	θ	$\tilde{\theta}_{\text{ng}}$	$\tilde{\theta}_{\text{fg}}$	$g(\tilde{\theta})$
Neg. bin.	Gamma	r, s	α, β		$\alpha/(1 - \beta^{-1}), \beta^{-1}$
Neg. bin. mix.	Gamma mix. model	(π_i, r_i, s_i)	$(\pi_i, \alpha_i, \beta_i)$		$(\pi_i, \alpha_i/(1 - \beta_i^{-1}), \beta_i^{-1})$
Skew-normal	Normal	ξ, Ω, η	μ, Σ	η	μ, Σ, η
Skew-normal mix.	Normal mix. model	$(\pi_i, \xi_i, \Omega_i, \eta_i)$	(π_i, μ_i, Σ_i)	(η_i)	$(\pi_i, \mu_i, \Sigma_i, \eta_i)$
Skew- t	Normal	ξ, Ω, η, ν	μ, Σ	$\eta, \log(\nu - 2)$	μ, Σ, η, ν
Elliptical copula	Zero-mean normal	$R, \dots$	Σ	$\dots$	$\text{corr}(\Sigma), \dots$ ⁴

Appendix D

Appendices for Chapter 6

D.1 Conventional presentation of EP updates

In this appendix we give a more conventional presentation of the EP updates, and show equivalence with the atypical presentation of Section 6.2. We will assume here that $\beta_i = 1 \forall i$, but similar reasoning can be used to show equivalence in the power EP case, where $\beta_i \neq 1$. Note that the update of power EP is often presented as having scale proportional to β_i , with a separate damping parameter. We subsume both into α , with $\alpha = 1$ corresponding to the default damping suggested by Minka (2004). Given a target distribution with density of the form

$$p^*(z) \propto \exp\left(\eta_0^\top s(z)\right) \prod_{i=1}^m \exp\left(\ell_i(z)\right), \quad (\text{D.1})$$

EP attempts to find an approximation $q \in \mathcal{F}$ such that

$$q(z) \propto \exp\left(\eta_0^\top s(z)\right) \prod_i \exp\left(\lambda_i^\top s(z)\right) \approx p^*(z). \quad (\text{D.2})$$

The i -th *site parameter* λ_i can loosely be interpreted as the natural parameters of a $\mathcal{F}$ -approximation to the i -th target factor, $\exp(\ell_i(z))$. EP is also often used to approximate the normalising constant of $p^*(.)$ – we have omitted details here, but see e.g. Bishop (2007).

In order to update λ_i , EP performs a KL-divergence minimisation between $q(.)$ and a *local* approximation to the target distribution density, consisting of just one “real”

factor and $(m - 1)$ approximate ones. Specifically,

$$\lambda_i \leftarrow \arg \min_{\lambda_i} \text{KL}[\bar{q}_i(z) \parallel q(z; \lambda_i)], \quad (\text{D.3})$$

where

$$\bar{q}_i(z) \propto \exp \left(\left(\eta_0 + \sum_{j \neq i} \lambda_j \right)^\top s(z) + \ell_i(z) \right), \quad (\text{D.4})$$

and we have used the notation $q(z; \lambda_i)$ to make the dependence on λ_i explicit. It can be easily shown that the solution to (D.3) is found by moment matching. That is, the KL-divergence is minimised when

$$\mathbb{E}_{q(z; \lambda_i)}[s(z)] = \mathbb{E}_{\bar{q}_i(z)}[s(z)]. \quad (\text{D.5})$$

For a minimal exponential family $\mathcal{F}$, there is a unique member of $\mathcal{F}$ with mean parameters $\mu_i = \mathbb{E}_{\bar{q}_i(z)}[s(z)]$, and its natural parameters are given by $\nabla A^*(\mathbb{E}_{\bar{q}_i(z)}[s(z)])$. By equating this with the natural parameters of q , we have

$$\lambda_i = \nabla A^*(\mu_i) - \eta_0 - \sum_{j \neq i} \lambda_j. \quad (\text{D.6})$$

When (D.6) is used as an assignment to λ_i , we obtain the “standard” update of EP, which can be performed either serially or in parallel (over i). Often a damping factor α is used, giving

$$\lambda_i = (1 - \alpha)\lambda_i + \alpha \left(\nabla A^*(\mu_i) - \eta_0 - \sum_{j \neq i} \lambda_j \right). \quad (\text{D.7})$$

In Section 6.2 we presented EP and several variants as performing some number of “inner” updates, to decrease L with respect to λ_i , with an “outer” update with respect to η . These updates are given by

$$\textbf{Inner update: } \lambda_i \leftarrow \lambda_i - \alpha \left(\eta_0 + \sum_j \lambda_j - \nabla A^*(\mathbb{E}_{q_i(z)}[s(z)]) \right), \quad (\text{D.8})$$

$$\textbf{Outer update: } \eta \leftarrow \eta_0 + \sum_j \lambda_j. \quad (\text{D.9})$$

Immediately after an outer update we have that $\eta = \eta_0 + \sum_j \lambda_j$, and so for $\beta_i = 1$ we have

$$\begin{aligned} q_i(z) &\propto \exp\left((\eta - \lambda_i)^\top s(z) + \ell_i(z)\right) \\ &= \exp\left(\left(\eta_0 + \sum_{j \neq i} \lambda_j\right)^\top s(z) + \ell_i(z)\right) \\ &\propto \bar{q}_i(z). \end{aligned} \tag{D.10}$$

We stated in Section 6.2 that (non-double-loop) EP can be viewed as performing a single inner update per outer update, and so by rolling (D.8) and (D.9) into a single update for λ_i , we have

$$\begin{aligned} \lambda_i &\leftarrow \lambda_i - \alpha \left(\eta_0 + \sum_j \lambda_j - \nabla A^*(\mu_i) \right) \\ &= (1 - \alpha) \lambda_i + \alpha \left(\nabla A^*(\mu_i) - \eta_0 - \sum_{j \neq i} \lambda_j \right), \end{aligned} \tag{D.11}$$

which is identical to (D.7).

D.2 Natural gradient view of EP

Let $\tilde{q}_i^{(t)}$ be the member of $\mathcal{F}$ with natural parameter

$$\eta_i^{(t)}(\lambda_i) = \eta_0 + \sum_j \lambda_j^{(t)} + \alpha^{-1}(\lambda_i - \lambda_i^{(t)}). \tag{D.12}$$

Proposition 3 states that the moment-matching updates of EP can be viewed as performing NGD in L with respect to the mean parameters, μ_i , of $\tilde{q}_i^{(t)}$. We restate the proposition below, and give a proof. Note that the right hand side of (6.9), below, maps the EP update (6.4) onto mean parameters of $\tilde{q}_i^{(t)}$. The left hand side of (6.9) is simply a NGD update in μ_i .

Proposition 3. *For $\alpha > 0$, the moment-matching update of EP (6.4) is equivalent to performing an NGD step in L with respect to the mean parameters of $\tilde{q}_i^{(t)}$ with step size α^{-1} . That is, for $\mu_i = \mathbb{E}_{\tilde{q}_i^{(t)}(z)}[s(z)]$, and $\tilde{F}_i^{(t)}(\mu_i)$ the FIM of $\tilde{q}_i^{(t)}$ with respect to*

μ_i , we have

$$\mu_i - \alpha^{-1} \left[\tilde{F}_i^{(t)}(\mu_i) \right]^{-1} \frac{\partial L}{\partial \mu_i} = \left(\nabla A \circ \eta_i^{(t)} \right) \left(\lambda_i - \alpha \left(\eta_0 + \sum_j \lambda_j - \nabla A^*(\mathbb{E}_{q_i(z)}[s(z)]) \right) \right). \quad (6.9)$$

Proof. Taking the left hand side of (6.9), we have

$$\begin{aligned} \mu_i - \alpha^{-1} \left[\tilde{F}_i^{(t)}(\mu_i) \right]^{-1} \frac{\partial L_i}{\partial \mu_i} &= \mu_i - \alpha^{-1} \left[\tilde{F}_i^{(t)}(\mu_i) \right]^{-1} \frac{\partial \lambda_i}{\partial \mu_i} \frac{\partial L}{\partial \lambda_i} \\ &= \mu_i - \alpha^{-1} \left(\frac{\partial \eta_i}{\partial \mu_i} \right)^{-1} \frac{\partial \eta_i}{\partial \mu_i} \frac{\partial \lambda_i}{\partial \eta_i} \frac{\partial L}{\partial \lambda_i} \\ &= \mu_i - \frac{\partial L}{\partial \lambda_i} \\ &= \mu_i - \left(\nabla A \left(\eta_0 + \sum_j \lambda_j \right) - \mathbb{E}_{q_i(z)}[s(z)] \right) \\ &= \mu_i - \left(\mu_i - \mathbb{E}_{q_i(z)}[s(z)] \right) \\ &= \mathbb{E}_{q_i(z)}[s(z)], \end{aligned} \quad (D.13)$$

where the penultimate equality follows from the definition $\mu_i = \mathbb{E}_{\tilde{q}_i^{(t)}(z)}[s(z)]$, and the fact that $\lambda_j^{(t)} = \lambda_j \ \forall j$ before the update. All that remains is to show that the right hand side of (6.9) is equal to $\mathbb{E}_{q_i(z)}[s(z)]$.

$$\begin{aligned} &\left(\nabla A \circ \eta_i^{(t)} \right) \left(\lambda_i - \alpha \left(\eta_0 + \sum_j \lambda_j - \nabla A^*(\mathbb{E}_{q_i(z)}[s(z)]) \right) \right) \\ &= \nabla A \left(\alpha^{-1} (\lambda_i - \lambda_i^{(t)}) + \nabla A^*(\mathbb{E}_{q_i(z)}[s(z)]) \right) \\ &= \mathbb{E}_{q_i(z)}[s(z)] \end{aligned} \quad (D.14)$$

□

D.3 EP- η update derivation

Using the relation $\eta_i = \eta_i^{(t)}(\lambda_i)$, and therefore $\lambda_i = (\eta_i^{(t)})^{-1}(\eta_i)$, we have

$$\begin{aligned}
-\left[\tilde{F}_i(\eta_i)\right]^{-1} \frac{\partial L}{\partial \eta_i} &= -\left[\tilde{F}_i(\eta_i)\right]^{-1} \frac{\partial \lambda_i}{\partial \eta_i} \frac{\partial L}{\partial \lambda_i} \\
&= -\alpha \left(\frac{\partial \mu_i}{\partial \eta_i} \tilde{F}_i(\mu_i) \frac{\partial \mu_i}{\partial \eta_i} \right)^{-1} \frac{\partial L}{\partial \lambda_i} \\
&= -\alpha \left(\frac{\partial \eta_i}{\partial \mu_i} \tilde{F}_i(\mu_i)^{-1} \frac{\partial \eta_i}{\partial \mu_i} \right) \frac{\partial L}{\partial \lambda_i} \\
&= -\alpha \left(\frac{\partial \eta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \eta_i} \frac{\partial \eta_i}{\partial \mu_i} \right) \frac{\partial L}{\partial \lambda_i} \\
&= -\alpha \frac{\partial \eta_i}{\partial \mu_i} \left(\nabla A(\eta_0 + \sum_j \lambda_j) - \mathbb{E}_{q_i(z)}[s(z)] \right).
\end{aligned} \tag{D.15}$$

D.4 EP- μ update derivation

For $\alpha = 1$, we have

$$\begin{aligned}
\mu_i - \epsilon \left[\tilde{F}_i(\mu_i) \right]^{-1} \frac{\partial L}{\partial \mu_i} &= \mu_i - \epsilon \left[\tilde{F}_i(\mu_i) \right]^{-1} \frac{\partial \eta_i}{\partial \mu_i} \frac{\partial \lambda_i}{\partial \eta_i} \frac{\partial L}{\partial \lambda_i} \\
&= \mu_i - \epsilon \left(\frac{\partial \eta_i}{\partial \mu_i} \right)^{-1} \frac{\partial \eta_i}{\partial \mu_i} \frac{\partial \lambda_i}{\partial \eta_i} \left(\nabla A(\eta_0 + \sum_j \lambda_j) - \mathbb{E}_{q_i(z)}[s(z)] \right) \\
&= \mu_i - \epsilon \left(\nabla A(\eta_0 + \sum_j \lambda_j) - \mathbb{E}_{q_i(z)}[s(z)] \right).
\end{aligned} \tag{D.16}$$

D.5 Bias of EP and EP- μ

Let the bias in dimension d of $\mu_i^{(t+1)}$, after an update at time t , be defined as $\mathbb{E}[\mu_i^{(t+1)} - \bar{\mu}_i^{(t+1)}]_d$, where $\bar{\mu}_i^{(t+1)}$ is the value of $\mu_i^{(t+1)}$ after a noise-free update, and expectation is taken over the sampling distributions of all parallel updates at time t . Propositions 4 and 5, which we restate and prove below, summarise the effect of decreasing α and ϵ on EP and EP- μ , respectively.

Proposition 4. *After update (6.4) is executed in parallel over i , as $\alpha \rightarrow 0^+$, both the expected decrease in L , and the bias $\mathbb{E}[\mu_i^{(t+1)} - \bar{\mu}_i^{(t+1)}]_d$, are $O(\alpha)$ for all d .*

Proof. Let $\eta_i^{(t)}$ and $\mu_i^{(t)}$ be the pre-update natural and mean parameters of $\tilde{q}_i^{(t)}$ respectively. Furthermore, let

$$\begin{aligned}\mu_i'^{(t)} &= \mu_i^{(t)} - \alpha^{-1} \left[\tilde{F}_i^{(t)}(\mu_i) \right]^{-1} \frac{\partial \lambda_i}{\partial \mu_i} \frac{\partial L}{\partial \lambda_i} + \xi_i \\ &= \mathbb{E}_{q_i(z)}[s(z)] + \xi_i,\end{aligned}\tag{D.17}$$

be the mean parameters of $\tilde{q}_i^{(t)}$ after an EP inner update (6.4) but *before* being mapped to mean parameters of $\tilde{q}_i^{(t+1)}$ through map (6.11). ξ_i is some zero-mean noise, e.g. due to MC variation. The second equality follows from the results of Appendix D.2. Similarly, let $\eta_i'^{(t)} = \nabla A^*(\mu_i'^{(t)})$ be the corresponding natural parameters. Now apply map (6.11) to convert to mean parameters of $\tilde{q}_i^{(t+1)}$,

$$\begin{aligned}\mu_i^{(t+1)} &= \left(\nabla A \circ \eta_i^{(t+1)} \circ (\eta_i^{(t)})^{-1} \circ \nabla A^* \right) (\mu_i'^{(t)}) \\ &= \nabla A \left(\left(\eta_i^{(t+1)} \circ (\eta_i^{(t)})^{-1} \right) \left(\nabla A^* (\mu_i'^{(t)}) \right) \right) \\ &= \nabla A \left(\nabla A^* (\mu_i^{(t)}) + \alpha \sum_j \left(\nabla A^* (\mu_j^{(t)'}) - \nabla A^* (\mu_j^{(t)}) \right) \right),\end{aligned}\tag{D.18}$$

where the last equality follows from the definition of $\eta_i^{(t)}(\cdot)$, and by observing that

$$\begin{aligned}\lambda_j^{(t+1)} - \lambda_j^{(t)} &= \alpha(\eta_j' - \eta_j) \\ &= \alpha \left(A^* (\mu_j^{(t)'}) - \nabla A^* (\mu_j^{(t)}) \right),\end{aligned}\tag{D.19}$$

for all j . Let $\bar{\mu}_i'^{(t)}$ be the mean parameters of $\tilde{q}_i^{(t)}$ after a *noise-free* update, but before mapping to mean parameters of $\tilde{q}_i^{(t+1)}$, so that $\mu_i'^{(t)} = \bar{\mu}_i'^{(t)} + \xi_i$. Substituting this in, we have

$$\mu_i^{(t+1)} = \nabla A \left(\nabla A^* (\mu_i^{(t)}) + \alpha \sum_j \left(\nabla A^* (\bar{\mu}_j^{(t)'} + \xi_j) - \nabla A^* (\mu_j^{(t)}) \right) \right).\tag{D.20}$$

Taking a Taylor expansion around $\alpha = 0$, we have

$$\mu_i^{(t+1)} = \mu_i^{(t)} + \alpha \left[\nabla^2 A \left(\nabla A^* (\mu_i^{(t)}) \right) \right] \left(\sum_j \nabla A^* (\bar{\mu}_j^{(t)'} + \xi_j) - \nabla A^* (\mu_j^{(t)}) \right) + O(\alpha^2).\tag{D.21}$$

Now subtract the noise-free update $\bar{\mu}_i^{(t+1)}$, and take expectations with respect to ξ_j for $j = 1, \dots, m$,

$$\mathbb{E} \left[\mu_i^{(t+1)} - \bar{\mu}_i^{(t+1)} \right] = \mathbb{E} \left[\alpha \left[\nabla^2 A \left(\nabla A^* \left(\mu_i^{(t)} \right) \right) \right] \left(\sum_j \nabla A^* \left(\bar{\mu}_j^{(t)'} + \xi_j \right) - \nabla A^* \left(\bar{\mu}_j^{(t)'} \right) \right) + O(\alpha^2) \right]. \quad (\text{D.22})$$

the first term on the right hand side does not have zero expectation in general— $\nabla A^*(\cdot)$ is not generally affine—and so the bias is $O(\alpha)$ as $\alpha \rightarrow 0^+$.

To see that the expected change in L is also $O(\alpha)$, take the Taylor expansion of L along the update direction—obtained by subtracting λ_i from the right hand side of (6.4))—as a function of the step size around $\alpha = 0$,

$$-\alpha \left(\eta_0 + \sum_{j \neq i} \lambda_j - \nabla A \left(E_{q_i(z)}[s(z)] + \xi_i \right) \right)^\top \frac{\partial L}{\partial \lambda_i} + O(\alpha^2), \quad (\text{D.23})$$

which is clearly $O(\alpha)$ as $\alpha \rightarrow 0^+$. $\square$

Proposition 5. *After update (6.17) is executed in parallel over i , as $\epsilon \rightarrow 0^+$, the expected decrease in L is $O(\epsilon)$, and the bias $\mathbb{E} \left[\mu_i^{(t+1)} - \bar{\mu}_i^{(t+1)} \right]_d$ is $O(\epsilon^2)$, for all d .*

Proof. Let $\eta_i^{(t)}$ and $\mu_i^{(t)}$ be the pre-update natural and mean parameters of $\tilde{q}_i^{(t)}$ respectively. Furthermore, let

$$\begin{aligned} \mu_i'^{(t)} &= \mu_i^{(t)} - \epsilon \left(\nabla A \left(\eta_0 + \sum_j \lambda_j \right) - \mathbb{E}_{q_i(z)}[s(z)] - \xi_i \right) \\ &= \mu_i^{(t)} - \epsilon \left(\mu_i^{(t)} - \mathbb{E}_{q_i(z)}[s(z)] - \xi_i \right), \end{aligned} \quad (\text{D.24})$$

be the mean parameters of $\tilde{q}_i^{(t)}$ after an EP- μ update (6.17) but *before* being mapped to mean parameters of $\tilde{q}_i^{(t)}$ through (6.11). ξ_i is some zero-mean noise, e.g. due to MC variation. The second equality follows from the definition $\mu_i^{(t)} = \mathbb{E}_{\tilde{q}_i^{(t)}(z)}[s(z)]$, and the fact that $\tilde{q}_i^{(t)} = q$ before an update. Now apply map (6.11) to convert to mean

parameters of $\tilde{q}_i^{(t+1)}$,

$$\begin{aligned}
\mu_i^{(t+1)} &= \left(\nabla A \circ \eta_i^{(t+1)} \circ (\eta_i^{(t)})^{-1} \circ \nabla A^* \right) (\mu_i'^{(t)}) \\
&= \nabla A \left(\left(\eta_i^{(t+1)} \circ (\eta_i^{(t)})^{-1} \right) \left(\nabla A^* (\mu_i'^{(t)}) \right) \right) \\
&= \nabla A \left(\nabla A^* (\mu_i^{(t)}) + \sum_j \left(\nabla A^* (\mu_j^{(t)'}) - \nabla A^* (\mu_j^{(t)}) \right) \right),
\end{aligned} \tag{D.25}$$

where the last equality is obtained from the definition of $\eta_i^{(t)}(\cdot)$ for $\alpha = 1$, and using

$$\begin{aligned}
\lambda_j^{(t+1)} - \lambda_j^{(t)} &= \eta_j' - \eta_j \\
&= A^* (\mu_j^{(t)'}) - \nabla A^* (\mu_j^{(t)})
\end{aligned} \tag{D.26}$$

for all j . Substituting in the definition of $\mu_j'^{(t)}$, we have

$$\begin{aligned}
\mu_i^{(t+1)} &= \nabla A \left(\nabla A^* (\mu_i^{(t)}) \right. \\
&\quad \left. + \sum_j \left[\nabla A^* \left((1 - \epsilon) \mu_j^{(t)} + \epsilon (\mathbb{E}_{q_j(z)}[s(z)] + \xi_j) \right) - \nabla A^* (\mu_j^{(t)}) \right] \right).
\end{aligned} \tag{D.27}$$

Taking a Taylor expansion around $\epsilon = 0$, we have

$$\begin{aligned}
\mu_i^{(t+1)} &= \mu_i^{(t)} + \epsilon \left[\nabla^2 A \left(\nabla A^* (\mu_i^{(t)}) \right) \right] \sum_j \frac{\partial}{\partial \epsilon} \left\{ \right. \\
&\quad \left. \nabla A^* \left((1 - \epsilon) \mu_j^{(t)} + \epsilon (\mathbb{E}_{q_j(z)}[s(z)] + \xi_j) \right) \right\} \Big|_0 + O(\epsilon^2) \\
&= \mu_i^{(t)} + \epsilon \left[\nabla^2 A \left(\nabla A^* (\mu_i^{(t)}) \right) \right] \left[\nabla^2 A^* (\mu_i^{(t)}) \right] \sum_j \frac{\partial}{\partial \epsilon} \left\{ \right. \\
&\quad \left. (1 - \epsilon) \mu_j^{(t)} + \epsilon (\mathbb{E}_{q_j(z)}[s(z)] + \xi_j) \right\} \Big|_0 + O(\epsilon^2) \\
&= \mu_i^{(t)} + \epsilon \left[\nabla^2 A \left(\nabla A^* (\mu_i^{(t)}) \right) \right] \left[\nabla^2 A^* (\mu_i^{(t)}) \right] \sum_j \left(\right. \\
&\quad \left. \mathbb{E}_{q_j(z)}[s(z)] + \xi_j - \mu_j^{(t)} \right) + O(\epsilon^2).
\end{aligned} \tag{D.28}$$

Finally, subtract the noise free $\bar{\mu}_i^{(t+1)}$, and take expectations (over ξ_j for $j = 1, \dots, m$) to obtain the bias

$$\mathbb{E}[\mu_i^{(t+1)} - \bar{\mu}_i^{(t+1)}] = \mathbb{E}\left[\epsilon \left[\nabla^2 A\left(\nabla A^*(\mu_i^{(t)})\right) \right] \left[\nabla^2 A^*(\mu_i^{(t)}) \right] \sum_j \xi_j + O(\epsilon^2)\right]. \quad (\text{D.29})$$

By assumption $E[\xi_j] = \mathbf{0} \forall j$, hence the first order term disappears, and the bias is $O(\epsilon^2)$.

To see that the expected decrease in L is also $O(\epsilon)$, note that we are simply following the gradient in L with respect to μ , multiplied by the inverse Fisher and scaled by ϵ . Neither the reparameterisation (from λ_i to μ_i) or the Fisher depend on ϵ , and so the change in L must be $O(\epsilon)$ as $\epsilon \rightarrow 0^+$ by simple Taylor expansion arguments. $\square$

D.6 Implicit geometries of EP variants

In Section 6.3 we showed that the updates of EP, EP- η , and EP- μ can be interpreted as performing NGD of L with respect to the distributions $\tilde{q}_1^{(t)}, \dots, \tilde{q}_m^{(t)}$. The SNEP method of Hasenclever et al. (2017), in contrast, performs NGD of L with respect to the *site potentials*, treating them as bona fide distributions in $\mathcal{F}$ for the purposes of NGD – that is, NGD is performed with respect to the distributions with densities given by $\exp(\lambda_i^\top s(z) - A(\lambda_i))$ for $i = 1, \dots, m$. In Section 6.3.4, we argued that the statistical manifolds of $\tilde{q}_1^{(t)}, \dots, \tilde{q}_m^{(t)}$ are far more relevant for the optimisation of L than those of the site potential pseudo-distributions. In this appendix we shall justify this claim.

The loss function L , given by (6.3), has a unique optimum with respect to $[\lambda_i]_i$ when $\nabla_{\lambda_i} L(\eta, [\lambda_j]_j) = 0 \forall i$, which implies that

$$\begin{aligned} \nabla A(\eta_0 + \sum_j \lambda_j) &= -\beta_i \nabla_{\lambda_i} A_i(\eta - \beta^{-1} \lambda_i, \beta^{-1}) \Leftrightarrow \\ \mathbb{E}_{q(z)}[s(z)] &= \mathbb{E}_{q_i(z)}[s(z)]. \end{aligned} \quad (\text{D.30})$$

The minimisation of L with respect to $[\lambda_i]_i$ can therefore be viewed as a joint optimisation in the $m + 1$ distributions $q, q_1, \dots, q_m$, with the minimum found when their expected $\mathcal{F}$ sufficient statistics match. Intuitively, it seems that optimisation of L with respect to λ_i should be performed with consideration for the geometry of q and q_i . By instead performing NGD with respect to the distributions $\tilde{q}_1^{(t)}, \dots, \tilde{q}_m^{(t)}$, in the case of EP, EP- η and EP- μ , or with respect to the site potential psuedo-distributions, in the case of SNEP, these methods can be interpreted as using surrogate distributions

for NGD, as described in Chapter 5. It is possible to show that the Fisher information matrix (FIM) of $\tilde{q}_i^{(t)}$ with respect to λ_i is equal to that of q , up to a scalar constant, and furthermore, it can also be seen as a close approximation to that of q_i . This motivates the use of NGD with respect to $\tilde{q}_i^{(t)}$ for the optimisation of L . In contrast, the FIMs of the site-potential pseudo-distributions, used by SNEP, can bear little relation to those of q or q_i . In particular, the difference will be greatest when $\eta_0 + \sum_j \lambda_j$ is significantly different than λ_i , and we believe this may be why Hasenclever et al. (2017) found that increasing the number of sites reduced the convergence speed of SNEP.

We can make these arguments more concrete by considering the curvature of L . Taking second derivatives of L with respect to λ_i , we have

$$\nabla_{\lambda_i}^2 L(\eta, [\lambda_j]_j) = \nabla_{\lambda_i}^2 A(\eta_0 + \sum_j \lambda_j) + \beta_i \nabla_{\lambda_i}^2 A_i((\eta - \beta_i^{-1} \lambda_i, \beta_i^{-1})). \quad (\text{D.31})$$

Let $\lambda_j^{(t)} = \lambda_j \forall j$ at time t , then

$$\begin{aligned} \nabla_{\lambda_i=\lambda_i^{(t)}}^2 L(\eta, [\lambda_j]_j) &= \nabla_{\lambda_i}^2 A(\eta_0 + \sum_j \lambda_j^{(t)} + (\lambda_i - \lambda_i^{(t)})) + \beta_i \nabla_{\lambda_i}^2 A_i((\eta - \beta_i^{-1} \lambda_i, \beta_i^{-1})) \\ &= \frac{\partial \mu_i}{\partial \eta_i} + \beta_i \nabla_{\lambda_i}^2 A_i((\eta - \beta_i^{-1} \lambda_i, \beta_i^{-1})), \end{aligned} \quad (\text{D.32})$$

where η_i and μ_i are the natural and mean parameters of $\tilde{q}_i^{(t)}$, respectively. The Hessian of the second term will not generally be available in closed form, or easily invertible. We can however find a convenient approximation. Note that after an outer update, at time t , we have $\eta = \eta_0 + \sum_j \lambda_j^{(t)}$. Also we have that $c \exp(\lambda_i^\top s(z)) \approx \exp(\ell_i(z))$, at least near convergence, where c is some unknown scalar constant. Combining these,

we have

$$\begin{aligned}
\nabla_{\lambda_i}^2 A_i((\eta - \beta_i^{-1} \lambda_i, \beta_i^{-1})) &= \nabla_{\lambda_i}^2 \left\{ \log \int \exp \left((\eta - \beta^{-1} \lambda_i)^\top s(z) + \beta^{-1} \ell_i(z) \right) d\nu(z) \right\} \\
&\approx \nabla_{\lambda_i}^2 \left\{ \log \int \exp \left(\left(\eta_0 + \sum_j \lambda_j^{(t)} - \beta^{-1} \lambda_i + \beta^{-1} \lambda_i^{(t)} \right)^\top s(z) \right) d\nu(z) \right\} \\
&= \nabla_{\lambda_i}^2 A \left(\eta_0 + \sum_j \lambda_j^{(t)} + \beta^{-1} (\lambda_i^{(t)} - \lambda_i) \right) \\
&= \nabla_{\lambda_i}^2 A \left(\eta_0 + \sum_j \lambda_j^{(t)} + \beta^{-1} (\lambda_i - \lambda_i^{(t)}) \right) \\
&= \beta^{-2} \frac{\partial \mu_i}{\partial \eta_i}.
\end{aligned} \tag{D.33}$$

Combining this with (D.32), we have

$$\nabla_{\lambda_i}^2 L(\eta, [\lambda_j]_j) \approx (1 + \beta_i^{-1}) \frac{\partial \mu_i}{\partial \eta_i}, \tag{D.34}$$

and if we use this curvature approximation to perform a quasi-Newton step in L with respect to λ_i and step size $\tilde{\epsilon}$, we have

$$\begin{aligned}
\lambda_i &\leftarrow \lambda_i - \tilde{\epsilon} (1 + \beta_i^{-1})^{-1} \frac{\partial \eta_i}{\partial \mu_i} \nabla_{\lambda_i} L(\eta, [\lambda_j]_j) \\
&= \lambda_i - \tilde{\epsilon} (1 + \beta_i^{-1})^{-1} \frac{\partial \eta_i}{\partial \mu_i} \left(\nabla A(\eta_0 + \sum_j \lambda_j) - \mathbb{E}_{q_i(z)}[s(z)] \right).
\end{aligned} \tag{D.35}$$

By letting $\epsilon = \tilde{\epsilon} (1 + \beta_i^{-1})^{-1}$ we have equivalence with the update of EP- η , given by (6.16). We therefore have that EP- η is equivalent to performing a quasi-Newton step in L . Furthermore, the approximate curvature becomes exact at the optimum if $\ell_i(z)$ is an affine function of $s(z)$, e.g. if $\exp(\ell_i(z))$ is an (unnormalised) member of $\mathcal{F}$.

In contrast, let us consider the implicit curvature approximation used by SNEP. Let γ_i be mean parameters of the member of $\mathcal{F}$ with natural parameter λ_i . To perform the inner minimisation of the variational objective, SNEP performs NGD in L with respect to γ_i . The implicit curvature matrix used by the SNEP update is then the Fisher with respect to γ_i , which is given by $\frac{\partial \lambda_i}{\partial \gamma_i} = \nabla A^*(\gamma_i)$. Now consider the actual

curvature of L with respect to γ_i ,

$$\nabla_{\gamma_i}^2 L(\gamma_i) = \frac{\partial \lambda_i}{\partial \gamma_i} \nabla_{\lambda_i}^2 L(\eta, \lambda_1, \dots, \lambda_m) \frac{\partial \lambda_i}{\partial \gamma_i} + \sum_{k=1}^d \left[\frac{\partial L}{\partial \lambda_i} \right]_k \nabla_{\gamma_i}^2 \lambda_i(\gamma_i), \quad (\text{D.36})$$

where we have used $\lambda_i(\gamma_i) = \nabla A^*(\gamma_i)$ to denote the backwards map. Near the optimum we have $\frac{\partial L}{\partial \lambda_i} \approx \mathbf{0}$, and so the curvature will be approximately equal to the first term as we approach the optimum. We have already shown that

$$\nabla_{\lambda_i}^2 L(\eta, [\lambda_j]_j) \approx (1 - \beta_i^{-1}) \frac{\partial \mu_i}{\partial \eta_i} = (1 - \beta_i^{-1}) \nabla A(\eta_0 + \sum_j \lambda_j), \quad (\text{D.37})$$

and so near the optimum, the curvature with respect to γ_i is

$$\begin{aligned} \nabla_{\gamma_i}^2 L(\gamma_i) &\approx (1 - \beta_i^{-1}) \frac{\partial \lambda_i}{\partial \gamma_i} \nabla^2 A(\eta_0 + \sum_j \lambda_j) \frac{\partial \lambda_i}{\partial \gamma_i} \\ &= (1 - \beta_i^{-1}) \frac{\partial \lambda_i}{\partial \gamma_i} \nabla^2 A(\eta_0 + \sum_j \lambda_j) \left(\nabla^2 A(\lambda_i) \right)^{-1}. \end{aligned} \quad (\text{D.38})$$

Compare this with the implicit curvature used by SNEP of $\frac{\partial \lambda_i}{\partial \gamma_i}$, and it is clear that the two are proportional only when $\eta_0 + \sum_j \lambda_j = \lambda_i$. This suggests that if there are many sites (m is large), or if the prior is very informative, the curvature matrix implicitly used by SNEP is likely to be significantly different from the true curvature, even near the optimum.

D.7 Computational cost analysis

Let c_{samp} be the cost of drawing a single sample from one of the tilted distributions. Let c_{fwd} be the cost of converting from natural to mean parameters in the base family $\mathcal{F}$ (the forward mapping). Similarly, let c_{bwd} be the cost of converting from mean to natural parameters in $\mathcal{F}$ (the backward mapping). Recall that m is the number of sites. We now state the computational complexity of a round of parallel updates in each of the EP variants evaluated in Section 6.4.

EP EP draws n_{samp} samples from each of the m sites. It then performs m backward mappings, to map from moments back to updated site parameters. The total cost is then $mn_{\text{samp}}c_{\text{samp}} + mc_{\text{bwd}}$.

EP- η EP- η draws n_{samp} samples from each of the m sites. Each update also involves one forward mapping per iteration. In addition, each iteration also requires m JVPs through the backward mapping $\nabla A^*(.)$. The primals of this JVP are the same for each site, so the linearisation only needs to be performed once per update, costing $\approx 2c_{\text{bwd}}$, but we have m tangents. The overall cost of an EP- η iteration is therefore $\approx mn_{\text{samp}}c_{\text{samp}} + c_{\text{fwd}} + (2 + m)c_{\text{bwd}}$.

EP- μ EP- μ draws n_{samp} samples from each of the m sites. Each update involves one forward mapping, and m backward mappings per iteration, but notably does *not* require any JVPs. The overall cost of an EP- μ iteration is therefore $mn_{\text{samp}}c_{\text{samp}} + c_{\text{fwd}} + mc_{\text{bwd}}$.

SNEP SNEP draws n_{samp} samples from each of the m sites, and performs m backward mappings per iteration. The total cost is then $mn_{\text{samp}}c_{\text{samp}} + mc_{\text{bwd}}$, which is the same as EP.

In this work we largely assume that c_{samp} is the dominant cost. This is often the case when the sampling is performed using MCMC. For example, in the NUTS sampler with default numpyro settings – as used in our evaluation – each sample can involve evaluating up to 1024 gradients of the tilted distribution log density.

The other costs, c_{fwd} and c_{bwd} , depend on $\mathcal{F}$. In the case of a MVN family with dense covariance, the forward and backward mappings are $O(d^3)$, for d the dimensionality of z . When d is large, c_{fwd} and c_{bwd} become more significant, in which case the balance will shift in favour of using more samples per update (and taking larger steps). When d is very large, however, it may be necessary to choose a diagonal covariance base family instead, in which case the mappings can be performed in $O(d)$ time. Also note that when d increases, c_{samp} will also typically increase at a rate faster than d , even with optimal tuning (Brooks et al., 2011). In our evaluation we used just a single sample per update for EP- η and EP- μ , and they significantly outperformed the baselines when measure in both NUTS steps *and* wall-clock-time.

We also note that the per-iteration cost of c_{fwd} and c_{bwd} could be reduced by exploiting the results of previous iterations. For example, in the case of a MVN with dense covariance, the cost of the forward and backward mappings are dominated by matrix inversions. This cost could be significantly reduced by using the result from the previous iteration to warm-start a Newton-style iterative inversion routine. This approach was used by Anil et al. (2021) to reduce the cost of computing inverse matrix roots as part of a wider deep-learning optimisation scheme. We did not make use of such optimisations in our experiments.

D.8 Evaluation details

In this appendix, we give details about the experiments presented in Section 6.4. To evaluate the performance of the different variants, we monitored KL divergence to an estimate of the optimum, obtained by running EP to convergence with a large number of samples ($n_{\text{samp}} = 10^5$ for the final iterations). We used 500 different hyperparameter settings for each variant, chosen using random search, and repeated each run using 5 different random seeds, which were used to seed the MCMC samplers. Hyperparameter settings that failed in any of the 5 runs were discarded. We used this setup for all experiments in Section 6.4.

The x -axis in Figure 6.3 of Section 6.4 corresponds to the number of NUTS steps, or more specifically, the number of NUTS candidate evaluations. This number is hardware and implementation agnostic, and roughly corresponds to the total computational cost incurred up to that point. Wall-clock time results are given in Appendix D.9.

Hyperparameter search spaces The step size for EP, α , was drawn log-uniformly in the range $(10^{-4}, 1)$, and n_{samp} was drawn log-uniformly between $[d + 2.5, 10000.5)$ and then rounded to the nearest integer, where d is the dimensionality of z . Note that the debiasing estimator used by Xu et al. (2014) for MVN families is not defined for $n_{\text{samp}} \leq d + 2$. The thinning ratio was drawn uniformly from $\{1, 2, 3, 4\}$. The step size for EP- η and EP- μ , ϵ , was drawn log-uniformly in the range $(10^{-5}, 10^{-2})$, and n_{samp} was fixed to 1. The step size for SNEP was drawn log-uniformly in the range $(10^{-5}, 10^{-2})$, with n_{samp} and n_{inner} (independently) drawn log-uniformly in the range $[.5, 10.5)$ and then rounded to the nearest integer.

Double-loops All the methods considered can be used in a double-loop manner by taking $n_{\text{inner}} > 1$ inner steps per outer update. We did not find this necessary in any of our experiments, and so we fixed $n_{\text{inner}} = 1$ for EP, EP- η and EP- μ . For SNEP we included n_{inner} in the search space (as described above) to ensure that setting it to 1 was not negatively impacting its performance.

Hardware All experiments were executed on 76-core Dell PowerEdge C6520 servers, with 256GiB RAM, and dual Intel Xeon Platinum 8368Q (Ice Lake) 2.60GHz processors. Each individual run was assigned to a single core.

Software Implementations were written in JAX (Bradbury et al., 2018), with NUTS (Hoffman and Gelman, 2014) used as the underlying sampler. We used the numpyro

(Phan et al., 2019) implementation of NUTS with default settings. For experiments with a NIW base family, we performed mean-to-natural parameter conversions using the method described in Appendix B.2, with JAXopt (Blondel et al., 2022) used to perform implicit differentiation through the iterative solve.

MCMC hyperparameter adaptation We performed regular warm-up phases in order to adapt the samplers to the constantly evolving tilted distributions, consistent with prior work (Xu et al., 2014, Vehtari et al., 2020). The frequency and duration of these warm-up phases was also included in the hyperparameter search as follows. We drew the duration (number of samples) of each warm-up phase log-uniformly in the range $[99.5, 1000.5)$ and then rounded to the nearest integer. We then drew the sampling-to-warm-up ratio log-uniformly in the range $(1, 4)$. This ratio then determined how frequently the warm-up was performed, which we rounded to the nearest positive integer number of updates. Note that the Pareto frontier plots in Figures 6.3 and D.1 include the computation / time spent during warm-up phases.

Site parameter initialisation For EP, EP- η and EP- μ , we initialised the site parameters to $\lambda_i = \mathbf{0} \ \forall i$ for all models, with the exception of the cosmic radiation model for reasons we discuss later. SNEP is not compatible with improper site potentials, and so for models where $\mathcal{F}$ was MVN we used $\lambda_i = (2m)^{-1}\eta_0 \ \forall i$, consistent with Vehtari et al. (2020). Unfortunately this initialisation strategy for SNEP is not always valid when $\mathcal{F}$ is NIW. This is because the site potentials in SNEP are restricted to being proper distributions, and this constraint was violated when using this initialisation in our experiments. We tried various other initialisation strategies, but none produced satisfactory results or allowed us to make a meaningful comparison, hence we omitted SNEP from the experiments with NIW $\mathcal{F}$; namely, the political survey hierarchical logistic regression and neural response model experiments.

We set $\beta_i = 1 \ \forall i$ in all experiments, corresponding to the original (not power) EP of Minka (2001). All of the models in these experiments followed the same general structure, given by

$$p_0(z) \prod_i p(w_i | z) p(\mathcal{D}_i | w_i, z), \quad (\text{D.39})$$

where p_0 is a member of a tractable, minimal exponential family $\mathcal{F}$. This is shown graphically in Figure 6.2. We now provide details about individual experiments.

D.8.1 Hierarchical logistic regression

In the hierarchical logistic regression (HLR) experiments, there were m groups (logistic regression problems) each with n covariate/response pairs, so that $\mathcal{D}_i = \{(x_{i,j}, y_{i,j})\}$, $x_{i,j} \in \mathbb{R}^d$, and $y_{i,j} \in \{0, 1\}$, for $i = 1$ to m , and $j = 1$ to n . Each group had its own unobserved vector of regression coefficients $w_i \in \mathbb{R}^d$, with density $p(w_i | z)$ parameterised by global parameters z .

Synthetic data with MVN prior In the synthetic experiment, we had $m = 16$ groups, with $d = 4$ and $n = 20$. $\mathcal{F}$ was MVN, and $z = (\mu_1, \log \sigma_1^2, \dots, \mu_4, \log \sigma_4^2) \in \mathbb{R}^8$ corresponded to the means and log-variances for the independent normal density $p(w_i | z) = \prod_j \mathcal{N}(w_{i,j}; \mu_j, \sigma_j^2)$.¹ The prior on z was $\mathcal{N}(\mathbf{0}, \text{diag}((4, 2, 4, 2, \dots)))$, where $\text{diag}(\cdot)$ constructs a diagonal matrix from its vector-valued argument. The dataset was generated using the same procedure as Vehtari et al. (2020).

Political survey data with NIW prior In the political survey data experiment, $\mathcal{F}$ was a normal-inverse-Wishart (NIW) family of distributions, and $z = (\mu, \text{vech}(\Sigma))$ was used to parameterise the group-level coefficient density $p(w_i | z) = \mathcal{N}(w_i; \mu, \Sigma)$. p_0 was a NIW prior on μ and Σ , with parameters $\mu_0 = \mathbf{0}$, $\nu = 9$, $\lambda = 1$, and $\Psi = I$ (following Wikipedia notation). We used a log-Cholesky parameterisation of Σ for the purposes of sampling. The dataset consisted of binary responses to the statement “*Allow employers to decline coverage of abortions in insurance plans (Support / Oppose)*”. We constructed $m = 50$ regression problems, corresponding to the 50 US states, and truncated the data so that there were exactly $n = 97$ responses for each state. We used 6 predictor variables from the dataset, corresponding to characteristics of a given survey participant. The predictors were binary variables conveying: age (3 groups), ethnicity (2 groups), education (3 groups) and gender. We also included a state-level intercept to capture variation in the base response level between states, so that $d = 7$. This setup is based on one used by Lopez-Martin et al. (2022). The data is available at <https://github.com/JuanLopezMartin/MRPCaseStudy>.

¹Note that in this section we use μ to denote the mean of a normal or multivariate normal distribution. This should not to be confused with the usage for exponential family mean parameters in the main text.

D.8.2 Cosmic radiation model

In this experiment, there were m nonlinear regression problems, each corresponding to a model of the relationship between diffuse galactic far ultraviolet radiation (FUV) and 100- μm infrared (i100) emission in a particular sector of the observable universe.

The nonlinear regression model for sector i had parameters $w_i \in \mathbb{R}^9$. The m regression problems were related through common hyperparameters $z \in \mathbb{R}^{18}$, which parameterised the section-level densities $p(\mathcal{D}_i, w_i | z)$ where $\mathcal{D}_i$ is the observed data for sector i . $\mathcal{F}$ was the family of MVN distributions, and the prior on $z \in \mathbb{R}^{18}$ was $\mathcal{N}(\mathbf{0}, 10I)$. The specifics of $p(\mathcal{D}_i, w_i | z)$ are quite involved and we refer the reader to Vehtari et al. (2020) for further details.

Vehtari et al. (2020) applied this model to data obtained from the Galaxy Evolution Explorer telescope. We were unable to obtain the dataset, and so we generated synthetic data using hyperparameters that were tuned by hand to try and match the qualitative properties of the original data set (see Appendix D.11 for examples). We used a reduced number of $m = 36$ sites and $n = 200$ observations per site to reduce computation, allowing us to perform a comprehensive hyperparameter search.

Using the conventional site parameter initialisation of $\lambda_i = \mathbf{0} \forall i$ resulted in most runs of EP failing during early iterations. We found that this was resolved by initialising with the method used by Vehtari et al. (2020) for SNEP, that is, $\lambda_i = (2m)^{-1}\eta_0 \forall i$, and so we used this initialisation for all methods. We note, however, that the performances of EP- η and EP- μ were largely unaffected by this change.

D.8.3 Neural response model

In this experiment we performed inference in a hierarchical Bayesian neural response model, using recordings of V1 complex cells in an anaesthetised adult cat. 10 neurons in a specific area of cat V1 were simultaneously recorded under the presentation of 18 different visual stimuli, each repeated 8 times, for a total of 144 trials. Neural data were recorded by Tim Blanche in the laboratory of Nicholas Swindale, University of British Columbia, and downloaded from the NSF-funded CRCNS Data Sharing website <https://crcns.org/data-sets/vc/pvc-3> (Blanche, 2009).

$z = (\mu, \text{vech}(\Sigma)) \in \mathbb{R}^{65}$ was used to parameterise $\mathcal{N}(\log r_j; \mu, \Sigma)$, the density for latent log firing rates in trial j , for $j = 1, \dots, 144$. $p_0 \in \mathcal{F}$ was NIW with parameters $\mu_0 = \mathbf{1}$, $\nu = 12$, $\lambda = 2.5$, and $\Psi = 1.25I$ (again following Wikipedia notation). The observed spike count for neuron k in trial j , $x_{j,k} \in \mathbb{N}$, was modelled as

$\text{Poisson}(c_{j,k}, \exp(r_{j,k}))$, for $j = 1, \dots, 144$, $k = 1, \dots, 10$. We again used a log-Cholesky parameterisation of Σ for sampling.

We grouped trials together into $m = 8$ batches of $n = 18$ trials, so that the vector $w_i = (\log r_{180(i-1)+1}, \dots, \log r_{180i}) \in \mathbb{R}^{180}$ was the concatenation of log firing rates for all 10 neurons across 18 trials, with $\mathcal{D}_i = \{(x_{j,1}, \dots, x_{j,10})\}$ the observed spike counts, for $j = 180(i-1) + 1$ to $180i$.

D.9 Pareto frontiers in wall-clock time

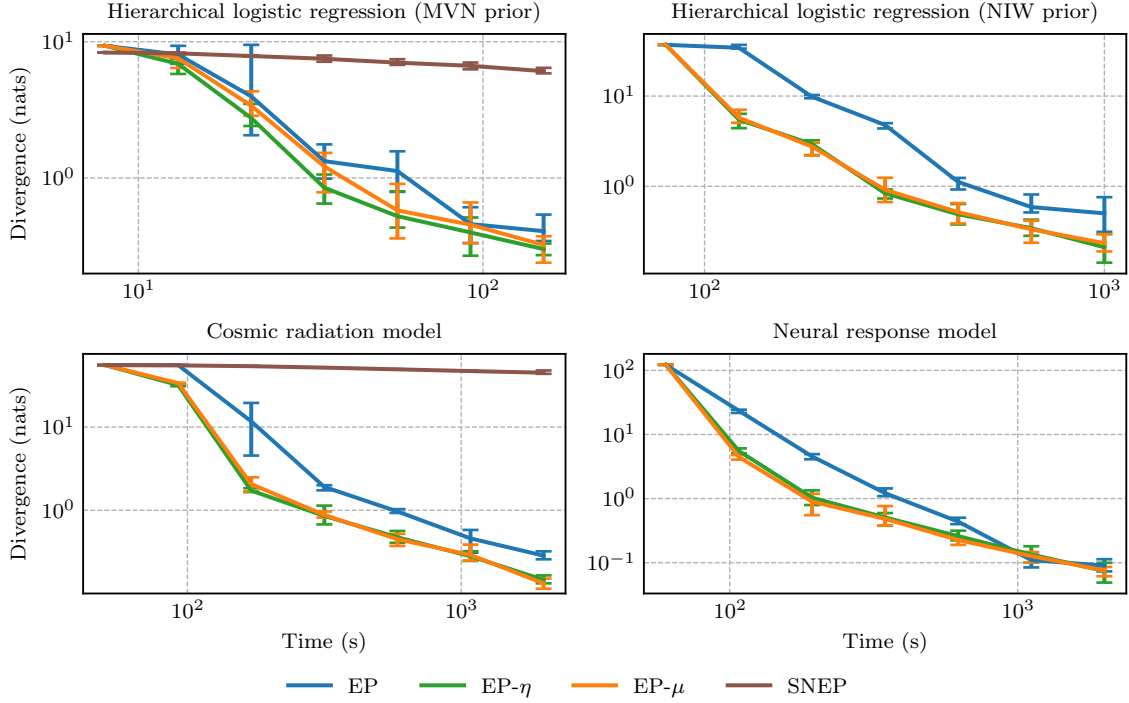


Fig. D.1 Pareto frontiers showing the number of seconds elapsed (x -axis) against the KL divergence from p to an estimate of the optimum (y -axis). Each point on the plot marks the lowest KL divergence attained by *any* hyperparameter setting by that time. Error bars mark the full range of values for the marked hyperparameter setting across 5 random seeds.

In Figure D.1, we show Pareto frontiers, with the y -axis showing the lowest KL divergence achieved by any hyperparameter setting, and the x -axis shows the cumulative number of seconds elapsed – that is, the wall-clock time equivalent of Figure 6.3. These results are in broad agreement with those of Figure 6.3, which suggests that the sampling cost does indeed dominate the computational overheads of EP- η and EP- μ in these experiments. The wall-clock time results for EP- η and EP- μ would likely be improved by using more than one sample per update, by making more efficient use of hardware resources and minimising per-iteration overheads.

We note that these times are necessarily implementation and hardware dependent. We did not make particular efforts to optimise for wall-clock time. In Appendix D.7 we discuss approaches for minimising per-iteration overheads. These would likely improve performance for all methods, but should disproportionately favour EP- η and EP- μ , due to their relatively frequent updates and large per-iteration overheads.

D.10 Pairwise marginals in CVI comparison

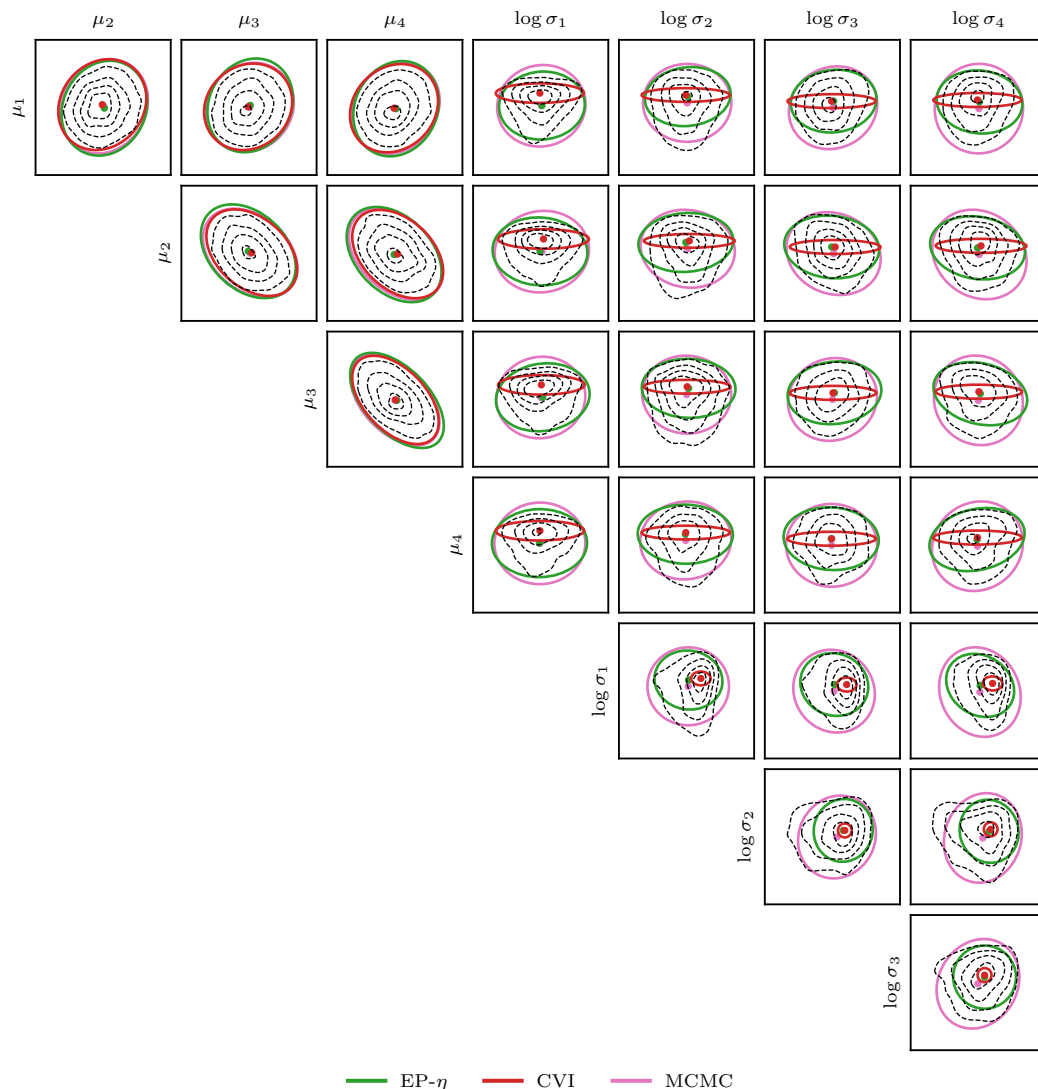


Fig. D.2 Contours of pairwise posterior marginals for the hierarchical logistic regression experiment of Section 6.6, overlaid with the MVN approximations of EP- η , CVI, and one obtained by fitting directly to MCMC samples.

D.11 Cosmic radiation data

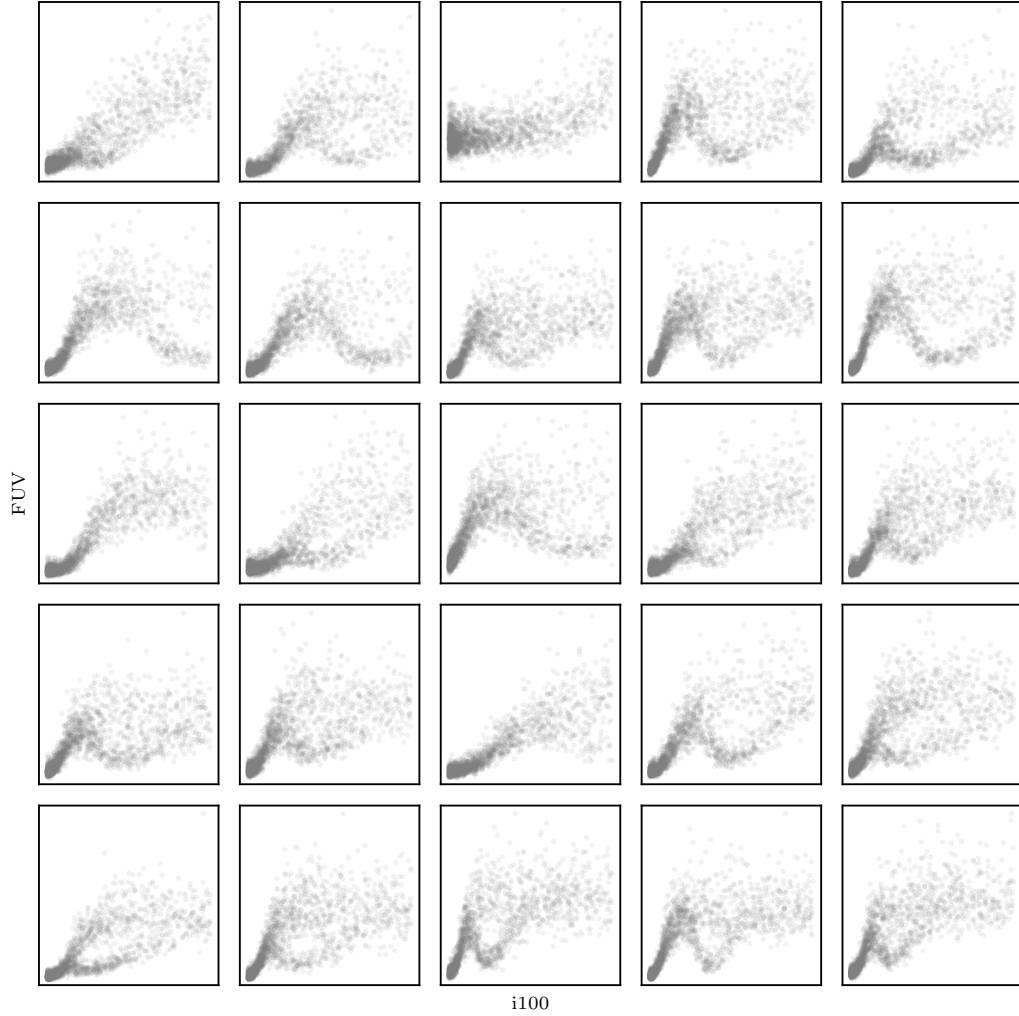


Fig. D.3 Synthetic data, generated using the cosmic radiation model of Vehtari et al. (2020). Each plot shows galactic far ultraviolet radiation (FUV) versus infrared radiation (i100) for a single sector of the observable universe.

Vehtari et al. (2020) used a hierarchical Bayesian model to capture the nonlinear relationship between diffuse galactic far ultraviolet radiation (FUV) and 100- μm infrared emission (i100) in various sectors of the observable universe, using data obtained from the Galaxy Evolution Explorer telescope. We were unable to obtain the dataset, and so we generated synthetic data using hyperparameters that were tuned by hand to try and match the qualitative properties of the original data set. Example data generated using these hyperparameters is shown in Figure D.3. For comparison, see Figure 9 of Vehtari et al. (2020).

