
THE UNIVERSITY OF WESTERN AUSTRALIA

# Comparing Markov Chain Monte Carlo and Variational Methods for Bayesian Inference on Mixtures of Gamma Distributions [1]

Isaac H. Breen

Supervised by John Lau and Edward Cripps

November 15, 2019

1. Submitted in partial fulfilment of the requirements for the Bachelor of Science degree with Honours at the University of Western Australia

Markov chain Monte Carlo (MCMC) and variational inference (VI) are methods for approximating intractable integrals that arise in statistics, machine learning, and physics. In Bayesian statistics, they facilitate posterior inference on mixture models when analytical solutions are unreachable. We derive a variational inference procedure for approximating the posterior of a mixture of gamma distributions. We then compare its performance to that of a Gibbs sampler, an important variant of MCMC that is used widely for mixture models, on several synthetic datasets and a dataset of Australian rainfall. The crucial difference between MCMC and VI is that, while MCMC *samples* from the posterior distribution, VI seeks an *approximation* that minimises the Kullback-Leibler divergence between itself and the true posterior. MCMC is older, more thoroughly studied, and has desirable asymptotic properties. However, VI promises much faster convergence and yields a closed-form approximation to the true posterior. One significant barrier to implementing VI for a mixture of gamma distributions is the need to calculate the expectation $\mathbb{E}[\log \Gamma(\alpha_k)]$ where $\alpha_k$ is the shape parameter of the gamma distribution. Our solution involves using a combination of Stirling's approximation and a Taylor approximation to the log gamma function. A significant drawback of VI is that, in order to make the optimisation problem tractable, we must assume independence between parameters in approximating density. As a consequence, we find that, under the traditional parameterisation of the gamma distribution, VI significantly underestimates the variance of the posterior distribution and fails to capture the relationship between the shape and rate parameters. We show that an orthogonal reparameterization of the gamma distribution in terms of the shape and the mean of the distribution overcomes this short-falling at a negligible computational cost.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Mixtures are an expressive class of statistical distributions that are constructed from a linear combination of simpler component distributions. The are useful for modelling data that consist of multiple distinct subpopulations. The general form of the probability density function (PDF) of a mixture of $K$ distributions is

$$p(x \mid \boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k p_k(x \mid \boldsymbol{\theta}_k)$$

where $p_k(x \mid \boldsymbol{\theta}_k)$ is the PDF of the $k^{\text{th}}$ component, $\boldsymbol{\theta}_k$ are its parameters, $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K)$ is defined for notational convenience, and $\boldsymbol{\pi} = (\pi_1, ..., \pi_K)$ is a vector of mixing weights with the constraints that $\pi_k > 0$ for $k = 1, ..., K$ and $\sum_{k=1}^{K} \pi_k = 1$. It is usual to suppress the first subscript in $p_k(x \mid \boldsymbol{\theta}_k)$ and simply write $p(x \mid \boldsymbol{\theta}_k)$. The most common and well-studied choice for the component distribution is the normal distribution. However, since it has support on the entire real line, the normal distribution is not suitable for modelling non-negative phenomena such as rainfall. For this, can use the *gamma distribution*.

We aim to develop, compare, and contrast methods for performing statistical inference on (or 'guessing' the parameters of) mixtures of gamma distributions[1]. In particular, we take a *Bayesian* perspective on inference, meaning that we wish to infer a probability distribution over the parameters[2]. To this end, the dominant technique in the statistical literature is Markov chain Monte Carlo (MCMC) (Metropolis et al. 1953; Hastings, W 1970), which is a method for sampling from analytically intractable probability densities. Many MCMC algorithms are very well-studied and come with guarantees on their asymp-

1. Fitting mixture distributions is also closely related to clustering; by fitting a modal mixture distribution, we implicitly solve a clustering problem.

2. This contrasts with the *frequentist* perspective in which we assume that the data generating process (which, for our mixture model, is parameterised by $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$) is fixed, thus does not a probability distribution over its parameters.

totic behaviour. For example, under rather general conditions, the Metropolis-Hastings algorithm is guaranteed to converge to its target distribution (Clarke and Billingsley 1980; Madras and Sezer 2010). In some cases, the rates convergence are also provable (Tsvetkov, Hristov, and Angelova-Slavova 2013).

Variational inference (VI) is a more recent approach to Bayesian inference. The fundamental difference between MCMC and VI is that, while MCMC *samples* from the posterior distribution, VI seeks an *approximation* to it that minimises the divergence between itself and the true posterior. Variational inference is generally much faster and yields a closed-form approximation to true posterior. However, though local convergence is typically assured, there is no guarantee of convergence to the true posterior. In particular, for highly modal ('peaky') distributions such as mixtures of gammas, components tend to get 'stuck' in a state where the objective function is locally but not globally optimised. This can occur when the parameters of two components converge to the same values, degenerating into a single component.

Another known drawback of VI is that, in order to make the optimisation tractable for multi-parameter distributions, we usually need to assume independence between parameters in the approximating distribution. This is called the *mean-field* approximation and, like many of the methods employed in this thesis, it originates from statistical physics (specifically in the study of phase transitions (Landau 1936)). The mean-field approximation, allows us to solve (up to a normalising constant) the optimal variational distribution of each parameter while holding the other variational distributions fixed (Bishop 2006, p. 465). This locally optimal distribution is found by taking an expectation of the logarithm of the true posterior. This way, we can optimise the complete variational distribution by repeatedly optimising each of its factors one-after-another. Since this procedure 'climbs' the objective function coordinate-by-coordinate, it is called coordinate ascent-variational inference (CAVI).

Unfortunately, since the mean-field approximation neglects the dependence between parameters, it usually underestimates the variance of the posterior. As we will demonstrate, the gamma distribution with the common shape-rate parameterisation[3] is particularly prone to this effect due to the strong dependence between its parameters. As a remedy, we propose using an orthogonal parameterisation - one for which the Fisher information matrix is diagonal - in terms of the shape and mean. We find that VI derived from this reparameterisation yields a posterior approximation that is much more consistent with the MCMC baseline than that of VI derived from the shape-rate parameterisation. Furthermore, this algorithm produces a much more plausible predictive distribution than

3. The two usual parameterisations of the gamma distribution are the shape-rate parameterisation and the shape-scale parametrisation. Both suffer from the same mean-field approximation problem.

both the MCMC and the VI with the shape-rate parameterisation for models with a high number of components.

## 1.1 Literature review

### 1.1.1 Markov chain Monte Carlo

The variant of MCMC that we study in this thesis is the Gibbs sampler. Although Gibbs samplers are (almost always) a special case of the Metropolis-Hastings algorithm, they, in fact, emerged from two very different efforts. The *Metropolis* algorithm was originally devised for solving "problems in equilibrium statistical mechanics" in a seminal 1953 paper by Greek-American physicist Nicholas Metropolis (after whom the algorithm is named) as well as physicists Edward Teller, Arianna Rosenbluth, Marshall Rosenbluth, and computer scientist Augusta Teller. The term 'Monte Carlo' was coined by Metropolis during his work in designing the early computer MANIAC I (**M**athematical **A**nalyser, **N**umerical **I**ntegrator **A**nd **C**omputer Model I) at the Los Alamos Research Laboratory, the birthplace of the atomic bomb (Robert and Casella 2011).

Canadian statistician W. K. Hastings (1970) generalised the work of Metropolis et al. into a procedure for sampling from high-dimensional probability distributions. This procedure, known today as the Metropolis-Hastings algorithm, works by repeatedly sampling from a (possibly non-symmetric) proposal distribution: for $t = 1, ..., T$, we draw a sample $\theta'$ of parameters from the proposal distribution and either accept it with a certain probability, in which case $\theta^{[t]} \leftarrow \theta'$, or reject it, in which case $\theta^{[t]} \leftarrow \theta^{[t-1]}$. The target posterior density need only be known up to a normalising constant: an essential advantage in Bayesian statistics where normalisation constants are frequently impossible to solve analytically. We provide a detailed treatment of this algorithm in chapter 2. Hastings also recognised that, in multivariate inference, a joint posterior could be sampled by individually sampling each variable from its posterior conditioned on the rest of the variables. This became known as the *Gibbs* sampler. However, it wasn't until many years later that two papers by Geman and Geman (1984) and Gelfand and Smith (1990) convinced the statistical community of the potential of the methods developed by Hastings, W and Metropolis et al. These papers, combined with a dramatic increase in the affordability of computing power, sparked a flurry of research into MCMC methods and their applications.

However, MCMC has its drawbacks. For one, MCMC does not return a closed-form approximation to the posterior, only samples from it. This means that if we wish to find the expectation of some function a random variable, we must compute it as a *Monte Carlo*

expectation. Although this enables us to calculate almost any expectation with ease, it is usually slower than evaluating an analytic solution when one is available (for example, the higher-order moments of the gamma distribution). Additionally, for a large dataset or a very complex model MCMC's rate of convergence may be prohibitively slow.

### 1.1.2 Variational inference

Blei, Kucukelbir, and McAuliffe (2017) trace the first work on variational Bayesian methods to Peterson and Anderson (1987) who use mean-field theory to train a Boltzmann Machine - a type of artificial neural network. Further research into Bayesian neural networks was carried out in the 1990s by neural network pioneer Geoffrey Hinton and colleagues (Williams and Hinton 1991; Frey and Hinton 1999). In parallel, Michael Jordan's lab (Saul, Jaakkola, and Jordan 1996; Jordan et al. 1999) studied variational methods for neural networks as well as for other probabilistic models. The early 2000s saw an explosion of interest in variational Bayesian methods, including a seminal paper in the field of natural language processing by Blei, Ng, and Jordan (2003) on latent Dirichlet allocation which has been cited over $28,000$ time to date.

Variational inference frames Bayesian inference as an optimisation problem rather than a sampling one. The goal is to find an optimal *variational* distribution that minimises some measure of error between itself and the true posterior. The usual choice this error measure is the Kullback-Leibler (KL) Divergence, and it can be shown that minimising the KL Divergence is equivalent to maximising a lower bound on the marginal likelihood $\log p(x) = \log \int_{\Theta} p(\boldsymbol{x}, \boldsymbol{\theta}) d\boldsymbol{\theta}$ of the data $\boldsymbol{x}$ (Jordan et al. 1999; Bishop 2006, p. 463). This bound, known as the **e**vidence **l**ower-**bo**und (ELBO) and denoted by $\mathcal{L}(q)$, is the objective function central to most VI optimisation. Other $f$-divergences (of which KL divergence is a special case) have also been studied (Bamler et al. 2017; Futami, Sato, and Sugiyama 2017; Wang, Liu, and Liu 2018) and may give tighter ELBOs (Leisink and Kappen 2001) or more accurate posterior predictive distributions. Crucially, like the Metropolis-Hastings algorithm, VI only requires the true posterior to be known up to a normalising constant. Variational inference is named after calculus of variations (Euler 1744), the study of using small changes in functions or functionals (a mapping from a vector space to a field of scalars) to find functional minima and maxima, which allows one to find optimal approximating distributions; however, the VI techniques used in this thesis avoid the need to employ the calculus of variations.

By reframing inference as an optimisation problem, VI naturally draws upon techniques such as stochastic optimisation from the mathematical optimisation literature. Hoffman et al. (2013) develop a framework for stochastic variational inference (SVI) in which

the variational distribution is optimised on small subsets of the data at a time. They demonstrate their techniques on a Bayesian mixture of multivariate Gaussians and find that it both decreases computation time and helps avoid local minima. Remarkably, Hoffman et al. also prove that stochastic coordinate-ascent variational inference (CAVI) is equivalent to gradient descent on noisy estimates of the *natural gradient*[4] of the ELBO. Hoffman et al. Ranganath, Gerrish, and Blei (2014) extend this idea to 'black-box' SVI (BBSVI) which leverages the automatic differentiation capabilities of modern numerical computation libraries (Baydin et al. 2018) to vastly reduce the effort required on the part of the practitioner to painstakingly solve expectations (for CAVI and SVI) or derivatives (for SVI).

The mean-field approximation ignores the relationship between variables, leading to underestimation of the variance of the posterior. One approach to restoring this dependency is structured stochastic variational inference (SSVI) (Hoffman and Blei 2015) in which the joint approximating distribution, for example, $\theta_1$ and $\theta_2$ is decomposed into $q(\theta_1, \theta_2) = q(\theta_1 \mid \theta_2)q(\theta_2)$. In pseudocode, SSVI proceeds as follows.

1. Optimise $q(\theta_2)$ by the ascent on a noisy estimate of the natural gradient.
2. Sample $\theta_2'$ from $q(\theta_2)$.
3. Optimise $q(\theta_1 \mid \theta_2')$ and take its 'weighted average' with the previous $q(\theta_1 \mid \theta_2)$.
4. Repeat 1-3 until convergence.

Another strategy proposed by Tran, Blei, and Airoldi (2015) and Tran (2018) is to augment the mean-field variational distribution with *copulas*: multivariate distributions that model the relationship between variables. Hence, they term this strategy *copula variational inference* (CVI). Sklar's theorem (Sklar 1959) asserts that a copulas always exist and that the full probability density of $\boldsymbol{\theta} = (\theta_1, ..., \theta_K)$ can be written as $q(\boldsymbol{\theta}) = c(Q_1(\boldsymbol{\theta}_1), ..., Q_M(\boldsymbol{\theta}_M)) \prod_{i=1}^{M} q_i(\boldsymbol{\theta}_i)$ where $q_i$ is the marginal density of $\boldsymbol{\theta}_1$, $Q_k$ is its cumulative density function, and $c(Q_1(\boldsymbol{\theta}_1), ..., Q_M(\boldsymbol{\theta}_M)) \prod_{i=1}^{M} q_i(\boldsymbol{\theta}_i)$ is the copula function. Tran, Blei, and Airoldi (2015) and Tran (2018) construct the full copula using a copula *vine* which combines sets of bivariate conditional copulas. They demonstrate the superiority of the copula approach on a Gaussian mixture model and find that, although copula VI takes much longer to converge, it produces posterior approximations more consistent to those of a Gibbs sampler than mean-field VI.

4. Hoffman et al. (2013) neatly summarise the meaning of the natural gradient in relation to VI: "While the Euclidean gradient points in the direction of steepest ascent in Euclidean space, the natural gradient points in the direction of steepest ascent in the Riemannian space, that is, the space where local distance is defined by KL divergence rather than the $L^2$ norm."

However, both SSVI and CVI add time and complexity to the optimisation procedure. For a mixture of gamma distributions, we show that it suffices to use an orthogonal parameterisation of the component gamma distributions and that this produces far better posterior approximations and predictive distributions than the naive shape-rate parameterisation at a negligible computational cost.

To our knowledge, no published works have addressed variational inference for a mixture of gamma distributions. Knowles (2015) use gamma variational distributions to approximate the posterior of the parameters of a gamma-distributed latent variable via SVI. Knowles applies this to network data (an infinite edge partition model) (Kemp et al. 2006; Blundell and Teh 2013) as well as gamma process factor analysis. However, they do not consider mixture models.

Chaney (2015) use BBSVI to perform variational inference on a single gamma distribution. Like us, they suggest the same mean-rate parameterisation of the gamma distribution for VI. Their justification for this is that it "makes the variables much easier to fit" and is "more interpretable". While correct, this ignores the most significant advantage of this parameterisation which is that it produces a far superior mean-field posterior approximation.

In an unpublished manuscript, Llera et al. (2016) claim to have implemented variational inference for a mixture of gamma distributions. Like us, they use Taylor expansions to compute the requisite expectations on the shape parameter. However, they provide no further technical details.

### 1.1.3 Applications of gamma distributions and their mixtures

Although our primary aim is to develop and benchmark a variational inference procedure for mixtures of gamma distributions, we are motivated by many important applications of mixtures of gamma distributions. One important application of mixtures of gamma distributions is in rainfall modelling. This is of particular relevance to Australia due to the continent's high rainfall variability (Nicholls, Drosdowsky, and Lavery 1997). As a consequence, between 1961 and 2009, the Australian agricultural industry recorded the highest revenue volatility in the world (Howden, Nelson, and Zammit 2018). Furthermore, climate change is expected to significantly increase rainfall variability, especially during drier seasons (Elouissi et al. 2017; Ayanlade et al. 2018),. By the estimation of Pendergrass et al. (2017), rainfall variability over land will increase by, on average, 4-5% per kelvin. Climate change is also expected to increase the frequency of extreme - or 'black-swan' - weather events. As these events can be disruptive and costly, being able to predict their occurrence in advance is of great importance to industry and government. The

extreme component of the rainfall distribution is known as a 'fat' or heavy tail, and it is a crucial focus of the rainfall literature. Bertolacci et al. (2019) employ a mixture of gamma distributions with time-dependent mixing weights $\pi_{t,k}$ to construct a spatiotemporal model of Australian rainfall. To capture zero-rainfall days, they also add a Dirac delta component $\delta(0)$. In this thesis, we present preliminary results for rainfall modelling at four major Australian dam sites.

Gamma distributions also have applications in medicine. In computational neuroscience, they are used to model neuronal spike timing across cell types for humans and animals (Li et al. 2018). In oncology, Belikov (2017) address the multiple-hit hypothesis of carcinogenesis, which holds that cancer is caused by some fixed number $n$ of successive mutation events that accumulate within a cell over time. If these mutation events are Poisson distributed, then the time taken for a cell to accumulate $n$ mutations and become cancerous must be gamma-distributed. Belikov shows using clinical data that this is indeed the case - that cancer incidence by age closely follows an Erlang distribution, a special case of the gamma distribution for integers. Although they do not consider the mixture case, it stands to reason that different subpopulations (e.g. smokers and non-smokers) would have different rates of cancer incidence. A mixture distribution may, therefore, uncover such insights.

Mohammadi, Salehi-Rad, and Wit (2013) use a mixture of Gaussians to model a M/G/1 queue[5] with a first-come first-served policy. In this problem, customers arrive at independent Poisson-distributed arrival times and wait in queue for service. The authors consider a variation where there is a possibility that the service *fails*, in which case the customer is added to a failed service queue and re-serviced only after all customers in the main queue have been served. Intuitively, it is clear that this results in multiple subpopulations of customers - one for customers whose service succeeds the first time, a second for customers whose service fails once, a third for customers whose service fails twice - and that each successive subpopulation has a longer total waiting time. Since it can be shown that the waiting time until the $n^{\text{th}}$ event of a Poisson process follows a gamma distribution, they find that a mixture of gamma distributions reliably captures the distribution of a customer's total waiting time. This kind of model has wide applications, for instance, in automobile maintenance and customer service.

5. In queueing theory, **M/G/1** is a **M**arkovian (or rando**M**) queue with **G**eneral service times and **1** server (Jones 1989).

## 1.2 Contributions and Organisation of the Thesis

Our main contribution is an algorithm for performing stochastic coordinate-ascent variational inference on a mixture of gamma distributions. We compare our algorithm to a fast C++ implementation of a Gibbs sampler from the `R` package storm the written by UWA PhD candidate Michael Bertolacci (R Development Core Team 2011; Bertolacci et al. 2019). We show that our variational procedure agrees well with the Gibbs sampler, and executes in a fraction of the time. To make fast variational inference on mixtures of gamma distributions available to statistical practitioners without the burden of technical details, we develop a Python package based on the high-performance numerical computation library TensorFlow 2.0 (Dean 2016). This enables practitioners to perform variational inference on a mixture of gammas within a concise API. To illustrate its usage, suppose that we have an array `x` of data and we wish to fit a mixture of gamma distributions with 5 components. A call to `params = mix_gamma_vi(x, K=5)` returns `params` which contains the fitted parameters for a variational approximation to the posterior. The package and its source code are available at `https://github.com/IsaacBreen/MixGammaVI`.

In chapter 2, we introduce the reader to the core concepts of Bayesian inference. This chapter starts with an introduction to Bayesian statistics in section 2.1. Next, we introduce the Metropolis-Hastings algorithm in section 2.2.1, followed by the Gibbs sampler in section 2.2.2. In section 2.3, we discuss variational inference, the mean-field approximation, and derive coordinate-ascent variational inference in the general case.

Chapter 3 begins with section 3.1, an introduction to gamma distributions and their mixtures. Then, we present three algorithms for Bayesian inference on mixtures of gamma distributions: a Gibbs sampler (section 3.2); variational inference with the shape rate parameterisation of the gamma distribution, which we term `VI-1` (section 3.3); and variational inference on the shape-mean parameterisation of the gamma distribution, which we term `VI-2` (section 3.4). section 3.5 compares the conditional posterior distributions used to construct the Gibbs sampler with the variational distributions used to construct `VI-1` and `VI-2` and reveals a deep link between Gibbs sampling and VI. Section 3.6 describes implementation details for `VI-1` and `VI-2`.

Chapter 4 concerns an experimental comparison between the Gibbs sampler, `VI-1`, and `VI-2`. Section 4.1 explains our datasets and methodology, while section 4.2 presents the results. Lastly, chapter 5 concludes this thesis with some final remarks and suggests improvements and extensions for future research.

# Chapter 2

# Bayesian Inference

## 2.1 Introduction to Bayesian Statistics

In the frequentist interpretation of probability, the data $\boldsymbol{x}$ is generated by some fixed process parameterised by $\boldsymbol{\theta} = (\theta_1, ..., \theta_M)$ with the probability distribution function $p(\boldsymbol{x} \mid \boldsymbol{\theta})$. In Bayesian statistics, we relax this notion that the data-generating process is fixed, and this allows us to consider the probability distribution $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ of the parameters $\boldsymbol{\theta}$ conditioned on the samples $\boldsymbol{x}$. This is called the *posterior* probability distribution. Using Bayes' rule, we can write

$$p(\boldsymbol{\theta} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})} \tag{2.1}$$

where $p(\boldsymbol{x} \mid \boldsymbol{\theta})$ is now called the *likelihood* function, and $p(\boldsymbol{\theta})$ is called the *prior* probability of $\boldsymbol{\theta}$. That is, $p(\boldsymbol{\theta})$ quantifies *a priori* knowledge or assumptions that we may have about $\boldsymbol{\theta}$ *before* we observe any samples $\boldsymbol{x}$ from the model. If one has no prior information, then one may represent this by a flat prior $p(\boldsymbol{\theta}) \propto 1$ which is called improper since it does not integrate to one (although such priors are somewhat controversial).

The denominator term $p(\boldsymbol{x})$ is called the marginal likelihood. Since $p(\boldsymbol{x}) = \int_\Theta p(\boldsymbol{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$, we may rewrite eq. (2.1) as

$$p(\boldsymbol{\theta} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{\int_\Theta p(\boldsymbol{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}. \tag{2.2}$$

For complex distributions such as mixtures, the denominator $\int_\Theta p(\boldsymbol{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$ is typically analytically intractable. However, noting that $p(\boldsymbol{x})$ is a normalisation constant that does not depend on $\boldsymbol{\theta}$, we can write

$$p(\boldsymbol{\theta} \mid \boldsymbol{x}) \propto p(\boldsymbol{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{2.3}$$

MCMC and variational inference are techniques for dealing with $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ that avoid the need to evaluate the denominator $\int_\Theta p(\boldsymbol{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$. A common task is to approximate the expectation of some function

$$\mathbb{E}[g(\boldsymbol{\theta}) \mid \boldsymbol{x}] = \int_\Theta g(\boldsymbol{\theta})p(\boldsymbol{\theta} \mid \boldsymbol{x})d\boldsymbol{\theta}. \tag{2.4}$$

In Bayesian statistics, $g(\boldsymbol{\theta})$ often takes the form of the mean, variance, or even the posterior density of $\boldsymbol{\theta}$. The approach of MCMC is to draw $T$ samples $\boldsymbol{\theta}^{[1]}, ..., \boldsymbol{\theta}^{[T]}$ from $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ and calculate the mean

$$\mathbb{E}[g(\boldsymbol{\theta}) \mid \boldsymbol{x}] \approx \frac{1}{M}\sum_{t=1}^{T} g(\boldsymbol{\theta}^{[t]}). \tag{2.5}$$

This is called *Monte Carlo integration*. In contrast, the approach of variational inference is to approximate $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ by some variational distribution $q(\boldsymbol{\theta})$. The integral then becomes

$$\mathbb{E}[g(\boldsymbol{\theta}) \mid \boldsymbol{x}] \approx \int_\Theta g(\boldsymbol{\theta})q(\boldsymbol{\theta})d\boldsymbol{\theta}. \tag{2.6}$$

## 2.2 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a broad class of algorithms for sampling from probability distributions. In Bayesian statistics, MCMC techniques are used for statistical inference (inferring the parameters of a model from data) when it is impractical to do so using analytical techniques. MCMC typically involves sampling from the posterior distribution $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ indirectly. Our research will focus on the application of MCMC statistical inference but, more generally, we can use MCMC to approximate any integral of the form $\int p(x)f(x)dx$ where $p(x)$ is a probability density and $f(x)$ is a function of interest.

### 2.2.1 Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm allows us to sample from any probability distribution whose density we know up to a normalisation constant. This means that we can use eq. (2.3) to draw from eq. (2.2), without having to calculate its intractable denominator.

Suppose, without loss of generality[1], that we wish to sample from some univariate probability distribution with density function $p(\theta)$ for $\theta \in \Theta$. First, we choose a probability

1. In practice, we are almost always concerned vector $\boldsymbol{\theta}$ rather than a scalar $\theta$. Additionally, in Bayesian statistics, we usually want to condition on some data $\boldsymbol{x}$ to obtain the posterior. We first define the general Metropolis-Hastings algorithm, but the extension to these cases is trivial.

density function $q(\theta_1 \mid \theta_2)$ where $\theta_1, \theta_2 \in \Theta$ whose support contains the support of $p(\theta)$. Since we use $q(\theta_1 \mid \theta_2)$ to generate *proposals* for $\theta$, it is called the *proposal distribution*. The Metropolis-Hastings algorithm then proceeds as follows.

**Algorithm 1:** Metropolis-Hastings

1 Initialise $\theta^{[0]}$ arbitrarily (e.g. $\theta^{[0]} \leftarrow 1$)
2 **for** *t in* $(1, ..., T)$ **do**
3  Sample a proposal $\theta'$ from $q(\theta' \mid \theta^{[t-1]})$
4  $r \leftarrow \frac{p(\theta')}{p(\theta^{[t-1]})} \frac{q(\theta^{[t-1]} \mid \theta')}{q(\theta' \mid \theta^{[t-1]})}$
5  Sample $u$ from Uniform$(0, 1)$
6  **if** $u < r$ **then**
7  $\theta^{[t]} \leftarrow \theta'$ (accept)
8  **else**
9  $\theta^{[t]} \leftarrow \theta^{[t-1]}$ (reject)
10  **end**
11 **end**

Now suppose that we also wish to condition on the data $\boldsymbol{x}$. Then the only evaluation of the posterior $p(\theta \mid \boldsymbol{x})$ required by the Metropolis-Hastings algorithm occurs on line 4 where we calculate $\frac{p(\theta' \mid \boldsymbol{x})}{p(\theta^{[t-1]} \mid \boldsymbol{x})}$. But here, problematic normalisation constant $\int_\Theta p(\theta \mid \boldsymbol{x}) p(\boldsymbol{x}) d\theta$ cancels out, and we are left with

$$\frac{p(\theta' \mid \boldsymbol{x})}{p(\theta^{[t]} \mid \boldsymbol{x})} = \frac{p(\boldsymbol{x} \mid \theta') p(\theta')}{p(\boldsymbol{x} \mid \theta^{[t]}) p(\theta^{[t]})}. \tag{2.7}$$

It can be shown that, under certain (very general) conditions, the output sequence $\left(\theta^{[0]}, \theta^{[1]}, ..., \theta^{[T]}\right)$ of Metropolis-Hastings algorithm converges to a sequence that 'looks like' one from the target distribution $p(\theta \mid \boldsymbol{x})$ as $T \to \infty$ (Clarke and Billingsley 1980; Madras and Sezer 2010; Tsvetkov, Hristov, and Angelova-Slavova 2013). In the probability literature, this type of convergence is known as *ergodicity*, and such a chain is called *ergodic*.

Part of the power of the Metropolis-Hastings algorithm stems from the fact that we can choose almost any proposal $q(\theta_1 \mid \theta_2)$ we like and many of its asymptotic properties still hold (so long as the support of $q$ contains the support of $\theta$). Common proposals include a normal or uniform distribution centred at $\theta^{[t-1]}$. Since these distributions are both symmetric, we have $q(\theta^{[t-1]} \mid \theta') = q(\theta' \mid \theta^{[t-1]})$ and can drop these $q$'s out of line 4 in algorithm 1.

The extension of the Metropolis-Hastings algorithm to multivariate distributions such as $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ is trivial: sample $\boldsymbol{\theta'}$ from a multivariate proposal density and accept/reject

jointly. However, when the number of dimensions in the sampling space is high, the Metropolis-Hastings algorithm suffers from the curse of dimensionality. In such cases, a Gibbs sampler is more suitable.

### 2.2.2 Gibbs sampler

If $q(\theta_1 \mid \theta_2)$ is very dissimilar from the posterior $p(\theta)$, the probability of accepting the proposal will be small, and the algorithm may take a long time to converge. If possible, therefore, it is desirable to set $q(\theta_1 \mid \theta_2)$ equal to the posterior. Then the acceptance probability on line 6 is

$$r \leftarrow \frac{p(\theta')}{p(\theta^{[t]})} \frac{p(\theta^{[t]})}{p(\theta')} = 1 \tag{2.8}$$

and the acceptance condition is always satisfied.

For a multivariate distribution, it can be shown that Metropolis-Hastings sampling from $p(\boldsymbol{\theta})$ with $p(\boldsymbol{\theta})$ as the proposal is equivalent to sampling from the conditional posteriors of each of the dimensions sequentially. In doing this, we end up with a special case[2] of the Metropolis-Hastings algorithm called the Gibbs sampler (Geman and Geman 1984), which is also known as Glauber dynamics in physics. Algorithm 2 is the standard form of a Gibbs sampler; it outputs the samples $\boldsymbol{\theta}^{[t]} = (\theta_1^{[t]}, ..., \theta_M^{[t]})$ of $\boldsymbol{\theta}$ for $t = 1, ..., T$.

**Algorithm 2:** Gibbs Sampler

1 Initialise $\boldsymbol{\theta}^{[0]}$
2 **for** *t in* $(1, ..., T)$ **do**
3 | Sample $\theta_1^{[t]}$ from $p(\theta_1 \mid \theta_2^{[t-1]}, \theta_3^{[t-1]}, ..., \theta_M^{[t-1]})$
4 | Sample $\theta_2^{[t]}$ from $p(\theta_2 \mid \theta_1^{[t]}, \theta_3^{[t-1]}, ..., \theta_M^{[t-1]})$
5 | $\vdots$
6 | Sample $\theta_M^{[t]}$ from $p(\theta_M \mid \theta_1^{[t]}, \theta_2^{[t]}, ..., \theta_{M-1}^{[t]})$
7 **end**

When it is difficult to find the conditional posterior for one or more of the variables, we can embed a Metropolis step in the Gibbs sampler by sampling a proposal and then carrying out a standard Metropolis-Hastings acceptance test.

2. While it is usually asserted that the Gibbs sampler is a special case of the Metropolis-Hastings algorithm, this is not *always* the case. See VanDerwerken (2017) for more information.

## 2.3 Variational Inference

Variational Inference frames inference as an optimisation problem. First, we must define Kullback-Leibler divergence (Kullback and Leibler 1951).

**Definition 2.1.** Let $p$ and $q$ be probability distribution functions of $\boldsymbol{\theta}$ defined on the probability space $\boldsymbol{\Theta}$. The Kullback-Leibler (KL) divergence between $p$ and $q$ is

$$\begin{aligned} D_{\mathrm{KL}}(p\|q) &= \mathbb{E}_p\left[\log \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}\right] \\ &= \int_{\boldsymbol{\Theta}} p(\boldsymbol{\theta}') \log \frac{p(\boldsymbol{\theta}')}{q(\boldsymbol{\theta}')} d\boldsymbol{\theta}'. \end{aligned}$$

The KL divergence is not a metric since it neither satisfies the symmetry condition nor the triangle inequality. Nevertheless, it is often intuited as a measure of distance or difference between probability distributions due to some of its metric-like properties: $\mathrm{D}_{\mathrm{KL}}(p\|q) \geq 0$ for all $p$ and $q$, and $\mathrm{D}_{\mathrm{KL}}(p\|q) = 0$ if and only if $p = q$. $\mathrm{D}_{\mathrm{KL}}(p\|q)$ also called the *relative entropy* of $p$ with respect to $q$, reflecting its relationship to differential entropy $-\mathbb{E}_q[\log q(\boldsymbol{\theta})]$ and cross-entropy $-\mathbb{E}_q[\log p(\boldsymbol{\theta})]$.

In (Bayesian) variational inference, we approximate the posterior $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ by some variational distribution $q(\boldsymbol{\theta})$ from a family of candidate distributions $S$. The goal is to minimise the KL divergence between $q$ and $p$:

$$\mathrm{D}_{\mathrm{KL}}(q\|p) = \mathbb{E}_q\left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \boldsymbol{x})}\right].$$

Since $p(\boldsymbol{\theta} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{x})}{p(\boldsymbol{x})}$, the expression above decomposes into

$$\mathrm{D}_{\mathrm{KL}}(q\|p) = \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right] - \mathbb{E}_q\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right] + \mathbb{E}_q\left[\log p(\boldsymbol{x})\right]. \tag{2.9}$$

Let

$$\mathcal{L}(q) = \mathbb{E}_q\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right] - \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right]. \tag{2.10}$$

Note that $\mathbb{E}_q\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right]$ does not depend on $\boldsymbol{\theta}$. Additionally, since $p(\boldsymbol{x})$ does not depend on $\boldsymbol{\theta}$ we have $\mathbb{E}_q\left[\log p(\boldsymbol{x})\right] = \log p(\boldsymbol{x})$. This is the logarithm of the marginal likelihood, also called the *log evidence* for $\boldsymbol{x}$. Then, substituting $\mathcal{L}(q)$ into eq. (2.9) and rearranging gives

$$\log p(\boldsymbol{x}) = \mathrm{D}_{\mathrm{KL}}(q\|p) + \mathcal{L}(q).$$

Since KL divergence is strictly non-negative, we must have

$$\log p(\boldsymbol{x}) \geq \mathcal{L}(q).$$

Hence, $\mathcal{L}(q)$ is called the evidence lower bound (ELBO). Since $\log p(\boldsymbol{x})$ is fixed with respect to $q$, maximising the ELBO with respect $q$ is equivalent to minimising the KL divergence $\mathrm{D}_{\mathrm{KL}}(q\|p)$. We denote this optimal distribution by $q^*$. Thus,

$$\begin{aligned} q^*(\boldsymbol{\theta}) &= \operatorname*{argmin}_{q \in S} \mathrm{D}_{\mathrm{KL}}(q\|p) \\ &= \operatorname*{argmax}_{q \in S} \mathcal{L}(q). \end{aligned}$$

Note that while $q(\boldsymbol{\theta})$ and $q^*(\boldsymbol{\theta})$ are not conditional on $\boldsymbol{x}$, they also not prior distributions; information about $\boldsymbol{x}$ is transferred to them through the optimisation process.

### 2.3.1 The mean-field approximation

We *can* maximise $\mathcal{L}(q)$ with respect to the density function $q$ using techniques from the calculus of variations (thus, *variational* inference). However, in statistical inference the *mean-field approximation* (from mean-field theory in physics) simplifies optimisation problem greatly and avoids the need to resort to the calculus of variations. In the mean-field framework, we assume that $q$ factorises into $M$ distributions $q_i \in S_i$ over disjoint subsets $\boldsymbol{\theta}_i \subseteq \boldsymbol{\theta}$ of the parameters for $i = 1, ..., M$ where $S_i$ is a family of candidate distributions.

$$q(\boldsymbol{\theta}) = \prod_{i=1}^{M} q_i(\boldsymbol{\theta}_i)$$

The mean-field approximation follows directly from assuming independence between $\boldsymbol{\theta}_i$'s. To see this, we can use Bayes' theorem rewrite the exact $q$ as

$$q(\boldsymbol{\theta}) = q(\boldsymbol{\theta}_1 \mid \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, ..., \boldsymbol{\theta}_M) q(\boldsymbol{\theta}_2 \mid \boldsymbol{\theta}_3 ..., \boldsymbol{\theta}_M) \dots q(\boldsymbol{\theta}_M). \tag{2.11}$$

When we assume independence between $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M$, we have $q(\boldsymbol{\theta}_i \mid \boldsymbol{\theta}_{i+1}, ..., \boldsymbol{\theta}_M) = q(\boldsymbol{\theta}_i)$ and eq. (2.11) collapses to the mean-field approximation in eq. (2.12).

We proceed to maximise $\mathcal{L}(q)$ by iteratively optimising each factor $q_i$ while holding the other factors fixed. Denote these locally optimal $q_i$ by $q_i^*$.

$$q_i^*(\boldsymbol{\theta}_i) = \operatorname*{argmax}_{q_i \in S_i} \mathcal{L}(q) \tag{2.12}$$

Letting $\mathbb{E}_{q_j : j \neq i}$ denote the expectation over all $q_j(\boldsymbol{\theta}_j)$ where $j \neq i$, we can rewrite $\mathcal{L}(q)$ as

$$\begin{aligned} \mathcal{L}(q) &= \mathbb{E}_{q_i} \left[ \log \frac{\exp\left( \mathbb{E}_{q_j : j \neq i} \left[ \log p(\boldsymbol{\theta}, \boldsymbol{x}) \right] \right)}{q_i(\boldsymbol{\theta}_i)} \right] - \sum_{j : j \neq i}^{M} \mathbb{E}_{q_j} \left[ \log q_j(\boldsymbol{\theta}_j) \right] \\ &= -\mathrm{D}_{\mathrm{KL}} \left( q_i(\boldsymbol{\theta}_i) \| c^{-1} \exp\left( \mathbb{E}_{q_j : j \neq i} \left[ \log p(\boldsymbol{\theta}, \boldsymbol{x}) \right] \right) \right) - \log c - \sum_{j : j \neq i}^{M} \mathbb{E}_{q_j} \left[ \log q_j(\boldsymbol{\theta}_j) \right] \end{aligned} \tag{2.13}$$

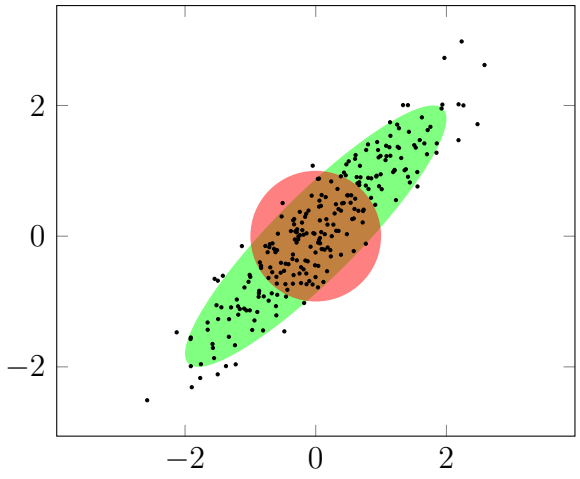



Figure 2.1: Samples from a multivariate normal $(X, Y) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu} = \begin{bmatrix} 0 & 0 \end{bmatrix}$ and $\boldsymbol{\Sigma}$ is non-diagonal. The green region contains 95% of the probability mass of the true likelihood, and the red region is the corresponding mean-field approximation. Clearly, $X$ and $Y$ are highly correlated, but the mean-field approximation, which assumes independence between $X$ and $Y$, fails to capture this dependence.

where $c = \int_{\boldsymbol{\Theta}_i} \exp\left(\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right]\right) d\boldsymbol{\theta}_i$ is a normalisation constant. See appendix A.1 for a full derivation of eq. (2.13). Clearly, the maximum of eq. (2.13) occurs where $q_i(\boldsymbol{\theta}_i) = c^{-1} \exp\left(\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right]\right)$. Therefore,

$$\begin{aligned} q_i^*(\boldsymbol{\theta}_i) &\propto \exp\left(\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right]\right) \\ &\propto \exp\left(\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta} \mid \boldsymbol{x})\right]\right). \end{aligned} \tag{2.14}$$

The procedure of iteratively these updating factors of $q$ is called coordinate-ascent variational inference (CAVI). At each step, $q_i^*$ is the global minimiser of $\mathrm{D}_{\mathrm{KL}}(q\|p)$ were all the other $q_j^*$ are fixed. While CAVI guarantees convergence to a local minimum of $\mathrm{D}_{\mathrm{KL}}(q\|p)$ with respect to $q$, this may not coincide with the global minimum. This makes the choice of initial functions for each $q_i^*$ particularly important. Techniques to avoid getting 'stuck' in a local minimum, such as stochastic variational inference (Hoffman et al. 2013) and proximity variational inference (Altosaar and Blei 2018), are a focus of current research.

In practice, we choose the partitions $\boldsymbol{\theta}_i$ so that the form of $\exp\left(\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right]\right)$ is readily identifiable as a known density such as a normal or gamma density. We end up iteratively updating parameters of these distributions in a loop until some convergence criterion is satisfied.

Variational inference using the KL divergence systematically underestimates the posterior variance. An intuitive reason for this is that $\mathrm{D}_{\mathrm{KL}}(q\|p)$ over-penalises $q$ for *failing* to assign probability mass to where it exists in $p$. It does not, however, 'reward' $q$ for correctly guessing where it exists in $p$. Prying the KL divergence apart helps us intuit

why this is the case.

$$\begin{aligned} \mathrm{D}_{\mathrm{KL}}(q\|p) &= \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right] - \mathbb{E}_q\left[\log p(\boldsymbol{\theta} \mid \boldsymbol{x})\right] \\ &= -h(q) + h(q,p). \end{aligned} \tag{2.15}$$

The term $h(q)$ is the differential entropy of $q$ which measures the expected 'surprise' of the distribution. The flatter the distribution $q$ is, the more surprised we expect to be by an observation from it, and the higher its entropy. Conversely, if a distribution's probability mass is concentrated very tightly around a single point then we don't expect to be at all surprised by an observation from it. Such a distribution has low entropy.

The second term in eq. (2.15) is the differential cross-entropy between $q$ and $p$. By carefully examining $h(q,p) = -\mathbb{E}_q\left[\log p(\boldsymbol{\theta} \mid \boldsymbol{x})\right]$, it should be clear that the distribution $q$ that *minimizes* $h(q,p)$ is one which concentrates all of its probability into a point mass around the mode of $p$. Informally, $q$ does not 'care' about *missing* the probability mass of $p$; $q$ only cares about finding its *mode*.

So, eq. (2.15) comprises of two terms: the *mass-covering* negative entropy term $-h(q)$ which rewards 'flat' distributions of $q$, and the *mode-seeking* cross-entropy term $h(q,p)$ which rewards distributions that are close to the mode of $p$. By minimising $\mathrm{D}_{\mathrm{KL}}(q\|p)$, we implicitly establish a compromise between these objectives; however, neither strongly incentivises $q$ to cover the probability mass of $p$. Variational inference with the KL divergence as the error measure prefers mode-seeking behaviour to mass-covering behaviour and, as a result, systematically underestimates the posterior variance.

This effect is exacerbated by the mean-field approximation, causing $q$ to shrink even further (see fig. 2.1). Tran, Blei, and Airoldi (2015) and Tran (2018) investigate using copulas to recover the dependence between variables. They write $q$ as

$$q(\boldsymbol{\theta}) = c(Q_1(\boldsymbol{\theta}_1), ..., Q_M(\boldsymbol{\theta}_M)) \prod_{i=1}^{M} q_i(\boldsymbol{\theta}_i)$$

where $c(Q_1(\boldsymbol{\theta}_1), ..., Q_M(\boldsymbol{\theta}_M))$ is a multivariate probability density function called a copula that models the dependency between the partitions $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M$. It is a function of the cumulative distribution functions $Q_1, ..., Q_M$ corresponding to the density functions $q_1, ..., q_M$ respectively. Sklar's theorem (Sklar 1959) states such a copula always exists. However, for a mixture of gamma distributions, we show that an orthogonal reparameterization suffices to capture the dependence between the shape and rate parameters, avoiding the need to calculate copulas.

An advantage of framing inference as an optimisation problem is variational inference naturally lends itself to methods from the optimisation literature. One such method is stochastic optimisation which is used extensively in the field of machine learning and allows

algorithms to scale to large datasets (Bottou and Bousquet 2009; Cotter 2013). In artificial neural networks, this takes the form of stochastic gradient descent. As discussed briefly in the introduction, Hoffman et al. (2013) apply the concept of stochastic optimisation to CAVI. They term their algorithm *stochastic variational inference* (SVI) and find that it both vastly decreases computation time and helps avoid local minima. They also show that SVI is equivalent to performing gradient ascent on noisy estimates of the natural gradient, leading to a new, easier way to derive variational update equations by taking gradients instead of expectations. Any CAVI algorithm can be easily converted into SVI by (a) resampling the data at each iteration, (b) using the current global parameters to compute optimal local parameters associated with the subsampled data-points, and then (c) updating the global parameters. However, a SVI algorithm cannot generally be converted back into a CAVI algorithm. For this reason, it is advantageous to work with CAVI instead of SVI wherever possible.

# Chapter 3

# Bayesian Inference on Mixtures of Gamma Distributions

In this chapter, we derive methods for performing Bayesian inference on mixtures of gamma distributions. Section 3.1 introduces the mixture of gamma distributions parameterised by $\alpha$ and $\beta$ and describes some of its properties. Section 3.1.3 explains why an orthogonal reparameterisation of the gamma distribution in terms of $\mu$ and $\alpha$ may (and, as we will show in section 4.2, does) facilitate better variational inference. In section 3.2, we derive a Gibbs sampler for a mixture of gamma distributions parameterised by $\alpha$ and $\beta$. Sections 3.3 and 3.4 contain the major contributions of this thesis. In section 3.3 we derive `VI-1`, a coordinate-ascent variational inference procedure for a mixture of gamma distributions parameterised by $\alpha$ and $\beta$. Similarly, in section 3.4, we derive `VI-2` for a mixture of gamma distributions parameterised by $\mu$ and $\alpha$. Section 3.5 compares the update equations for the Gibbs sampler, `VI-1`, and `VI-2`, revealing a deep link between Gibbs sampling and variational inference. Lastly, section 3.6 describes some additional steps we undertook to implement the VI algorithms.

## 3.1 Introduction to Gamma Mixture Models and Their Properties

### 3.1.1 Gamma distributions

The probability density function of the gamma distribution parameterised by shape $\alpha > 0$ and rate $\beta > 0$ is

$$p(x \mid \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \mathrm{e}^{-\beta x} \tag{3.1}$$

for $x > 0$. Four examples of a gamma distribution are displayed in fig. 3.1.

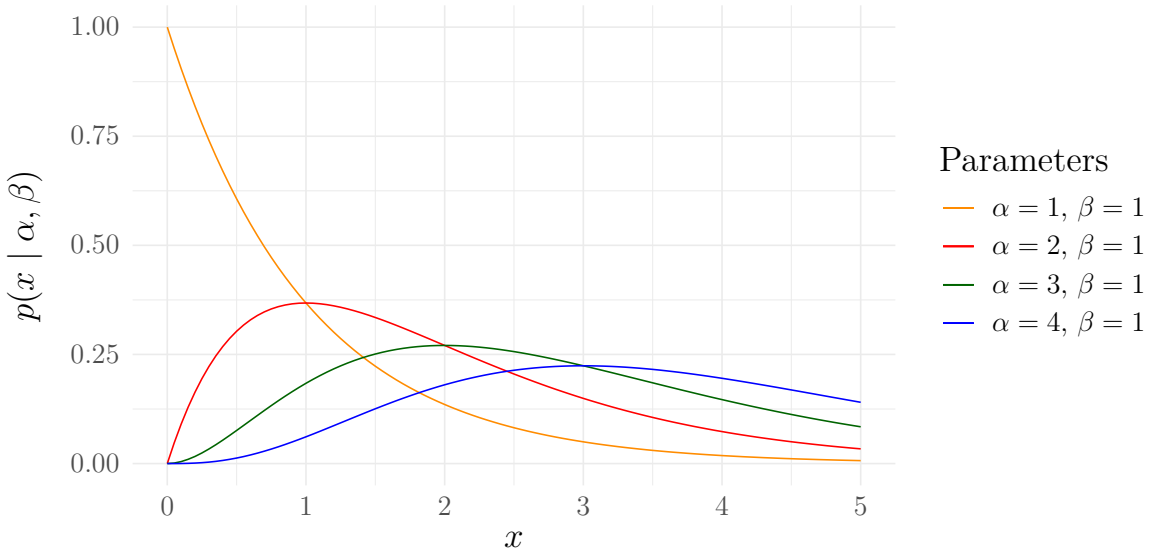



Figure 3.1: The probability density function $p(x \mid \alpha, \beta)$ of four gamma distributions with different shape parameters.

As its name suggests, the shape parameter $\alpha$ controls the shape of the gamma distribution, while the rate parameter $\beta$ simply scales it. The gamma random variable $X \sim \text{Gamma}(\alpha, \beta)$ has mean and variance

$$\mathbb{E}[X] = \frac{\alpha}{\beta} \qquad \text{Var}[X] = \frac{\alpha}{\beta^2}$$

respectively. Later, we will also need the logarithmic mean

$$\mathbb{E}[\log X] = \psi(\alpha) - \log \beta$$

where $\psi(\alpha) = \frac{\partial}{\partial \alpha} \log \Gamma(\alpha)$ is the first derivative of the log gamma function.

Like the normal distribution, the gamma distribution has many important theoretical properties. It has a finite mean and variance for all $\alpha > 0$, and a finite mode for $\alpha > 1$. The normal and gamma distributions are both maximum entropy distributions under their respective constraints. This is desirable for reasons expressed by the principle of maximum entropy (Jaynes 1957) which states that, given some constraints, the best probability distribution is the one that maximises entropy (or, intuitively, leaves the greatest uncertainty) subject to those constraints. The normal distribution is the maximum entropy distribution with finite mean and variance and, analogously, the gamma distribution is the maximum entropy distribution with positive, finite mean and finite logarithmic expectation. Furthermore, in the limit as $\alpha \to \infty$, the gamma distribution converges the normal distribution $\mathcal{N}\left(\frac{\alpha}{\beta}, \frac{\alpha}{\beta^2}\right)$.

### 3.1.2 Mixtures of gamma distributions

The probability density function of a mixture of gamma distributions is

$$\begin{aligned} p(x \mid \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \sum_{k=1}^{K} \pi_k p(x \mid \alpha_k, \beta_k) \\ &= \sum_{k=1}^{K} \pi_k \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} x^{\alpha_k - 1} \mathrm{e}^{-\beta_k x} \end{aligned} \tag{3.2}$$

and the likelihood of $N$ observations $\boldsymbol{x} = (x_1, ..., x_N)$ is the product

$$p(\boldsymbol{x} \mid \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} x_i^{\alpha_k - 1} \mathrm{e}^{-\beta_k x_i}. \tag{3.3}$$

Figure 3.2 shows an example of the density of a mixture of two gamma distributions.

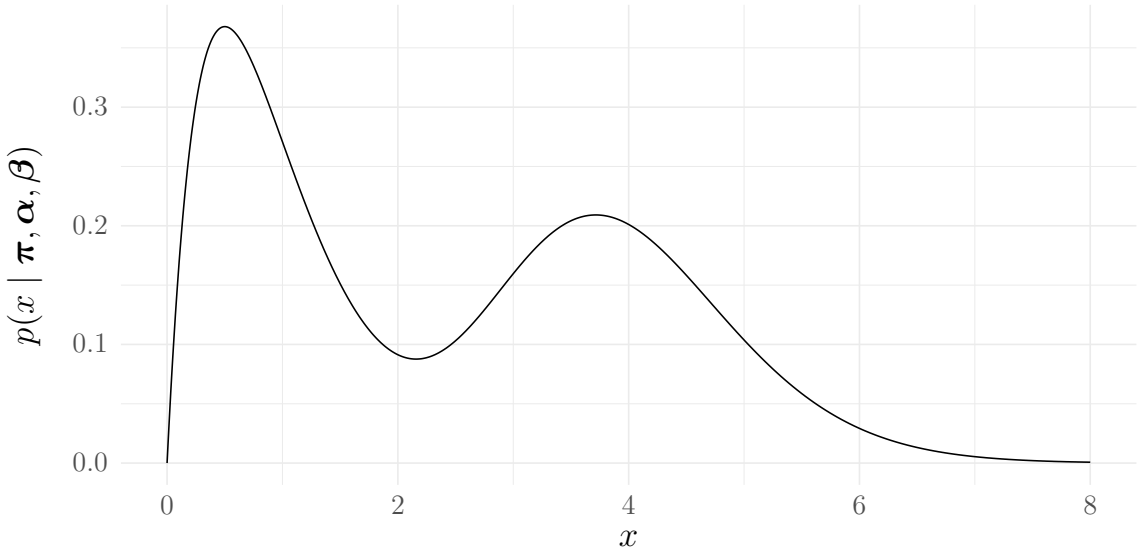



Figure 3.2: The probability density function $p(x \mid \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ of a mixture of two gamma distributions where $\boldsymbol{\pi} = \left(\frac{1}{2}, \frac{1}{2}\right)$, $\boldsymbol{\alpha} = (2, 16)$, and $\boldsymbol{\beta} = (2, 4)$.

The following theorem by DeVore and Lorentz (1993, p. 14) states that for any probability density function $f(x)$ with support contained in $(0, \infty)$, there exists a (infinite) mixture of gamma distributions that approximates $f(x)$ arbitrarily well.

**Theorem 3.1.** *Let $f(x)$ be a probability density on $(0, \infty)$ and define*

$$g_\beta(x) = \sum_{k=1}^{\infty} f\left(\frac{k}{\beta}\right) \frac{\beta^k}{\Gamma(k)} x^k \mathrm{e}^{-\beta x}.$$

*Then*

$$f(x) = \lim_{\beta \to \infty} g_\beta(x).$$

So, any density $f(x)$ can be approximated arbitrarily well by an infinite mixture of gamma distributions with shapes $k+1$ and mixture weights $f\left(\frac{k}{\beta}\right)$ for $k \in \mathbb{N}$ by selecting an appropriate value for the rate $\beta$.

Since we wish to perform Bayesian inference, we must define priors on our parameters. We use

$$\begin{aligned} p(\boldsymbol{\pi}) &= \text{Dirichlet}(\boldsymbol{\omega}), \\ p(\alpha_k) &\propto \frac{\mathrm{e}^{r_k \alpha_k}}{\Gamma(\alpha_k)^{s_k}}, \\ p(\beta_k) &= \text{Gamma}(c_k, d_k) \end{aligned}$$

where $c_k, d_k, r_k, s_k > 0$, $\boldsymbol{\omega} = (\omega_1, ..., \omega_K)$, and $\omega_k > 0$.

A *conjugate* prior is one which induces a posterior from the same family. The priors on $\boldsymbol{\pi}$ and $\beta_k$ are conjugate. The prior on $\alpha_k$ is a special case of Damsleth's (1975) conjugate conditional prior $p(\alpha_k \mid \beta_k) \propto \mathrm{e}^{r_k} \alpha_k \left(\beta_k^{v\alpha_k} \Gamma(\alpha_k)^{s_k}\right)^{-1}$ where $v = 0$. In practice, we usually set the same parameters $c_k, d_k, r_k, s_k$ for all $k$ rather than tuning the prior for each component manually.

To make inference on mixture models tractable, it is usually necessary to define a *latent* variable that assigns each observation $x_i$ to one of the $K$ components. Recall that there are $N$ samples in $\boldsymbol{x}$. We name this latent variable the *component assignment* variable and denote it by $z_i \in 1, .., K$ for $i = 1, ..., N$. Also, for notational convenience denote the number of data assigned to each component by $n_k = \sum_{i=1}^{N} \mathbb{I}_{z_i=k}$ where we have used the indicator random variable

$$\mathbb{I}_{z_i=k} = \begin{cases} 1 & \text{if } z_i = k \\ 0 & \text{if } z_i \neq k \end{cases}.$$

Additionally, let $\boldsymbol{n} = (n_1, ..., n_K)$. Conditioned on the latent variables $\boldsymbol{z}$, the likelihood collapses to

$$\begin{aligned} p(\boldsymbol{x} \mid \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}) &= \prod_{i=1}^{N} \sum_{k=1}^{K} \mathbb{I}_{z_i=k} \, \pi_k \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} x_i^{\alpha_k - 1} \mathrm{e}^{-\beta_k x_i} \\ &= \prod_{k=1}^{K} \pi_k^{n_k} \frac{\beta_k^{n_k \alpha_k}}{\Gamma(\alpha_k)^{n_k}} \prod_{i:z_i=k} x_i^{\alpha_k - 1} \mathrm{e}^{-\beta_k x_i}. \end{aligned} \tag{3.4}$$

### 3.1.3 Fisher information and orthogonal parameterisation

Recall that in order to make variational inference tractable, we usually need to assume independence between some partitions of the parameters in the approximating distribution $q$. For the gamma distribution, we must fully factorise the variational distribution and

assume that $\alpha$ and $\beta$ are independent in $q$. For a *mixture* of gamma distributions, since the mixture weights $\boldsymbol{\pi}$ total one we have to consider them jointly. Thus, to make the VI problem tractable for a mixture of gamma distributions, we assume that the parameter space partitions into $\boldsymbol{\pi}$, $\alpha_k$, $\beta_k$, and $z_i$ for $k = 1, ..., K$ and $i = 1, ..., N$.

However, as we will now argue in the limit as $N \to \infty$, the parameters $\alpha_k$ and $\beta_k$ are strongly dependent in the posterior distribution. Therefore, since mean-field VI assumes independence between $\alpha_k$ and $\beta_k$, it will, by design, be unable to capture this dependency. To remedy this, we propose using an *orthogonal* reparameterisation of the gamma distribution in terms of parameters that become independent in the posterior as $N \to \infty$. The following asymptotic argument only holds in this limit, but section 4.2 provides experimental evidence that it is true even when $N$ is a large value such as 4000. While we only consider a single gamma distribution, the extension to a mixture of gamma distributions is trivial once we condition on the component assignment variables $\boldsymbol{z}$.

Under certain regularity conditions, the Fisher information matrix of some parameters $\boldsymbol{\theta}$ can be written as

$$[\mathcal{I}(\boldsymbol{\theta})]_{i,j} = -\,\mathbb{E}\left[\frac{\partial^2}{\partial\theta_i \partial\theta_j} \log p(\boldsymbol{x}) \;\middle|\; \boldsymbol{\theta}\right]$$

where the expectation is taken over the data $\boldsymbol{x}$. Let $\boldsymbol{\theta}_0$ denote the true value of $\boldsymbol{\theta}$ and let $\hat{\boldsymbol{\theta}}$ denote its maximum likelihood estimator. The principle of asymptotic normality of maximum likelihood estimators states that

$$\sqrt{N}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0) \xrightarrow{d} \mathcal{N}\left(0, \frac{1}{N}\mathcal{I}^{-1}(\boldsymbol{\theta}_0)\right).$$

Arellano (2016) note that, for large $N$, this implies the posterior becomes close to

$$\boldsymbol{\theta} \mid \boldsymbol{x} \sim \mathcal{N}\left(\hat{\boldsymbol{\theta}}, \mathcal{I}^{-1}(\boldsymbol{\theta}_0)\right). \tag{3.5}$$

We can reach a similar result by taking a second-order Taylor approximation to the log-posterior[1] $\ell(\boldsymbol{\theta}) = \log p(\boldsymbol{\theta} \mid \boldsymbol{x})$ around its mode $\tilde{\boldsymbol{\theta}}$.

$$\begin{aligned}\log p(\boldsymbol{\theta} \mid \boldsymbol{x}) &\approx \ell(\tilde{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})\ell'(\tilde{\boldsymbol{\theta}}) + \frac{1}{2}(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^2\ell''(\tilde{\boldsymbol{\theta}}) \\ &= \ell(\tilde{\boldsymbol{\theta}}) + \frac{1}{2}(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^2\mathcal{J}(\tilde{\boldsymbol{\theta}})\end{aligned} \tag{3.6}$$

where $\mathcal{J}(\boldsymbol{\theta}) = \ell''(\boldsymbol{\theta})$ is the *observed information* at $\boldsymbol{\theta}$. It is related to the Fisher information by $\mathcal{I}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}}[\mathcal{J}(\boldsymbol{\theta})]$. Taking the exponential of eq. (3.6), we get

$$\begin{aligned}p(\boldsymbol{\theta} \mid \boldsymbol{x}) &= \mathrm{e}^{\ell(\tilde{\boldsymbol{\theta}})} \exp\left(\frac{1}{2}(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^2\mathcal{J}(\tilde{\boldsymbol{\theta}})\right) \\ &\approx \mathcal{N}\left(\tilde{\boldsymbol{\theta}}, \mathcal{J}^{-1}(\tilde{\boldsymbol{\theta}})\right).\end{aligned} \tag{3.7}$$

1. $\ell(\boldsymbol{\theta})$ is known as the *log-likelihood* in frequentist statistics.

When $\boldsymbol{\theta}$ are the parameters $(\alpha, \beta)$ of a gamma distribution, the observed information does not depend on the data $\boldsymbol{x}$. Thus, $\mathcal{J}(\hat{\boldsymbol{\theta}}) = \mathcal{I}(\hat{\boldsymbol{\theta}})$ and eq. (3.7) is identical to eq. (3.5) except that the Fisher information is evaluated $\hat{\boldsymbol{\theta}}$ in the former and $\boldsymbol{\theta}_0$ in the latter.

The inverse Fisher information matrix for a Gamma distribution parameterised by $\alpha$ and $\beta$ is

$$\mathcal{I}(\alpha, \beta) = N \begin{bmatrix} \psi^{(1)}(\alpha) & -\frac{1}{\beta} \\ -\frac{1}{\beta} & \frac{\alpha}{\beta^2} \end{bmatrix}$$

This suggests that, in the limit as $N \to \infty$, the parameters $\alpha$ and $\beta$ have non-negative covariance. This is well-supported by the results in fig. 4.6. In fact, as $\alpha = \beta$ goes to infinity, the covariance between $\alpha$ and $\beta$ approaches 1. This has dire implications for the mean-field approximation since it, by definition, cannot capture the dependence between $\alpha$ and $\beta$.

This motivates an *orthogonal* parameterisation of the gamma distribution for which the Fisher information matrix is diagonal. Let $\mu = \frac{\alpha}{\beta}$ and reparameterize the Gamma distribution by $\mu$ and $\alpha$. The new probability density function is

$$p(x \mid \mu, \alpha) = \frac{\left(\frac{\alpha}{\mu}\right)^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} \mathrm{e}^{-\frac{\alpha}{\mu}x}$$

and the Fisher information matrix is now

$$\mathcal{I}(\alpha, \mu) = N \begin{bmatrix} \psi^{(1)}(\alpha) - \frac{1}{\alpha} & 0 \\ 0 & \frac{\alpha}{\mu^2} \end{bmatrix}.$$

Since the Fisher information matrix is diagonal, its inverse must be diagonal too. Therefore, the covariance term in eq. (3.5) is diagonal too and, in the limit as $N \to \infty$, the joint posterior distribution $p(\mu, \alpha \mid \boldsymbol{x})$ approaches a multivariate normal for which $\mu$ and $\alpha$ are independent. This suggests that, for large $N$, the posterior of a gamma distribution parameterised by $\mu$ and $\alpha$ will be much better preserved under the mean-field approximation than if it were parameterised by $\alpha$ and $\beta$. A similar argument which is supported by the results in section 4.2 applies to a mixture of gamma distributions.

## 3.2 Gibbs sampler for a mixture of gamma distributions

In this section, we derive a Gibbs sampler from gamma mixture models. This samples from the posterior distribution $p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z} \mid \boldsymbol{x})$ by sampling the parameters conditionally for $t = 1, ..., T$ iterations.

1. $\boldsymbol{\pi}^{[t]} \sim p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}^{[t-1]}, \boldsymbol{\beta}^{[t-1]}, \boldsymbol{z}^{[t-1]}, \boldsymbol{x})$
2. $\alpha_j^{[t]} \sim p(\alpha_j \mid \boldsymbol{\pi}^{[t]}, \boldsymbol{\alpha}_{-j}^{[t]}, \boldsymbol{\beta}^{[t-1]}, \boldsymbol{z}^{[t-1]}, \boldsymbol{x})$ for $j = 1, ..., K$.
3. $\beta_j^{[t]} \sim p(\beta_j \mid \boldsymbol{\pi}^{[t]}, \boldsymbol{\alpha}^{[t]}, \boldsymbol{\beta}_{-j}^{[t]}, \boldsymbol{z}^{[t-1]}, \boldsymbol{x})$ for $j = 1, ..., K$.
4. $z_i^{[t]} \sim p(z_i \mid \boldsymbol{\pi}^{[t]}, \boldsymbol{\alpha}^{[t]}, \boldsymbol{\beta}^{[t]}, \boldsymbol{z}_{-i}^{[t]}, \boldsymbol{x})$ for $i = 1, ..., N$

Note that $\boldsymbol{\pi}$ must be sampled jointly. A detailed algorithm is given in algorithm 3.

We now derive these conditional posterior distributions. Using Bayes' rule, the joint conditional posterior of $\boldsymbol{\pi}$ is found to be

$$\begin{aligned} p(\boldsymbol{\pi} \mid \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}) &\propto p(\boldsymbol{x}, \mid \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}) p(\boldsymbol{\pi}) \\ &\propto \left( \prod_{i=1}^{N} \pi_{z_i} \right) \left( \prod_{k=1}^{K} \pi^{\omega_k - 1} \right) \\ &\propto \prod_{k=1}^{K} \pi_k^{n_k + \omega_k - 1} \\ &\propto \text{Dirichlet}(\boldsymbol{n} + \boldsymbol{\omega}) \end{aligned}$$

where we have recognised that the third line has the form of Dirichlet distribution up to a normalising constant - this is an invaluable trick in Bayesian statistics.

Next, we find the conditional posterior of $\alpha_k$. To avoid confusion with the product over $k = 1, ..., K$, we will instead denote $\alpha_k$ with the subscript $j$, but be aware that $\alpha_j$ and $\alpha_k$ are interchangeable. Let $\boldsymbol{\alpha}_{-j}$ denote the vector that contains all $\alpha_k$ except for $k = j$. Then

$$\begin{aligned} p(\alpha_j \mid \boldsymbol{\pi}, \boldsymbol{x}, \boldsymbol{\alpha}_{-j}, \boldsymbol{\beta}, \boldsymbol{z}) &\propto p(\alpha_j \mid \boldsymbol{x}, \beta_j, \boldsymbol{z}) \\ &\propto p(\alpha_j) p(\boldsymbol{x} \mid \alpha_j, \beta_j, \boldsymbol{z}) \\ &\propto \frac{\mathrm{e}^{r_j \alpha_j} \beta_j^{n_j \alpha_j}}{\Gamma(\alpha_j)^{n_j + s_j}} \prod_{i:z_i=j} x_i^{\alpha_j - 1} \end{aligned} \tag{3.8}$$

This density does not have a standard form. However, since it is log-concave, we can sample from it using adaptive rejection sampling (ARS)(Gilks and Wild 1992). See appendix A.4 for details.

The conditional posterior for $\beta_j$ is

$$\begin{aligned} p(\beta_j \mid \boldsymbol{\pi}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}_{-j}, \boldsymbol{z}) &\propto p(\beta_j \mid \boldsymbol{x}, \alpha_j, \boldsymbol{z}) \\ &\propto p(\beta_j) p(\boldsymbol{x} \mid \beta_j, \alpha_j, \boldsymbol{z}) \\ &\propto \beta_j^{c_j + n_j \alpha_j - 1} \mathrm{e}^{-\beta_j (d_j + n_j \bar{x}_j)} \\ &\propto \text{Gamma}(c_j + n_j \alpha_j, d_j + n_j \bar{x}_j) \end{aligned}$$

where $\bar{x}_j = \frac{1}{n_j} \sum_{n:z_i=j} x_i$.

Finally, we need to update the component assignment variables $z_i$ for $i = 1, .., N$. These are discrete multinomial random variables with the posterior

$$\begin{aligned} p(z_i = j \mid \boldsymbol{\pi}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}) &\propto p(z_i = j) p(x_i \mid \boldsymbol{x}_{-i}, \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}) \\ &\propto \pi_j p(x_i \mid \alpha_{z_i}, \beta_{z_i}) \\ &\propto \pi_j \frac{\beta_j^{\alpha_j}}{\Gamma(\alpha_j)} x_i^{\alpha_j - 1} \mathrm{e}^{-\beta_j x_i} \end{aligned}$$

We obtain full distribution for each $z_i$ by normalising the above expression:

$$\begin{aligned} p(z_i = j \mid \boldsymbol{\pi}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}) &= \frac{p(z_i = j \mid \boldsymbol{\pi}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}_i)}{\sum_{k=1}^{K} p(z_i = k \mid \boldsymbol{\pi}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}_i)} \\ &= \frac{\pi_j \frac{\beta_j^{\alpha_j}}{\Gamma(\alpha_j)} x_i^{\alpha_j - 1} \mathrm{e}^{-\beta_j x_i}}{\sum_{k=1}^{K} \pi_k \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} x_i^{\alpha_k - 1} \mathrm{e}^{-\beta_k x_i}} \\ &= p_{i,j} \end{aligned}$$

where we have defined $p_{i,j}$ for notational convenience. So, the conditional posterior is $z_i \mid \boldsymbol{\pi}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}_{-i} \sim \text{Categorical}(p_{i,1}, ..., p_{i,K})$. While this step is necessary to make the Gibbs sampler tractable, it is also very computationally expensive.

**Algorithm 3:** Gibbs sampler for mixture of gammas

**for** $t$ *in* $(1, ..., T)$ **do**
- Sample $z_i^{[t]}$ from Categorical$(p_{i,1}^{[t-1]}, ..., p_{i,K}^{[t-1]})$ for $i = 1, ..., N$
- Sample $\boldsymbol{\pi}^{[t]}$ from Dirichlet$(\boldsymbol{n}^{[t]} + \boldsymbol{\omega})$
- Sample $\alpha_j^{[t]}$ from $\frac{\mathrm{e}^{r_j \alpha_j} \left(\beta_j^{[t]}\right)^{n_j^{[t]} \alpha_j}}{\Gamma(\alpha_j)^{n_j^{[t]} + s_j}} \prod_{i:z_i^{[t]}=j} x_i^{\alpha_j - 1}$ using ARS for $j = 1, ..., K$
- Sample $\beta_j^{[t]}$ from Gamma$(c_j + n_j^{[t]} \alpha_j^{[t]}, d_j + n_j^{[t]} \bar{x}_j)$ for $j = 1, ..., K$

**end**

We use a high-performance Gibbs sampler from the `R` package Storm developed in C++ by UWA graduate Michael Bertolacci in contribution to his PhD candidature (R Development Core Team 2011; Bertolacci et al. 2019).

## 3.3 Variational Mixture of Gamma Distributions Parameterised by $\alpha$ and $\beta$

Recall that in eq. (2.14) we derived the optimal mean-field variational distributions for the partitioning of the parameter space $\boldsymbol{\theta}$ into $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M$. For a mixture of gamma distributions

we partition the parameter space into $\boldsymbol{\pi}$, $\alpha_k$ and $\beta_k$ for $k = 1, ..., K$, and $z_i$ for $i = 1, ..., N$. This implies that these partitions are independent in the approximating distribution $q$. The optimal mean-field variational distributions are then

$$q^*(\boldsymbol{\pi}) \propto \exp\left(\mathbb{E}_{q_{-\boldsymbol{\pi}}}\left[\log p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z} \mid \boldsymbol{x})\right]\right), \tag{3.9}$$

$$q^*(\alpha_k) \propto \exp\left(\mathbb{E}_{q_{-\alpha_k}}\left[\log p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z} \mid \boldsymbol{x})\right]\right), \tag{3.10}$$

$$q^*(\beta_k) \propto \exp\left(\mathbb{E}_{q_{-\beta_k}}\left[\log p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z} \mid \boldsymbol{x})\right]\right), \tag{3.11}$$

$$q^*(z_i = k) \propto \exp\left(\mathbb{E}_{q_{-z_i}}\left[\log p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z} \mid \boldsymbol{x})\right]\right) \tag{3.12}$$

and these factorise the full variational distribution

$$q(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}) = q^*(\boldsymbol{\pi}) \prod_{k=1}^{K} q^*(\alpha_j) q^*(\beta_j)$$

which approximates the true posterior $p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z} \mid \boldsymbol{x})$. Similarly as in section 2.3, $\mathbb{E}_{q_{-\boldsymbol{\pi}}}$ denotes the expectation over all dimensions of the variational distribution $q$ except for $\boldsymbol{\pi}$, and likewise for the other parameters. Thus, to derive these update equations we first need the complete posterior $p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z} \mid \boldsymbol{x})$. First, recall that the probability density for a mixture of gamma distributions (eq. (3.4)) conditioned on the component assignment variables $\boldsymbol{z}$ is given by

$$p(\boldsymbol{x} \mid \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}) = \prod_{k=1}^{K} \pi_k^{n_k} \frac{\beta_k^{n_k \alpha_k}}{\Gamma(\alpha_k)^{n_k}} \prod_{i:z_i=k} x_i^{\alpha_k - 1} \mathrm{e}^{-\beta_k x_i}.$$

Therefore, the posterior density is

$$\begin{aligned} p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z} \mid \boldsymbol{x}) &\propto p(\boldsymbol{\pi}) \left( \prod_{k=1}^{K} p(\alpha_k) p(\beta_k) \right) p(\boldsymbol{x} \mid \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}) \\ &= \prod_{k=1}^{K} \pi_k^{\omega_k + n_k - 1} \frac{\beta_k^{n_k \alpha_k + c_k - 1}}{\Gamma(\alpha_k)^{n_k + s_k}} \mathrm{e}^{r_k \alpha_k - d_k \beta_k} \prod_{\{i:z_i=k\}} x_i^{\alpha_k - 1} \mathrm{e}^{-\beta_k x_i}. \end{aligned}$$

For convenience, we work in exp-log form.

$$\begin{aligned} p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z} \mid \boldsymbol{x}) \propto \exp\Bigg( \sum_{k=1}^{K} \Bigg( & (\omega_k + n_k - 1) \log \pi_k + (n_k \alpha_k + c_k - 1) \log \beta_k \\ & - (n_k + s_k) \log \Gamma(\alpha_k) + r_k \alpha_k - d_k \beta_k \\ & + \sum_{i=1}^{N} \mathbb{I}_{z_i=k} \left( (\alpha_k - 1) \log x_i - \beta_k x_i \right) \Bigg) \Bigg) \end{aligned}$$

Here, we define $q_{i,k} = \mathbb{E}_q[\mathbb{I}_{z_i=k}] = q^*(z_i = k)$ and $n_k = \sum_{i=1}^{N} \mathbb{I}_{z_i=k}$. As a simple consequence, we have $\mathbb{E}_q[n_k] = \sum_{i=1}^{N} \mathbb{E}_q[\mathbb{I}_{z_i=k}] = \sum_{i=1}^{N} q_{i,k}$. We now derive the variational update equations.

### 3.3.1 Update equations

We begin by calculating $q^*(\boldsymbol{\pi})$. Applying eq. (3.9) yields

$$
\begin{aligned}
q^*(\boldsymbol{\pi}) &\propto \exp \mathbb{E}_{q_{-\pi}} \left[\log p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \boldsymbol{z} \mid \boldsymbol{x})\right] \\
&\propto \exp \mathbb{E}_{q_{-\pi}} \left[ \sum_{k=1}^{K} \left( \begin{aligned} &(\omega_k + n_k - 1) \log \pi_k + (n_k \alpha_k + c_k - 1) \log \beta_k \\ &\quad - (n_k + s_k) \log \Gamma(\alpha_k) + r_k \alpha_k - d_k \beta_k \\ &\quad + \sum_{i=1}^{N} \mathbb{I}_{z_i = k} \left( (\alpha_k - 1) \log x_i - \beta_k x_i \right) \end{aligned} \right) \right] \\
&\propto \exp \left( \sum_{k=1}^{K} (\omega_k + \mathbb{E}_q[n_k] - 1) \log \pi_k \right) \\
&\propto \prod_{k=1}^{K} \exp \left( \log \pi_k \left( \omega_k + \sum_{i=1}^{N} q_{i,k} - 1 \right) \right) \\
&\propto \prod_{k=1}^{K} \pi_k^{\omega_k + \sum_{i=1}^{N} q_{i,k} - 1} \\
&\propto \text{Dirichlet}\left(\zeta_1, ..., \zeta_K\right)
\end{aligned}
$$

where $\zeta_k = \omega_k + \sum_{i=1}^{N} q_{i,k}$. Notice that this is very close in form to the corresponding step of the Gibbs sampler in chapter 3 where the conditional posterior was Dirichlet distributed with parameters $\omega_k + \sum_{i=1}^{N} \mathbb{I}_{z_i=k}$ for $k = 1, ..., K$. This is the first of many parallels we will see between Gibbs sampling and variational inference.

Next, we derive the update equation for each $\beta_j$ with $j = 1, ..., K$. Applying eq. (3.11),

$$
\begin{aligned}
q^*(\beta_j) &= \exp \mathbb{E}_{q_{-\beta_j}} \left[\log p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \boldsymbol{z} \mid \boldsymbol{x})\right] \\
&\propto \exp \left( (\mathbb{E}_q[n_j] \, \mathbb{E}_q[\alpha_j] + c_j - 1) \log \beta_j - d_j \beta_j - \sum_{i=1}^{N} \mathbb{E}_q[\mathbb{I}_{z_i=j}] \beta_j x_i \right) \\
&\propto \beta_1^{\mathbb{E}_q[\alpha_j] \sum_{i=1}^{N} q_{i,j} + c_j - 1} \exp \left( -\beta_j \left( d_j + \sum_{i=1}^{N} q_{i,j} x_i \right) \right) \\
&= \text{Gamma}\left(\gamma_j, \lambda_j\right)
\end{aligned}
$$

where $\gamma_j = c_j + \mathbb{E}_q[\alpha_j] \sum_{i=1}^{N} q_{i,j}$ and $\lambda_j = d_j + \sum_{i=1}^{N} q_{i,j} x_i$. We hope to find $\mathbb{E}_q[\alpha_j]$ later on once we have derived a variational distribution for $\alpha_j$. This distribution will in turn depend on expectations of some functions $\beta_j$. Thus, we end up with a circular series of update equations that are evaluated one-after-another in a loop.

Next, the optimal variational distribution for $\alpha_j$ is

$$\begin{aligned} q^*(\alpha_j) &= \exp \mathbb{E}_{q_{-\alpha_j}} \left[\log p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \boldsymbol{z} \mid \boldsymbol{x})\right] \\ &\propto \exp\left(\alpha_j \, \mathbb{E}_q[n_j] \, \mathbb{E}_q[\log \beta_j] - (\mathbb{E}_q[n_j] + s_j) \log \Gamma(\alpha_j) + r_j \alpha_k + \sum_{i=1}^{N} \mathbb{E}_q[\mathbb{I}_{z_i=j}](\alpha_j - 1) \log x_i\right) \\ &\propto \exp\left(\alpha_j \left((\psi(\gamma_j) - \log(\lambda_j)) \sum_{i=1}^{N} q_{i,j} + \sum_{i=1}^{N} q_{i,j} \log x_i + r_j\right) - \left(\sum_{i=1}^{N} q_{i,j} + s_j\right) \log \Gamma(\alpha_j)\right). \end{aligned} \tag{3.13}$$

Due to $\log \Gamma(\alpha_j)$ term, we cannot identify this expression with any known distribution. This is problematic because, in order to calculate $\mathbb{E}_q[\alpha_j]$, we need the variational distribution of $\beta_j$. Furthermore, we will show later that the optimal variational distribution of $z_i$ requires us to know both $\mathbb{E}_q[\alpha_j]$ and $\mathbb{E}_q[\log \Gamma(\alpha_j)]$. However, while eq. (3.13) is not itself a known density, a second-order Taylor expansion of $\log \Gamma(\alpha_j)$ around some point $\hat{\alpha}_j$ (which we will derive later) *does* yield a normal distribution. The expansion is

$$\begin{aligned} \log \Gamma(\alpha_j) &= \log \Gamma(\hat{\alpha}_j) + \sum_{n=0}^{\infty} \frac{\psi^{(n)}(\hat{\alpha}_j)}{n!} (\alpha_j - \hat{\alpha}_j)^{n+1} \\ &\approx \log \Gamma(\hat{\alpha}_j) + (\alpha_1 - \hat{\alpha}_j)\psi(\hat{\alpha}_j) + \frac{1}{2}(\alpha_j - \hat{\alpha}_j)^2 \psi^{(1)}(\hat{\alpha}_j) \end{aligned} \tag{3.14}$$

where $\psi^{(n)}(x)$ is the polygamma function defined as $\psi^{(n)}(x) = \frac{d^n}{dx^n}\psi(x) = \frac{d^{n+1}}{dx^{n+1}} \log \Gamma(x)$. Substituting eq. (3.14) into eq. (3.13), we get the following approximation (where $\overset{\sim}{\propto}$ denotes 'proportional to approximately').

$$\begin{aligned} q^*(\alpha_j) \overset{\sim}{\propto} \exp\Bigg(&\alpha_j \left((\psi(\gamma_j) - \log(\lambda_j)) \sum_{i=1}^{N} q_{i,j} + \sum_{i=1}^{N} q_{i,j} \log x_i + r_j\right) \\ &- \left(\sum_{i=1}^{N} q_{i,j} + s_j\right) \left(\log \Gamma(\hat{\alpha}_j) + (\alpha_1 - \hat{\alpha}_j)\psi(\hat{\alpha}_j) + \frac{1}{2}(\alpha_j - \hat{\alpha}_j)^2 \psi^{(1)}(\hat{\alpha}_j)\right)\Bigg) \end{aligned} \tag{3.15}$$

Notice that this expression consists of the terms $\alpha_j$ and $\alpha_j^2$ multiplied by some constants that do not depend on $\alpha_j$. It is clear, therefore, that eq. (3.15) must be proportional to the density of some normal distribution $\mathcal{N}(\hat{\alpha}_j, \sigma_j^2)$. The parameters $\hat{\alpha}_j$ and $\sigma_j^2$ can be found by rearranging eq. (3.15). However, in our experiments we found that this approximation was not accurate enough to facilitate accurate convergence. Instead, we set $q(\alpha_j) = \mathcal{N}(\hat{\alpha}_j, \sigma_j^2)$ and seek values for $\hat{\alpha}_j$ and $\sigma_j$ which minimise the ELBO in eq. (2.10) directly. This is known in the statistical literature as the *delta method*.

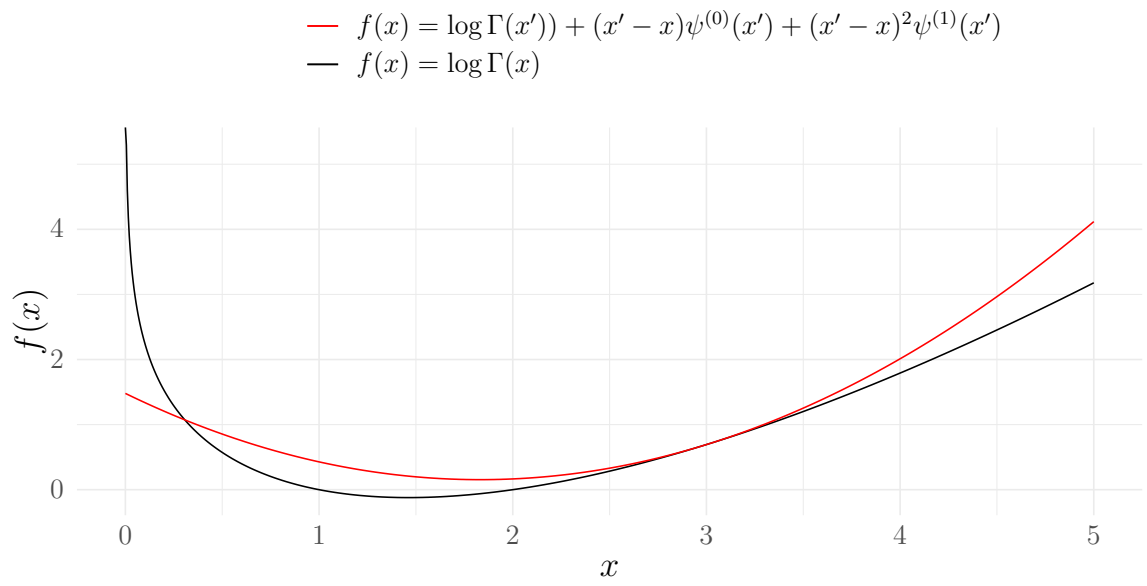



Figure 3.3: The second-order Taylor approximation (red) to the log gamma function (black) around $x' = 3$.

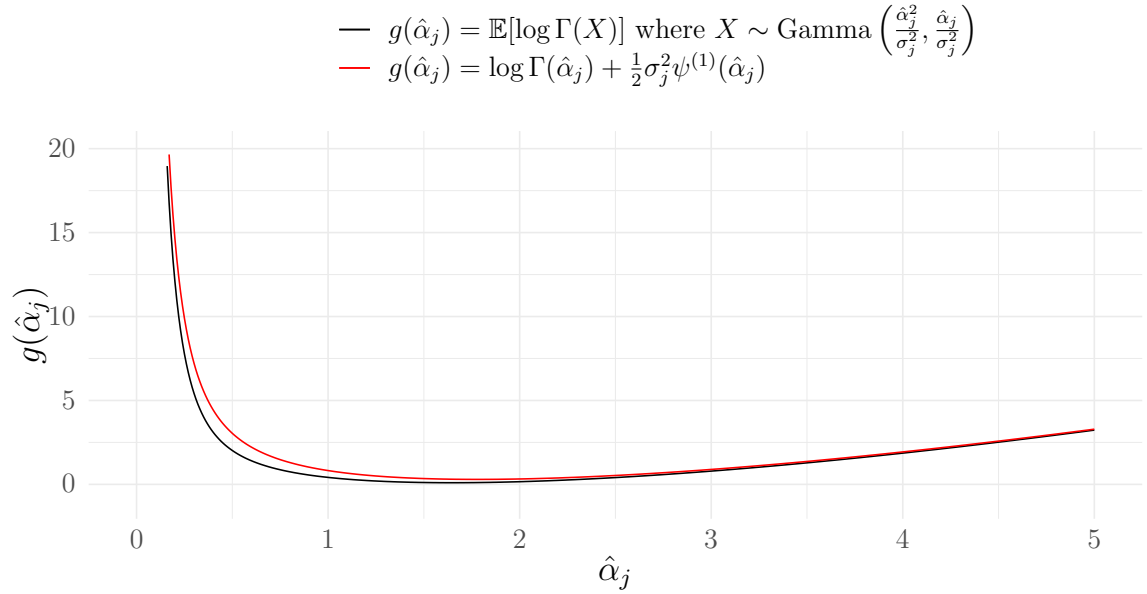



Figure 3.4: The expectation of $\log\Gamma(X)$ (black) where $X$ is gamma distributed with mean $\hat{\alpha}_j$ and variance $\sigma_j^2$ together with the expectation of the second-order Taylor expansion of $\mathbb{E}[\log\Gamma(\alpha_j)]$ around $\hat{\alpha}_j$ (red) where $\alpha_j$ is normally distributed with the same mean and variance.

$$
\begin{aligned}
(\hat{\alpha}_j, \sigma_j) &= \underset{(\hat{\alpha}_j,\sigma_j)\in\mathbb{R}_+^2}{\operatorname{argmin}} \; \mathbb{E}_q\left[\log\left(\frac{p(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\pi},\boldsymbol{z}\mid\boldsymbol{x})}{q(\alpha_j)}\right)\right] \\
&= \underset{(\hat{\alpha}_j,\sigma_j)\in\mathbb{R}_+^2}{\operatorname{argmin}} \Bigg(\alpha_j\left(\mathbb{E}_q[\log\beta\,\mathbb{E}_q[n_k] + \mathbb{E}_q[\mathbb{I}_{z_i=j}\log x_i] + r_j\right) \\
&\qquad\qquad - \left(\mathbb{E}_q[n_k] + s_j\right)\mathbb{E}_q[\log\Gamma(\alpha_j)] + \frac{1}{2}\log(\sigma_j^2) + \frac{\mathbb{E}_q[(\alpha_j-\hat{\alpha}_j)^2]}{2\sigma_j^2}\Bigg) \\
&= \underset{(\hat{\alpha}_j,\sigma_j)\in\mathbb{R}_+^2}{\operatorname{argmin}} \Bigg(\hat{\alpha}_j\left(\left(\psi(\gamma_j) - \log(\lambda_j)\right)\sum_{i=1}^{N} q_{i,j} + \sum_{i=1}^{N} q_{i,j}\log x_i + r_j\right) \\
&\qquad\qquad - \left(\sum_{i=1}^{N} q_{i,j} + s_j\right)\mathbb{E}_q[\log\Gamma(\alpha_j)] + \frac{1}{2}\log(\sigma_j^2)\Bigg)
\end{aligned}
\tag{3.16}
$$

To arrive at the expression above we have used the identities $\mathbb{E}_q[\alpha_j] = \hat{\alpha}_j$ and $\mathrm{Var}[\alpha_j] = \mathbb{E}_q\left[(\alpha_j - \hat{\alpha}_j)^2\right] = \sigma_j^2$ which follow directly from the assumption that $\alpha_j$ is normal, as well as the identity $\mathbb{E}_q[\beta_j] = \psi(\gamma_j) - \log\lambda_j$. We approximate the expectation of $\log\Gamma(\alpha_j)$ using eq. (3.14) to get

$$
\begin{aligned}
\mathbb{E}_q[\log\Gamma(\alpha_j)] &\approx \log\Gamma(\hat{\alpha}_j) + (\mathbb{E}_q[\alpha_j] - \hat{\alpha}_j)\psi(\hat{\alpha}_j) + \frac{1}{2}\,\mathbb{E}_q[(\alpha_j-\hat{\alpha}_j)^2]\psi^{(1)}(\hat{\alpha}_j)] \\
&= \log\Gamma(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2\psi^{(1)}(\hat{\alpha}_j).
\end{aligned}
\tag{3.17}
$$

Interestingly, although substituting the Taylor approximation directly into eq. (3.13) results in an inaccurate variational approximation, we find that the *expectation* of the Taylor approximation itself is quite close to the actual expectation (calculated using numerical methods). This is illustrated in fig. 3.3 where we show a direct numerical approximation to expectation of the log-gamma function alongside the expectation of its Taylor approximation. To produce this figure we must note that log-gamma function is strictly real for positive, real arguments, while for negative arguments it can take on complex values. To prevent any unexpected behaviour, instead of attempting to calculate $\mathbb{E}[\log\Gamma(\hat{\alpha}_j)]$ directly over the normal variable $\hat{\alpha}_j$, we use a gamma random variable $X$ with the same mean $\hat{\alpha}_j$ and variance $\sigma_j^2$, and then evaluate $\mathbb{E}[\log\Gamma(X)]$ numerically to produce the black line in fig. 3.3.

Substituting eq. (3.17) into eq. (3.16) gives us[2]

$$(\hat{\alpha}_j, \sigma_j) = \underset{(\hat{\alpha}_j,\sigma_j)\in\mathbb{R}_+^2}{\operatorname{argmin}} \left( \hat{\alpha}_j \left( (\psi(\gamma_j) - \log(\lambda_j)) \sum_{i=1}^{N} q_{i,j} + \sum_{i=1}^{N} q_{i,j} \log x_i + r_j \right) \right. \\ \left. - \left( \sum_{i=1}^{N} q_{i,j} + s_j \right) \left( \log\Gamma(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2 \psi^{(1)}(\hat{\alpha}_j) \right) \right). \tag{3.18}$$

We can find this minimum iteratively; first, by numerical optimisation with respect to $\hat{\alpha}_k$.

$$\hat{\alpha}_j = \underset{\hat{\alpha}_j\in\mathbb{R}_+}{\operatorname{argmin}} \left( \hat{\alpha}_j \left( (\psi(\gamma_j) - \log(\lambda_j)) \sum_{i=1}^{N} q_{i,j} + \sum_{i=1}^{N} q_{i,j} \log x_i + r_j \right) \right. \\ \left. - \left( \sum_{i=1}^{N} q_{i,j} + s_j \right) \left( \log\Gamma(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2 \psi^{(1)}(\hat{\alpha}_j) \right) \right)$$

We use the Newton-Raphson method to find the root of the first derivative. (See appendix A.2 for a brief introduction.) The objective function is

$$f(\hat{\alpha}_j) = \hat{\alpha}_j \left( r_j + (\psi(\gamma_j) - \log(\lambda_j) + \log x_i) \sum_{i=1}^{N} q_{i,j} \right) \\ - \left( \sum_{i=1}^{N} q_{i,j} + s_j \right) \left( \log\Gamma(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2 \psi^{(1)}(\hat{\alpha}_j) \right).$$

We need the first and second derivatives of $f$.

$$f'(\hat{\alpha}_j) = (\psi(\gamma_j) - \log(\lambda_j)) \sum_{i=1}^{N} q_{i,j} + \sum_{i=1}^{N} q_{i,j} \log x_i + r_j \\ - \left( \sum_{i=1}^{N} q_{i,j} + s_j \right) \left( \psi(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2 \psi^{(2)}(\hat{\alpha}_j) \right)$$

$$f''(\hat{\alpha}_j) = \left( \sum_{i=1}^{N} q_{i,j} + s_j \right) \left( \psi^{(1)}(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2 \psi^{(3)}(\hat{\alpha}_j) \right)$$

Then the $(t+1)^{\text{th}}$ Newton-Raphson update for $\hat{\alpha}_j$ is

$$\hat{\alpha}_j^{[t+1]} = \hat{\alpha}_j^{[t]} - \frac{f'\left(\hat{\alpha}_j^{[t]}\right)}{f''\left(\hat{\alpha}_j^{[t]}\right)} \tag{3.19}$$

The Newton-Raphson algorithm repeats this step until the change $|\hat{\alpha}_j^{[t+1]} - \hat{\alpha}_j^{[t]}|$ falls below a prescribed threshold.

2. Strictly speaking, the eq. (3.18) is an approximation eq. (3.16), not equal to it. But since we go through many consecutive approximations to arrive at $(\hat{\alpha}_j, \sigma_j^2)$ anyway it is convenient to suppress the $\approx$ and write $=$.

Next, we minimise with respect to $\sigma_j^2$. Dropping all the terms in eq. (3.18) that don't depend on $\sigma_j^2$, we get the following objective function.

$$\sigma_j \in \underset{(\hat{\alpha}_j,\sigma_j)\in\mathbb{R}_+^2}{\operatorname{argmin}} \left( -\frac{1}{2} \left( \sum_{i=1}^{N} q_{i,j} + s_j \right) \sigma_j^2 \psi^{(1)}(\hat{\alpha}_j) + \frac{1}{2} \log(\sigma_j^2) \right)$$

We can optimise $\sigma_j^2$ directly by solving from the root of first derivative:

$$\sigma_j^2 = \frac{1}{\left( \sum_{i=1}^{N} q_{i,j} + s_j \right) \psi^{(1)}(\hat{\alpha}_j)}. \tag{3.20}$$

This concludes the $\alpha_j$ step.

Next, we optimise $q_{i,j}$. Define $\tilde{q}_{i,j}$ as follows:

$$\begin{aligned}
q^*(z_i = j) &\propto \exp \mathbb{E}_{q_{-z_i}} \left[ \log p(\boldsymbol{x} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \boldsymbol{z}) \right] \\
&= \exp \mathbb{E}_{q_{-z_i}} \left[ \sum_{i=1}^{N} \mathbb{I}_{z_i=j} \left( \log \pi_j + \alpha_j \log \beta_j - \log \Gamma(\alpha_j) + (\alpha_j - 1) \log x_i - \beta_j x_i \right) \right] \\
&\approx \exp \left( \psi\left(\zeta_j\right) - \psi \left( \sum_{k=1}^{K} \omega_k + N \right) + \hat{\alpha}_j \left( \psi(\gamma_j) - \log(\lambda_j) \right) \right. \\
&\qquad \left. - \log \Gamma(\hat{\alpha}_j) - \frac{1}{2} \sigma_j^2 \psi^{(1)}(\hat{\alpha}_j) + (\hat{\alpha}_j - 1) \log x_i - \frac{\gamma_j}{\lambda_j} x_i \right) \\
&= \tilde{q}_{i,j}
\end{aligned}$$

where we have used the identities

$$\begin{aligned}
\mathbb{E}_q[\log(\pi_j)] &= \psi\left(\zeta_j\right) - \psi \left( \sum_{k=1}^{K} \zeta_k \right) \\
&= \psi\left(\zeta_j\right) - \psi \left( \sum_{k=1}^{K} \omega_k + N \right)
\end{aligned}$$

$$\mathbb{E}_q[\log(\beta_j)] = \psi(\gamma_j) - \log(\lambda_j)$$

and, as before,

$$\mathbb{E}_q[\log \Gamma(\alpha_j)] \approx \log \Gamma(\hat{\alpha}_j) + \frac{1}{2} \sigma_j^2 \psi^{(1)}(\hat{\alpha}_j).$$

Since $z_i$ has a categorical distribution, all the priors cancel out when we normalise $q^*(z_i = j)$ and we are left with

$$q_{i,j} = q^*(z_i = j) \approx \frac{\tilde{q}_{i,j}}{\sum_{k=1}^{K} \tilde{q}_{i,k}}.$$

Finally, evidence lower bound in eq. (2.10) is

$$\mathcal{L}(q) = \mathbb{E}_q \left[ \log p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \boldsymbol{x}) \right] - \mathbb{E}_q \left[ \log q(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}) \right] \tag{3.21}$$

The first term here is the expectation of the complete log posterior. The posterior is

$$\begin{aligned} p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \boldsymbol{x}) &= p(\boldsymbol{x} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}) p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}) \\ &= \left( \frac{1}{B(\boldsymbol{\omega})} C^{-1} \prod_{k=1}^{K} \pi_k^{\omega_k - 1} \frac{d_k^{c_k}}{\Gamma(c_k)} \beta_k^{c_k - 1} \mathrm{e}^{-d_k \beta_k} \frac{\mathrm{e}^{r_k \alpha_k}}{\Gamma(\alpha_k)^{s_k}} \right) \\ &\quad \times \left( \prod_{k=1}^{K} \prod_{\{i : z_i = k\}} \pi_k \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)} x_i^{\alpha_k - 1} \mathrm{e}^{-\beta_k x_i} \right) \end{aligned} \tag{3.22}$$

where normalisation constants $B(\boldsymbol{\omega}) = \frac{\prod_{k=1}^{K} \Gamma(\omega_k)}{\Gamma\left(\sum_{k=1}^{K} \omega_k\right)}$ and $C = \prod_{k=1}^{K} \int_0^\infty \frac{\mathrm{e}^{r_k x}}{\Gamma(x)^{s_k}} dx$ originate from the priors on $\boldsymbol{\pi}$ and $\boldsymbol{\alpha}$ respectively. In implementation, $B(\boldsymbol{\omega})$ and $C$ should be calculated at the start of the program to avoid unnecessary compute expenditure.

So, first term of the ELBO is

$$\begin{aligned} \mathbb{E}_q \left[ \log p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \boldsymbol{x}) \right] \approx & - \log B(\boldsymbol{\omega}) - \log C \\ & + \sum_{k=1}^{K} \left( \left( \omega_k + \sum_{i=1}^{N} q_{i,k} - 1 \right) \left( \psi\left(\zeta_k\right) - \psi\left( \sum_{j=1}^{K} \omega_j + N \right) \right) \right) \\ & + c_k \log d_k - \log \Gamma(c_k) + \left( c_k - 1 - \hat{\alpha}_k \sum_{i=1}^{N} q_{i,k} \right) \left( \log(\lambda_k) - \psi(\gamma_k) \right) \\ & - d_k \frac{\gamma_k}{\lambda_k} + r_k \hat{\alpha}_k - \left( s_k + \sum_{i=1}^{N} q_{i,k} \right) \left( \log \Gamma(\hat{\alpha}_k) + \frac{1}{2} \sigma_k^2 \psi^{(1)}(\hat{\alpha}_k) \right) \\ & + (\hat{\alpha}_k - 1) \sum_{i=1}^{N} q_{i,k} \log x_i - \frac{\gamma_k}{\lambda_k} \sum_{i=1}^{N} q_{i,k} x_i \Bigg) \end{aligned} \tag{3.23}$$

The second term in eq. (3.21) is the differential entropy $h(q(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi})) = -\mathbb{E}_q[\log q(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi})]$ which decomposes into a sum of entropies as follows.

$$\begin{aligned} -\mathbb{E}_q[\log q(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi})] &= -\left( \sum_{k=1}^{K} \mathbb{E}_q[\log q(\alpha_k)] + \sum_{k=1}^{K} \mathbb{E}_q\left[\log q(\beta_k)\right] + \mathbb{E}_q\left[\log q(\boldsymbol{\pi})\right] \right) \\ &= \sum_{k=1}^{K} h(q(\alpha_k)) + \sum_{k=1}^{K} h(q(\beta_k)) + h(q(\boldsymbol{\pi})) \end{aligned} \tag{3.24}$$

Here, $h(q(\alpha_k)) = \frac{1}{2} \log(2\pi \mathrm{e} \sigma_k^2)$ is the differential entropy of the normal distribution, $h(q(\beta_k)) = \gamma_k - \log \lambda_k + \log \Gamma(\gamma_k) + (1 - \gamma_k)\psi(\gamma_k)$ is the differential entropy of the inverse-gamma distribution, $h(q(\boldsymbol{\pi})) = \log B(\boldsymbol{\xi}) + \left( \sum_{k=1}^{K} \xi_k - K \right) \psi\left( \sum_{k=1}^{K} \xi_k \right) - \sum_{k=1}^{K} (\xi_k - 1)\psi(\xi_k)$ is the differential entropy of the Dirichlet distribution, and $B(\boldsymbol{\xi}) = \frac{\prod_{k=1}^{K} \Gamma(\xi_k)}{\Gamma\left(\sum_{k=1}^{K} \xi_k\right)}$. Substituting

these expressions into eq. (3.21) yields

$$
\begin{aligned}
-\mathbb{E}_q[\log q(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi})] = \sum_{k=1}^{K} \Bigg( & \frac{1}{2}\log(2\pi \mathrm{e}\sigma_k^2) + \log\Gamma(\xi_k) + \gamma_k - \log\lambda_k + \log\Gamma(\gamma_k) \\
& + (1-\gamma_k)\psi(\gamma_k) + \xi_k \psi\left(\sum_{j=1}^{K} \xi_j\right) - (\xi_k - 1)\psi(\xi_k) \Bigg) \\
& - \log\Gamma\left(\sum_{k=1}^{K} \xi_k\right) - K\psi\left(\sum_{k=1}^{K} \xi_k\right)
\end{aligned} \tag{3.25}
$$

Summing eqs. (3.23) and (3.25) gives the ELBO in eq. (3.21). For brevity, we will not write out the full expression.

### 3.3.2 Algorithm

In summary, algorithm 4 returns parameters for the following mean-field variational distributions:

$$q^*(\boldsymbol{\pi}) = \text{Dirichlet}\,(\zeta_1, ..., \zeta_K) \tag{3.26}$$

$$q^*(\beta_j) = \text{Gamma}\,(\gamma_j, \lambda_j) \tag{3.27}$$

$$q^*(\alpha_j) = \mathcal{N}(\hat{\alpha}_j, \sigma_j^2) \tag{3.28}$$

whose product approximates the joint posterior

$$p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z} \mid \boldsymbol{x}) \approx q^*(\boldsymbol{\pi}) \prod_{k=1}^{K} q^*(\alpha_j) q^*(\beta_j)$$

Note that in algorithm 4 we have omitted the superscripts $^{[t]}$ which indicate the current iteration. This is because the terminal values of the variational parameters are usually all that we need - they are the final result. This contrasts with the Gibbs sampler for which the intermittent samples (for $t = 1, ..., T$) *are* the output of the algorithm. We discuss implementation details in section 3.6.

## 3.4 Variational Mixture of Gamma Distributions Parameterized by $\mu$ and $\alpha$

We now consider a reparameterization of the gamma distribution in terms of $\mu_k$ and $\alpha_k$ where $\alpha_k$ is defined as before and $\mu = \frac{\alpha_k}{\beta_k}$. The single-component probability density function under this reparameterization becomes

$$p(x \mid \alpha, \mu) = \frac{\left(\frac{\alpha}{\mu}\right)^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} \mathrm{e}^{-\frac{\alpha}{\mu}x}$$

**Algorithm 4:** `VI-1` Coordinate-Ascent Variational Inference for Mixture of Gamma Distributions Parameterised by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$

Initialise $q_{i,k} \leftarrow \frac{1}{K}$ for all $i = 1, ..., N$ and $k = 1, ..., K$, $\hat{\boldsymbol{\alpha}}$ to a vector of unique positive numbers, and $\sigma_k^2 \leftarrow \left(s_j \psi^{(1)}(\hat{\alpha}_j) + \frac{1}{2\hat{\alpha}_j^2} \sum_{i=1}^N q_{i,j}\right)^{-1}$.

**repeat**

  $\zeta_k \leftarrow \omega_k + \sum_{i=1}^N q_{i,k}$

  $\gamma_j \leftarrow \hat{\alpha}_j \sum_{i=1}^N q_{i,j} + c_j$

  $\lambda_j \leftarrow d_j + \sum_{i=1}^N q_{i,j} x_i$

  **repeat**

    $\hat{\alpha}_j \leftarrow \hat{\alpha}_j + \frac{(\psi(\gamma_j) - \log(\lambda_j)) \sum_{i=1}^N q_{i,j} + \sum_{i=1}^N q_{i,j} \log x_i + r_j - \left(\sum_{i=1}^N q_{i,j} + s_j\right)\left(\psi(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2 \psi^{(2)}(\hat{\alpha}_j)\right)}{\left(\sum_{i=1}^N q_{i,j} + s_j\right)\left(\psi^{(1)}(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2 \psi^{(3)}(\hat{\alpha}_j)\right)}$

    $\sigma_j^2 \leftarrow \left(\left(\sum_{i=1}^N q_{i,j} + s_j\right) \psi^{(1)}(\hat{\alpha}_j)\right)^{-1}$

  **until** *convergence*

  $q_{i,j} \leftarrow \exp\left(\psi(\zeta_j) - \psi\left(\sum_{k=1}^K \omega_k + N\right) + \hat{\alpha}_j (\psi(\gamma_j) - \log(\lambda_j)) - \log\Gamma(\hat{\alpha}_j) - \frac{1}{2}\sigma_j^2 \psi^{(1)}(\hat{\alpha}_j) + (\hat{\alpha}_j - 1)\log x_i - \frac{\gamma_j}{\lambda_j} x_i\right)$

  $q_{i,j} \leftarrow \frac{q_{i,j}}{\sum_{k=1}^K q_{i,k}}$

**until** *change in ELBO falls below some prescribed threshold*

The probability density function of the corresponding mixture is

$$p(x \mid \boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}) = \sum_{k=1}^K \pi_k p(x \mid \mu_k, \alpha_k) \tag{3.29}$$

$$= \sum_{k=1}^K \pi_k \frac{\left(\frac{\alpha_k}{\mu_k}\right)^{\alpha_k}}{\Gamma(\alpha_k)} x^{\alpha_k - 1} \mathrm{e}^{-\frac{\alpha_k}{\mu_k} x} \tag{3.30}$$

and the likelihood of $N$ observations $\boldsymbol{x} = (x_1, ..., x_N)$ is

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}) = \prod_{i=1}^N \sum_{k=1}^K \pi_k \frac{\left(\frac{\alpha_k}{\mu_k}\right)^{\alpha_k}}{\Gamma(\alpha_k)} x_i^{\alpha_k - 1} \mathrm{e}^{-\frac{\alpha_k}{\mu_k} x_i}.$$

The joint posterior given observations $\boldsymbol{x}$ is

$$\begin{aligned} p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi} \mid \boldsymbol{x}) &= p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}) p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}) \\ &= p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}) \prod_{i=1}^N \sum_{k=1}^K \pi_k p(x_i \mid \mu_k, \alpha_k). \end{aligned}$$

We assume the same priors on $\boldsymbol{\pi}$ and $\alpha_k$.

$$p(\alpha_k) \propto \frac{\mathrm{e}^{r_k \alpha_k}}{\Gamma(\alpha_k)^{s_k}}$$

$$p(\boldsymbol{\pi}) = \text{Dirichlet}(\boldsymbol{\omega})$$

where $r_k, s_k, > 0$, $\boldsymbol{\omega} = (\omega_1, ..., \omega_K)$, and $\omega_k > 0$. For the prior on $\mu_k$ we use

$$p(\mu_k) = \text{Inv-Gamma}(\xi_k, \tau_k)$$

where $\xi_k, \tau_k > 0$. The joint posterior is then

$$\begin{aligned} p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi} \mid \boldsymbol{x}) &\propto p(\boldsymbol{\pi}) \left( \prod_{k=1}^{K} p(\mu_k) p(\alpha_k) \right) \left( \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k p(x_i \mid \mu_k, \alpha_k) \right) \\ &= \left( \prod_{k=1}^{K} \pi_k^{\omega_k - 1} \frac{\tau_k^{\xi_k}}{\Gamma(\xi_k)} \mu_k^{-1-\xi_k} \mathrm{e}^{-\frac{\tau_k}{\mu_k}} \frac{\mathrm{e}^{r_k \alpha_k}}{\Gamma(\alpha_k)^{s_k}} \right) \left( \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \frac{\left(\frac{\alpha_k}{\mu_k}\right)^{\alpha_k}}{\Gamma(\alpha_k)} x_i^{\alpha_k - 1} \mathrm{e}^{-\frac{\alpha_k}{\mu_k} x_i} \right). \end{aligned}$$

Introducing the component assignment variable $\boldsymbol{z} = (z_1, ..., z_n)$, the joint posterior becomes

$$\begin{aligned} p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{z} \mid \boldsymbol{x}) &\propto \left( \prod_{k=1}^{K} \pi_k^{\omega_k - 1} \frac{\tau_k^{\xi_k}}{\Gamma(\xi_k)} \mu_k^{-1-\xi_k} \mathrm{e}^{-\frac{\tau_k}{\mu_k}} \frac{\mathrm{e}^{r_k \alpha_k}}{\Gamma(\alpha_k)^{s_k}} \right) \left( \prod_{i=1}^{N} \sum_{k=1}^{K} \mathbb{I}_{z_i = k} \, \pi_k \frac{\left(\frac{\alpha_k}{\mu_k}\right)^{\alpha_k}}{\Gamma(\alpha_k)} x_i^{\alpha_k - 1} \mathrm{e}^{-\frac{\alpha_k}{\mu_k} x_i} \right) \\ &\propto \prod_{k=1}^{K} \pi_k^{\omega_k + n_k - 1} \frac{\mu_k^{1-\xi_k - n_k a_k} \alpha_k^{n_k \alpha_k}}{\Gamma(\alpha_k)^{n_k + s_k}} \mathrm{e}^{r_k \alpha_k - \frac{\tau_k}{\mu_k}} \prod_{\{i: z_i = k\}} x_i^{\alpha_k - 1} \mathrm{e}^{-\frac{\alpha_k}{\mu_k} x_i}. \end{aligned}$$

For convenience, we write this in exp-log form.

$$\begin{aligned} p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi} \mid \boldsymbol{x}) = \exp \Bigg( \sum_{k=1}^{K} \Bigg( & (\omega_k + n_k - 1) \log \pi_k + (1 - \xi_k - n_k \alpha_k) \log \mu \\ & + n_k \alpha_k \log \alpha_k - (n_k + s_k) \log \Gamma(\alpha_k) + r_k \alpha_k - \frac{\tau_k}{\mu_k} \\ & + \sum_{i=1}^{N} \mathbb{I}_{z_i = k} \left( (\alpha_k - 1) \log x_i - \frac{\alpha_k}{\mu_k} x_i \right) \Bigg) \Bigg). \end{aligned}$$

### 3.4.1 Update equations

Since many of the derivations under the reparameterisation are identical those in section 3.3, we omit much of the redundant details. The optimal variational distribution for $\boldsymbol{\pi}$ is unchanged.

$$q^*(\boldsymbol{\pi}) = \text{Dirichlet}\left(\zeta_1, ..., \zeta_K\right), \qquad \zeta_k = \omega_k + \sum_{i=1}^{N} q_{i,k}.$$

Recall that the update equation for $\beta_j$ had the form of a gamma distribution. Analogously, the update equation for $\mu_j = \frac{\alpha_j}{\beta_j}$ has the form of an inverse-gamma distribution.

$$\begin{aligned}
q^*(\mu_j) &= \exp \mathbb{E}_{q_{-\mu_j}} \left[\log p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{z} \mid \boldsymbol{x})\right] \\
&\propto \exp\left((1 - \xi_k - \mathbb{E}_q[n_j]\,\mathbb{E}_q[\alpha_j])\log\mu_j - \frac{\tau_k}{\mu_k} - \sum_{i=1}^{N} \mathbb{E}_q[\mathbb{I}_{z_i=j}]\frac{\alpha_k}{\mu_k}x_i\right) \\
&\propto \mu_j^{1-\xi_k-\mathbb{E}_q[\alpha_j]\sum_{i=1}^{N} q_{i,j}} \exp\left(-\frac{1}{\mu_k}\left(\tau_k + \mathbb{E}_q[\alpha_k]\sum_{i=1}^{N} q_{i,j}x_i\right)\right) \\
&= \text{Inv-Gamma}\,(\gamma_j, \lambda_j)
\end{aligned}$$

Here, $\gamma_j = \xi_k + \mathbb{E}_q[\alpha_j]\sum_{i=1}^{N} q_{i,j}$ and $\lambda_j = \tau_k + \mathbb{E}_q[\alpha_k]\sum_{i=1}^{N} q_{i,j}x_i$.

Next, the update equation for $\alpha_j$ is

$$\begin{aligned}
q^*(\alpha_j) &= \exp \mathbb{E}_{q_{-\alpha_j}} \left[\log p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{z} \mid \boldsymbol{x})\right] \\
&\propto \exp\left(\alpha_j\left((\psi(\gamma_j) - \log(\lambda_j) + \log\alpha_j)\sum_{i=1}^{N} q_{i,j} + \alpha_j\log(\alpha_j)\sum_{i=1}^{N} q_{i,j} + r_j\right.\right. \\
&\qquad \left.\left. + \sum_{i=1}^{N} q_{i,j}\left(\log x_i - \frac{\gamma_j}{\lambda_j}x_i\right)\right) - \left(\sum_{i=1}^{N} q_{i,j} + s_j\right)\log\Gamma(\alpha_j)\right)
\end{aligned}$$

As in eq. (3.13), this is not in the form of any known distribution. Therefore, we assume that $q(\alpha_j) = \mathcal{N}(\hat{\alpha}_j, \sigma_j^2)$ where $\hat{\alpha}_j$ and $\sigma_j^2$ satisfy

$$\begin{aligned}
(\hat{\alpha}_j, \sigma_j) &= \underset{(\hat{\alpha}_j,\sigma_j)\in\mathbb{R}_+^2}{\operatorname{argmin}} \mathbb{E}_q\left[\log\left(\frac{p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{z} \mid \boldsymbol{x})}{q(\alpha_j)}\right)\right] \\
&= \underset{(\hat{\alpha}_j,\sigma_j)\in\mathbb{R}_+^2}{\operatorname{argmin}} \left(\hat{\alpha}_j\left((\psi(\gamma_j) - \log(\lambda_j))\sum_{i=1}^{N} q_{i,j} + r_j + \sum_{i=1}^{N} q_{i,j}\left(\log x_i - \frac{\gamma_j}{\lambda_j}x_i\right)\right)\right. \\
&\qquad \left. + \mathbb{E}_q[\alpha_j\log\alpha_j]\sum_{i=1}^{N} q_{i,j} - \left(\sum_{i=1}^{N} q_{i,j} + s_j\right)\mathbb{E}_q[\log\Gamma(\alpha_j)] + \frac{1}{2}\log(\sigma_j^2)\right)
\end{aligned} \tag{3.31}$$

The expectations of $\alpha_j \log\alpha_j$ and $\log\Gamma(\alpha_j)$ intractable, so we seek a suitable approximation. Consider both terms together with their respective coefficients:

$$\begin{aligned}
\mathbb{E}_q[\alpha_j\log\alpha_j]\sum_{i=1}^{N} q_{i,j} - \left(\sum_{i=1}^{N} q_{i,j} + s_j\right)\mathbb{E}_q[\log\Gamma(\alpha_j)] = \mathbb{E}_q\left[\alpha_j\log\alpha_j - \log\Gamma(\alpha_j)\right]\sum_{i=1}^{N} q_{i,j} \\
- s_j\,\mathbb{E}_q\left[\log\Gamma(\alpha_j)\right]
\end{aligned} \tag{3.32}$$

*Stirling's formula* gives an asymptotic expansion for $\log\Gamma(\alpha_j)$ valid as $\alpha_j \to \infty$:

$$\log\Gamma(\alpha_j) \approx \frac{1}{2}\log(2\pi) + \alpha_j \log\alpha_j - \frac{1}{2}\log\alpha_j - \alpha_j + \sum_{n=1}^{\infty} \frac{B_{2n}}{2n(2n-1)\alpha_j^{2n-1}}$$

where $B_n$ is a Bernoulli number. By truncating the summation terms, we obtain the following expression known as *Stirling's approximation*.

$$\log\Gamma(\alpha_j) \approx \frac{1}{2}\log(2\pi) + \alpha_j \log\alpha_j - \frac{1}{2}\log\alpha_j - \alpha_j \tag{3.33}$$

Remarkably, the terms $\alpha_j \log\alpha_j$ cancel from eqs. (3.32) and (3.33). Substituting Stirling's approximation into $\mathbb{E}_q[\alpha_j \log\alpha_j - \log\Gamma(\alpha_j)]$ gives the approximation in eq. (3.35). Figure 3.5 shows that this is very accurate for $x \gg 1$.

$$\mathbb{E}_q\left[\alpha_j \log\alpha_j - \log\Gamma(\alpha_j)\right] \approx \mathbb{E}_q\left[\alpha_j \log\alpha_j - \frac{1}{2}\log(2\pi) - \alpha_j \log\alpha_j + \frac{1}{2}\log\alpha_j + \alpha_j\right] \tag{3.34}$$

$$= -\frac{1}{2}\log(2\pi) + \frac{1}{2}\mathbb{E}_q\left[\log\alpha_j\right] + \mathbb{E}_q[\alpha_j] \tag{3.35}$$

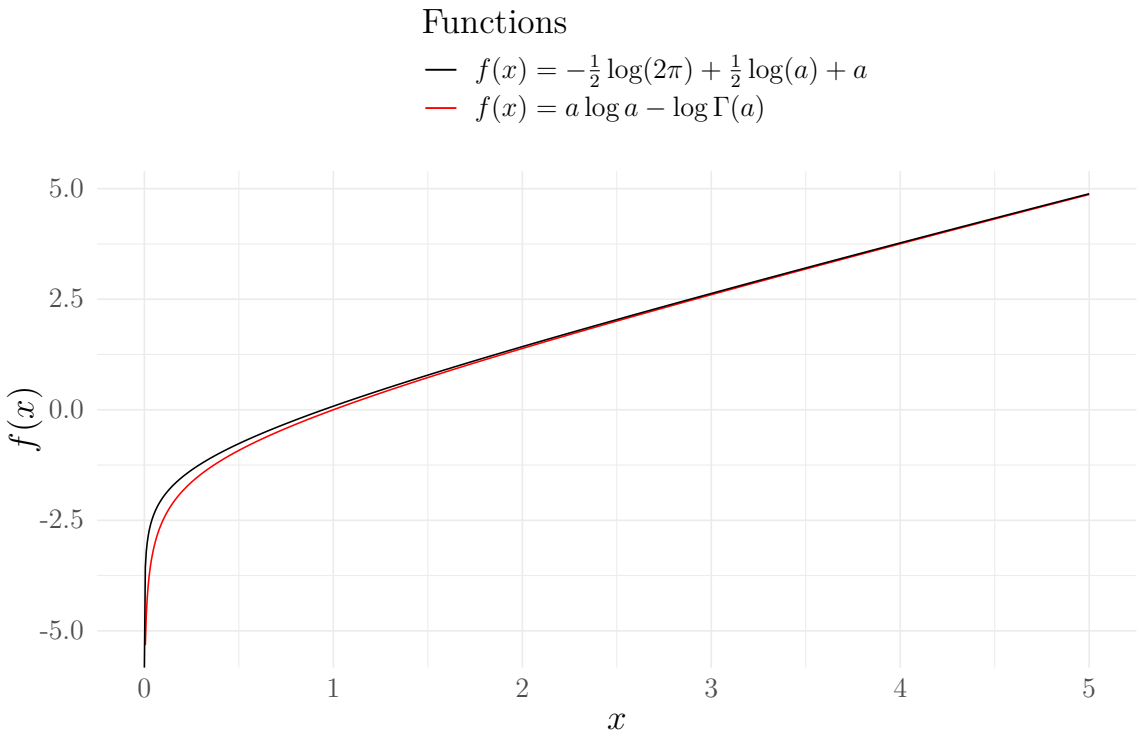



Figure 3.5: The function $a\log a - \log\Gamma(a)$ (red) together with an approximation derived from Stirling's expansion of the log-gamma function (black).

We already have $\mathbb{E}_q[\alpha_j] = \hat{\alpha}_j$, and we can approximate the expectation of $\log\alpha_j$ by a second-order Taylor expansion around $\hat{\alpha}_j$.

$$\mathbb{E}_q[\log\alpha_j] \approx \mathbb{E}_q\left[\log\hat{\alpha}_j + (\alpha_j - \hat{\alpha}_j)\frac{1}{\hat{\alpha}_j} - \frac{1}{2}(\alpha_j - \hat{\alpha}_j)^2\frac{1}{\hat{\alpha}_j^2}\right]$$

$$= \log\hat{\alpha}_j - \frac{\sigma_j^2}{2\hat{\alpha}_j^2}$$

Thus, eq. (3.35) becomes

$$\mathbb{E}_q\left[\alpha_j \log \alpha_j - \log \Gamma(\alpha_j)\right] \approx -\frac{1}{2}\log(2\pi) + \frac{1}{2}\log \hat{\alpha}_j - \frac{\sigma_j^2}{4\hat{\alpha}_j^2} + \hat{\alpha}_j. \tag{3.36}$$

Since, in practise, we would typically set $s_j$ to be very small (to induce a weakly-informative prior), the remaining term $s_j\,\mathbb{E}_q\left[\log \Gamma(\alpha_j)\right]$ in eq. (3.32) can be safely neglected. This way, we avoid need to call the computationally expensive polygamma function $\psi^{(n)}(\hat{\alpha}_j)$ at any point. However, for the sake of completeness, we can approximate it using the Taylor expansion from eq. (3.17).

$$\log \Gamma(\alpha_j) \approx \log \Gamma(\hat{\alpha}_j) + (\alpha_1 - \hat{\alpha}_j)\psi(\hat{\alpha}_j) + \frac{1}{2}(\alpha_j - \hat{\alpha}_j)^2\psi^{(1)}(\hat{\alpha}_j) \tag{3.37}$$

with the expectation

$$\mathbb{E}_q[\log \Gamma(\alpha_j)] \approx \log \Gamma(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2\psi^{(1)}(\hat{\alpha}_j). \tag{3.38}$$

Combining eq. (3.35) and eq. (3.38) gives the following approximation to eq. (3.32).

$$\begin{aligned}\mathbb{E}_q[\alpha_j \log \alpha_j]\sum_{i=1}^{N} q_{i,j} - \left(\sum_{i=1}^{N} q_{i,j} + s_j\right)\mathbb{E}_q[\log \Gamma(\alpha_j)] \approx \left(-\frac{1}{2}\log(2\pi) + \frac{1}{2}\log \hat{\alpha}_j - \frac{\sigma_j^2}{4\hat{\alpha}_j^2} + \hat{\alpha}_j\right)\sum_{i=1}^{N} q_{i,j} \\ - s_j\left(\log \Gamma(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2\psi^{(1)}(\hat{\alpha}_j)\right).\end{aligned}$$

Substituting this into eq. (3.31) gives

$$\begin{aligned}(\hat{\alpha}_j, \sigma_j) = \underset{(\hat{\alpha}_j,\sigma_j)\in\mathbb{R}_+^2}{\operatorname{argmin}} \Bigg(\hat{\alpha}_j\left((\psi(\gamma_j) - \log(\lambda_j))\sum_{i=1}^{N} q_{i,j} + r_j + \sum_{i=1}^{N} q_{i,j}\left(\log x_i - \frac{\gamma_j}{\lambda_j}x_i\right)\right) \\ + \frac{1}{2}\log(\sigma_j^2) + \left(-\frac{1}{2}\log(2\pi) + \frac{1}{2}\log \hat{\alpha}_j - \frac{\sigma_j^2}{4\hat{\alpha}_j^2} + \hat{\alpha}_j\right)\sum_{i=1}^{N} q_{i,j} \\ - s_j\left(\log \Gamma(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2\psi^{(1)}(\hat{\alpha}_j)\right)\Bigg).\end{aligned} \tag{3.39}$$

Similarly to in section 3.3, we can find this minimum iteratively. First, minimise with respect to $\hat{\alpha}_j$ using the Newton-Raphson algorithm. Let the objective function be

$$\begin{aligned}f(\hat{\alpha}_j) = \hat{\alpha}_j\left((\psi(\gamma_j) - \log(\lambda_j))\sum_{i=1}^{N} q_{i,j} + r_j + \sum_{i=1}^{N} q_{i,j}\left(\log x_i - \frac{\gamma_j}{\lambda_j}x_i\right)\right) \\ + \left(\frac{1}{2}\log \hat{\alpha}_j - \frac{\sigma_j^2}{4\hat{\alpha}_j^2} + \hat{\alpha}_j\right)\sum_{i=1}^{N} q_{i,j} - s_j\left(\log \Gamma(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2\psi^{(1)}(\hat{\alpha}_j)\right).\end{aligned}$$

To implement the Newton-Raphson algorithm, we require the first and second derivatives of $f$.

$$f'(\hat{\alpha}_j) = (\psi(\gamma_j) - \log(\lambda_j)) \sum_{i=1}^{N} q_{i,j} + r_j + \sum_{i=1}^{N} q_{i,j} \left( \log x_i - \frac{\gamma_j}{\lambda_j} x_i \right) + \left( \frac{1}{2\hat{\alpha}_j} + \frac{\sigma_j^2}{2\hat{\alpha}_j^3} + 1 \right) \sum_{i=1}^{N} q_{i,j} - s_j \left( \psi(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2 \psi^{(2)}(\hat{\alpha}_j) \right)$$

$$f''(\hat{\alpha}_j) = - \left( \frac{1}{2\hat{\alpha}_j^2} + \frac{3\sigma_j^2}{2\hat{\alpha}_j^4} \right) \sum_{i=1}^{N} q_{i,j} - s_j \left( \psi^{(1)}(\hat{\alpha}_j) + \frac{1}{2}\sigma_j^2 \psi^{(3)}(\hat{\alpha}_j) \right)$$

The $(t+1)^{\text{th}}$ Newton-Raphson update for $\hat{\alpha}_j$ is

$$\hat{\alpha}_j^{[t+1]} = \hat{\alpha}_j^{[t]} - \frac{f'\left(\hat{\alpha}_j^{[t]}\right)}{f''\left(\hat{\alpha}_j^{[t]}\right)}$$

As before, the Newton-Raphson algorithm repeats this step until the value $|\hat{\alpha}_j^{[t+1]} - \hat{\alpha}_j^{[t]}|$ falls below some prescribed tolerance.

We then minimise with respect to $\sigma_j^2$.

$$\sigma_j^2 = \underset{\sigma_j^2 \in \mathbb{R}_+}{\operatorname{argmin}} \left( -\sigma_j^2 \left( \frac{1}{4\hat{\alpha}_j^2} \sum_{i=1}^{N} q_{i,j} + \frac{s_j}{2} \psi^{(1)}(\hat{\alpha}_j) \right) + \frac{1}{2} \log(\sigma_j^2) \right)$$

We can find the optimum $\sigma_j^2$ from solving the root of first derivative of eq. (3.39) with respect to $\sigma_j^2$.

$$\sigma_k^2 = \left( s_j \psi^{(1)}(\hat{\alpha}_j) + \frac{1}{2\hat{\alpha}_j^2} \sum_{i=1}^{N} q_{i,j} \right)^{-1}$$

This concludes the $\alpha_j$ step.

Next, we need to calculate $q_{i,j}$. As in section 3.3, the priors drop out during the normalisation of the categorical distribution and we are left with

$$\begin{aligned} q_{i,j} &\propto \exp \mathbb{E}_{q_{-z_i}} \left[ \log p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{z}) \right] \\ &= \exp \left( \psi(\zeta_j) - \psi \left( \sum_{k=1}^{K} \omega_k + N \right) - \frac{1}{2} \log(2\pi) + \frac{1}{2} \log \hat{\alpha}_j - \frac{\sigma_j^2}{4\hat{\alpha}_j^2} + \hat{\alpha}_j \right. \\ &\qquad \left. + \hat{\alpha}_j (\psi(\gamma_j) - \log(\lambda_j)) + (\hat{\alpha}_j - 1) \log x_i - \hat{\alpha}_j \frac{\gamma_j}{\lambda_j} x_i \right) \end{aligned}$$

where we have used the following identities which derive from the fact that the reciprocal of an inverse-gamma-distributed random variable $\mu_j \sim \text{Inv-Gamma}(\gamma_j, \lambda_j)$ is gamma-distributed with $\frac{1}{\mu_j} \sim \text{Gamma}(\gamma_j, \lambda_j)$.

$$\mathbb{E}_q \left[ \frac{1}{\mu_j} \right] = \frac{\gamma_j}{\lambda_j}$$

$$\mathbb{E}_q[\log \mu_j] = \log(\lambda_j) - \psi(\gamma_j)$$

As in eq. (3.36),

$$\mathbb{E}_q\left[\alpha_j \log \alpha_j - \log \Gamma(\alpha_j)\right] \approx -\frac{1}{2}\log(2\pi) + \frac{1}{2}\log \hat{\alpha}_j - \frac{\sigma_j^2}{4\hat{\alpha}_j^2} + \hat{\alpha}_j.$$

Finally, the evidence lower bound from eq. (2.10) is

$$\mathcal{L}(q) = \mathbb{E}_q\left[\log p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{x})\right] - \mathbb{E}_q\left[\log q(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi})\right]. \tag{3.40}$$

The first term is the expectation of the complete log posterior. The posterior is

$$\begin{aligned} p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{x}) &= p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}) p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}) \\ &= \left( \frac{1}{B(\boldsymbol{\omega})} C^{-1} \prod_{k=1}^{K} \pi_k^{\omega_k - 1} \frac{\tau_k^{\xi_k}}{\Gamma(\xi_k)} \mu_k^{-1-\xi_k} \mathrm{e}^{-\frac{\tau_k}{\mu_k}} \frac{\mathrm{e}^{r_k \alpha_k}}{\Gamma(\alpha_k)^{s_k}} \right) \\ &\quad \times \left( \prod_{k=1}^{K} \prod_{\{i:z_i=k\}} \pi_k \frac{\left(\frac{\alpha_k}{\mu_k}\right)^{\alpha_k}}{\Gamma(\alpha_k)} x_i^{\alpha_k - 1} \mathrm{e}^{-\frac{\alpha_k}{\mu_k} x_i} \right) \end{aligned} \tag{3.41}$$

where, as before, normalisation constants are $B(\boldsymbol{\omega}) = \frac{\prod_{k=1}^{K} \Gamma(\omega_k)}{\Gamma\left(\sum_{k=1}^{K} \omega_k\right)}$ and $C = \prod_{k=1}^{K} \int_0^\infty \frac{\mathrm{e}^{r_k x}}{\Gamma(x)^{s_k}} dx$ and derive from the priors on $\boldsymbol{\pi}$ and $\boldsymbol{\alpha}$ respectively. Therefore, the first term of the ELBO is

$$\begin{aligned} \mathbb{E}_q\left[\log p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{x})\right] \approx & -\log B(\boldsymbol{\omega}) - \log C \\ & + \sum_{k=1}^{K} \left( \left( \omega_k + \sum_{i=1}^{N} q_{i,k} - 1 \right) \left( \psi\left(\zeta_k\right) - \psi\left( \sum_{j=1}^{K} \omega_j + N \right) \right) \right) \\ & + \xi_k \log \tau_k - \log \Gamma(\xi_k) + \left( -1 - \xi_k + \hat{\alpha}_k \sum_{i=1}^{N} q_{i,k} \right) \left( \psi(\gamma_k) - \log(\lambda_k) \right) \\ & - \tau_k \frac{\gamma_k}{\lambda_k} + r_k \hat{\alpha}_k - \left( -\frac{1}{2}\log(2\pi) + \frac{1}{2}\log \hat{\alpha}_k - \frac{\sigma_k^2}{4\hat{\alpha}_k^2} + \hat{\alpha}_k \right) \sum_{i=1}^{N} q_{i,j} \\ & - s_k \left( \log \Gamma(\hat{\alpha}_k) + \frac{1}{2}\sigma_k^2 \psi^{(1)}(\hat{\alpha}_k) \right) \\ & + (\hat{\alpha}_k - 1) \sum_{i=1}^{N} q_{i,k} \log x_i - \hat{\alpha}_k \frac{\gamma_k}{\lambda_k} \sum_{i=1}^{N} q_{i,k} x_i \Bigg) \end{aligned} \tag{3.42}$$

The second term in eq. (3.40) is the differential entropy $h(q(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi})) = -\mathbb{E}_q[\log q(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi})]$ which decomposes into the following sum of entropies.

$$\begin{aligned} -\mathbb{E}_q[\log q(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi})] &= -\left( \sum_{k=1}^{K} \mathbb{E}_q[\log q(\mu_k)] + \sum_{k=1}^{K} \mathbb{E}_q\left[\log q(\alpha_k)\right] + \mathbb{E}_q\left[\log q(\boldsymbol{\pi})\right] \right) \\ &= \sum_{k=1}^{K} h(q(\mu_k)) + \sum_{k=1}^{K} h(\log q(\alpha_k)) + h(\log q(\boldsymbol{\pi})) \end{aligned} \tag{3.43}$$

Here, $h(q(\mu_k) = \gamma_k + \log\lambda_k + \log\Gamma(\gamma_k) - (\gamma_k+1)\psi(\gamma_k)$ is the differential entropy of the inverse-gamma distribution, $h(q(\alpha_k)) = \frac{1}{2}\log(2\pi\mathrm{e}\sigma_k^2)$ is the differential entropy of the normal distribution, $h(q(\boldsymbol{\pi})) = \log B(\boldsymbol{\xi}) + \left(\sum_{k=1}^K \xi_k - K\right)\psi\left(\sum_{k=1}^K \xi_k\right) - \sum_{k=1}^K (\xi_k - 1)\psi(\xi_k)$ is the differential entropy of the Dirichlet distribution, and $B(\boldsymbol{\xi}) = \frac{\prod_{k=1}^K \Gamma(\xi_k)}{\Gamma\left(\sum_{k=1}^K \xi_k\right)}$. Substituting these values into eq. (3.43) yields

$$
\begin{aligned}
-\mathbb{E}_q[\log q(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi})] = \sum_{k=1}^K \Bigg( & \gamma_k + \log(\lambda_k\Gamma(\gamma_k)) - (\gamma_k+1)\psi(\gamma_k) + \frac{1}{2}\log(2\pi\mathrm{e}\sigma_k^2) \\
& + \log\Gamma(\xi_k) + \xi_k\psi\left(\sum_{j=1}^K \xi_j\right) - (\xi_k - 1)\psi(\xi_k)\Bigg) \\
& - \log\Gamma\left(\sum_{k=1}^K \xi_k\right) - K\psi\left(\sum_{k=1}^K \xi_k\right)
\end{aligned} \tag{3.44}
$$

Adding eqs. (3.42) and (3.44) gives the ELBO in eq. (3.40). Again, we will not write out the full expression since it is quite long.

### 3.4.2 Algorithm

`VI-2` is summarised by algorithm 5. It returns the parameters to the following mean-field variational distributions

$$q^*(\boldsymbol{\pi}) = \text{Dirichlet}\left(\zeta_1, ..., \zeta_K\right), \tag{3.45}$$

$$q^*(\mu_j) = \text{Inv-Gamma}\left(\gamma_j, \lambda_j\right), \tag{3.46}$$

$$q^*(\alpha_j) = \mathcal{N}(\hat{\alpha}_j, \sigma_j^2), \tag{3.47}$$

whose product approximates the joint posterior

$$p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta} \mid \boldsymbol{x}) \approx q^*(\boldsymbol{\pi}) \prod_{k=1}^K q^*(\mu_k) q^*(\alpha_k).$$

## 3.5 Comparison Between the Update Expressions for Variational Inference and the Gibbs Sampler

Table 3.1 compares the conditional distributions used to construct the Gibbs sampler against the variational distributions for `VI-1` and `VI-2`. We have written these in a way that reveals a deep link between Gibbs sampling and CAVI. In fact, the variational distributions for $\boldsymbol{\pi}$, $\beta_k$, and $\mu_k$ can be obtained by taking the expectation of the parameters of their respective conditional distributions with respect to $q$. For instance,

**Algorithm 5:** `VI-2` Variational Inference for Mixture of Gamma Distributions Parameterised by $\boldsymbol{\mu}$ and $\boldsymbol{\alpha}$

Initialise $q_{i,k} \leftarrow \frac{1}{K}$ for all $i = 1, ..., N$ and $k = 1, ..., K$, $\hat{\boldsymbol{\alpha}}$ to a vector of unique positive numbers, and $\sigma_k^2 \leftarrow \left( s_j \psi^{(1)}(\hat{\alpha}_j) + \frac{1}{2\hat{\alpha}_j^2} \sum_{i=1}^N q_{i,j} \right)^{-1}$.

**repeat**

  $\zeta_k \leftarrow \omega_k + \sum_{i=1}^N q_{i,k}$

  $\gamma_j \leftarrow \xi_k + \hat{\alpha}_j \sum_{i=1}^N q_{i,j}$

  $\lambda_j \leftarrow \tau_k + \hat{\alpha}_j \sum_{i=1}^N q_{i,j} x_i$

  **repeat**

    $$\hat{\alpha}_j \leftarrow \hat{\alpha}_j + \frac{(\psi(\gamma_j) - \log(\lambda_j)) \sum_{i=1}^N q_{i,j} + r_j + \sum_{i=1}^N q_{i,j} \left( \log x_i - \frac{\gamma_j}{\lambda_j} x_i \right) + \left( \frac{1}{2\hat{\alpha}_j} + \frac{\sigma_j^2}{2\hat{\alpha}_j^3} + 1 \right) \sum_{i=1}^N q_{i,j} - s_j \left( \psi(\hat{\alpha}_j) + \frac{1}{2} \sigma_j^2 \psi^{(2)}(\hat{\alpha}_j) \right)}{\left( \frac{1}{2\hat{\alpha}_j^2} + \frac{3\sigma_j^2}{2\hat{\alpha}_j^4} \right) \sum_{i=1}^N q_{i,j} + s_j \left( \psi^{(1)}(\hat{\alpha}_j) + \frac{1}{2} \sigma_j^2 \psi^{(3)}(\hat{\alpha}_j) \right)}$$

    $\sigma_k^2 \leftarrow \left( s_j \psi^{(1)}(\hat{\alpha}_j) + \frac{1}{2\hat{\alpha}_j^2} \sum_{i=1}^N q_{i,j} \right)^{-1}$

  **until** *convergence*

  $q_{i,j} \leftarrow \exp\left( \psi(\zeta_j) - \psi\left( \sum_{k=1}^K \omega_k + N \right) - \frac{1}{2} \log(2\pi) + \frac{1}{2} \log \hat{\alpha}_j - \frac{\sigma_j^2}{4\hat{\alpha}_j^2} + \hat{\alpha}_j + \hat{\alpha}_j(\psi(\gamma_j) - \log(\lambda_j)) + (\hat{\alpha}_j - 1) \log x_i - \hat{\alpha}_j \frac{\gamma_j}{\lambda_j} x_i \right)$

  $q_{i,j} \leftarrow \frac{q_{i,j}}{\sum_{k=1}^K q_{i,k}}$

**until** *change in ELBO falls below some prescribed threshold*

the conditional posterior of $\boldsymbol{\pi}$ is Dirichlet$(\zeta_1, ..., \zeta_K)$ with $\zeta_k = \omega_k + \sum_{i=1}^N \mathbb{I}_{z_i=k}$, while the optimal variational distributions of $\boldsymbol{\pi}$ in both `VI-1` and `VI-2` are Dirichlet$(\zeta_1, ..., \zeta_K)$ with $\zeta_k = \omega_k + \sum_{i=1}^N \mathbb{E}_q[\mathbb{I}_{z_i=k}]$. A similar parallel exists between the conditional posterior distribution and variational distribution of $\beta_k$ and the variational distribution of $\mu_k$. The same is not true for $\alpha_k$ and $z_i$, however, due to complications from the term $\log \Gamma(\alpha_k)$.

| Parameter | Conditional posterior | Variational distribution | |
|---|---|---|---|
| | Gibbs | `VI-1` | `VI-2` |
| $\boldsymbol{\pi}$ | Dirichlet$(\zeta_1, ..., \zeta_K)$ where<br>$\zeta_k = \omega_k + \sum_{i=1}^N \mathbb{I}_{z_i=k}$ | Dirichlet $(\zeta_1, ..., \zeta_K)$ where<br>$\zeta_k = \omega_k + \sum_{i=1}^N \mathbb{E}_q[\mathbb{I}_{z_i=k}]$ | Dirichlet $(\zeta_1, ..., \zeta_K)$ where<br>$\zeta_k = \omega_k + \sum_{i=1}^N \mathbb{E}_q[\mathbb{I}_{z_i=k}]$ |
| $\alpha_k$ | $\frac{\mathrm{e}^{r_k \alpha_k} \beta_k^{n_k \alpha_k}}{\Gamma(\alpha_k)^{n_k+s_k}} \prod_{i:z_i=k} x_i^{\alpha_k - 1}$<br>(sampled via ARS) | $\mathcal{N}(\hat{\alpha}_k, \sigma_k^2)$ where<br>$(\hat{\alpha}_k, \sigma_k^2) = \operatorname{argmin} \mathrm{D}_{\mathrm{KL}}(q(\alpha_k) \Vert p(\alpha_k \mid ...))$<br>(via Newton-Raphson) | $\mathcal{N}(\hat{\alpha}_k, \sigma_k^2)$ where<br>$(\hat{\alpha}_k, \sigma_k^2) = \operatorname{argmin} \mathrm{D}_{\mathrm{KL}}(q(\alpha_k) \Vert p(\alpha_k \mid ...))$<br>(via Newton-Raphson) |
| $\beta_k$ for Gibbs & `VI-1`<br>$\mu_k$ for `VI-2` | Gamma$(\gamma_k, \lambda_k)$ where<br>$\gamma_k = c_k + \alpha_k \sum_{i=1}^N \mathbb{I}_{z_i=k}$<br>$\lambda_k = d_k + \sum_{i=1}^N \mathbb{I}_{z_i=k}\, x_i$ | Gamma$(\gamma_k, \lambda_k)$ where<br>$\gamma_k = c_k + \hat{\alpha}_k \sum_{i=1}^N \mathbb{E}_q[\mathbb{I}_{z_i=k}]$<br>$\lambda_k = d_k + \sum_{i=1}^N \mathbb{E}_q[\mathbb{I}_{z_i=k}] x_i$ | Inv-Gamma$(\gamma_k, \lambda_k)$ where<br>$\gamma_k = \xi_k + \hat{\alpha}_k \sum_{i=1}^N \mathbb{E}_q[\mathbb{I}_{z_i=k}]$<br>$\lambda_k = \tau_k + \hat{\alpha}_k \sum_{i=1}^N \mathbb{E}_q[\mathbb{I}_{z_i=k}] x_i$ |
| $z_i$ | $\frac{p(z_i=j \mid \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}_{\setminus i}, \boldsymbol{x})}{\sum_{k=1}^K p(z_i=k \mid \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}_{\setminus i}, \boldsymbol{x})}$ | $\frac{\exp \mathbb{E}_{q_{-(i,k)}}[\log p(\boldsymbol{x} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \boldsymbol{z})]}{\sum_{k=1}^K \exp \mathbb{E}_{q_{-(i,j)}}[\log p(\boldsymbol{x} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, \boldsymbol{z})]}$ | $\frac{\exp \mathbb{E}_{q_{-(i,k)}}[\log p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{z})]}{\sum_{k=1}^K \exp \mathbb{E}_{q_{-(i,j)}}[\log p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{z})]}$ |

Table 3.1: Conditional posterior distributions for the Gibbs sampler algorithm 3 together with the variational distributions for `VI-1` and `VI-2` in algorithms 4 and 5, respectively.

## 3.6 Implementation Details for Variational Inference

We implement `VI-1` and `VI-2` in Python on top of the high-performance tensor computation library Tensorflow (Dean 2016).

Our library gives the practitioner the option of performing stochastic variational inference as well as CAVI. Recall from section 2.3.1 that stochastic variational inference can be constructed from a coordinate-ascent variational inference by resampling $\boldsymbol{x}$ from the full dataset at each iteration (Hoffman et al. 2013). The only algorithmic changes required in order to augment the CAVI algorithms in algorithm 4 and algorithm 5 into SVI are to sample a *subset* (in the machine learning literature this is called a *batch*) of size $N_{\text{batch}}$ from $\boldsymbol{x}$ at each iteration immediately before line 10 and then scale each sum over $i = 1, ..., N$ as if it were a sum over the whole dataset. That is,

1. replace every instance of $\sum_{i=1}^{N} q_{i,j}$ with $\frac{N}{N_{\text{batch}}} \sum_{i=1}^{N} q_{i,j}$,
2. replace every instance of $\sum_{i=1}^{N} q_{i,j} x_i$ with $\frac{N}{N_{\text{batch}}} \sum_{i=1}^{N} q_{i,j} x_i$, and
3. replace every instance of $\sum_{i=1}^{N} q_{i,j} \log x_i$ with $\frac{N}{N_{\text{batch}}} \sum_{i=1}^{N} q_{i,j} \log x_i$.

To perform standard CAVI, the user simply needs to set a parameter called `BATCH_SIZE` to $N$, the size of the full dataset.

Following the suggestion of Hoffman et al., we increase the batch size as training progresses. This gives us both the speed of SVI and the accuracy of CAVI. To ensure that none of the mixing weights in the SVI phase collapse to zero, we set a very informative prior on $\boldsymbol{\pi}$ by taking $\omega_k$ to be very large. By experimentation, we found that $w_k = 10,000$ for all $k = 1, ..., K$ works well.

The unnormalised values for $q_{i,j}$ on line 10 of algorithms 4 and 5 can easily cause underflow errors during division (when the denominator is very close to zero) and overflow errors during exponentiation[3]. To avoid this, we employ the exponential normalisation

3. Arithmetic underflow errors occur when the result of a computer calculation is smaller than the smallest possible value supported by the working datatype. Similarly, arithmetic overflow errors occur when the result of a calculation is too large for a datatype. In most high-level programming languages such as Python (but not low level languages such as C) the result of an underflow operation is a value of 0, and division by zero is represented by a value of `NaN` (`Not a Number`). Any numerical operation (multiplication, addition, and so on) involving a `NaN` value itself returns `NaN`. Therefore, underflow errors on denominators of fractions quickly propagate throughout our program and cause unexpected behaviour. Overflow returns a value of `Inf` which has a similar effect effect to `NaN` except that division by `Inf` results in zero whereas most other operations return another `Inf`.

trick:

$$\frac{\exp y_{i,j}}{\sum_{k=1}^{K} \exp y_{i,k}} = \frac{\exp(y_{i,j} - b_i)}{\sum_{k=1}^{K} \exp(y_{i,k} - b_i)}$$

where we let $y_{i,j}$ denote the the value inside the exponential on line 10. We set the normalisation term $b_i$ to the maximum of $y_{i,k}$ for $k = 1, ..., K$. This both avoids overflow caused by infinite exponentiation and division by zero caused by underflow.

We also found it necessary to use 64-bit floating-point precision in order to evaluate the Newton-Raphson step for $\hat{\alpha}_j$, since the denominator can become very small and cause underflow errors.

# Chapter 4

# Experimental Comparison

In this chapter, we compare the performance of the three algorithms derived in chapter 3. We are interested in their speed and quality-of-fit. There are, broadly speaking, two aspects to the quality-of-fit. The first regards the posterior distribution. For this, we take MCMC as the benchmark since it is well-studied and understood to converge asymptotically to the true posterior. The second regards the *posterior predictive distribution* (PPD)[1] which can be intuited as the probability density function of a new observation $\tilde{x}$ 'averaged out' over all possible values of the parameters. Our results show that the mean-shape parameterised variational inference algorithm `VI-2` produces PPDs that are much more consistent with the Gibbs sampler than those produced by `VI-1`. Furthermore, `VI-2` produces more plausible posterior predictive densities than either the Gibbs sampler or `VI-1`; however, in this respect, the comparison between `VI-2` and the Gibbs sampler is not entirely fair since the Gibbs sampler uses the shape-rate parameterisation while `VI-2` uses the mean-shape parameterisation.

## 4.1 Datasets and Methodology

At this point, it is useful to distinguish between the number of components in the dataset and the number of components in the model. Let $K_{\text{data}}$ denote the former and $K_{\text{model}}$ the latter. We consider a number of synthetic (programatically generated) datasets as well as a dataset of Australian rainfall. In this thesis, we do not attempt to infer the optimal $K_{\text{model}}$ from the data - we leave this for future works. Instead, since we know the underlying value $K_{\text{data}}$ used to generate our synthetic datasets, we set $K_{\text{model}} = K_{\text{data}}$. For

1. The posterior predictive distribution is also a posterior predictive density when the random variable of interest is continuous; we use 'PPD' to represent both cases.

the Australian rainfall data, we take $K_{\text{model}} = 5$ in line with the rainfall literature.

We test our models on two datasets. First, following Marron and Wand (1992), we generate a synthetic from a mixture of gamma distributions in which the means of the components are evenly spaced at positive integer values with equal variances and mixture weights. We generate 1000 samples for each component for up to 20 components, so $N = 1000 \times K_{\text{data}}$. The PDF one of these datasets with $K_{\text{data}} = 2$ is shown in fig. 4.1. After we generate the dataset, we fit each of the models - Gibbs sampler, `VI-1`, and `VI-2`- and then apply our evaluation metrics. Furthermore, we also test our models on a synthetic dataset with *randomised* component means, variances, and weights. These datasets are designed to challenge our models under conditions that are as difficult or more difficult than those they would typically encounter.

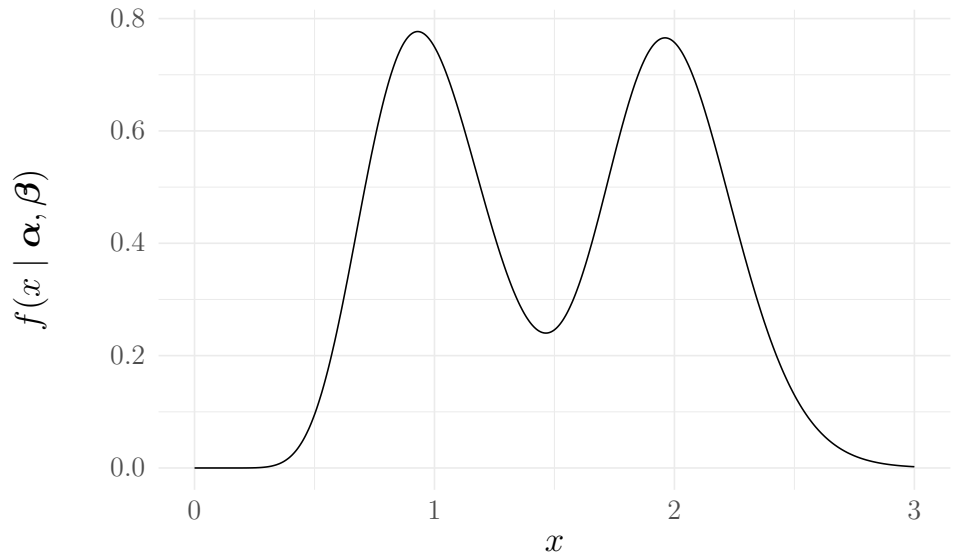



Figure 4.1: A PDF that we use to generate one of our synthetic datasets. Here $K_{\text{data}} = 2$, but the extension to $K_{\text{data}} = 3, 4, ...$ simply consists of $K_{\text{data}}$ components with equal weight and variance and evenly spaced means at positive integers $1, 2, ..., K_{\text{data}}$.

We also test our models on a dataset of daily Australian rainfall collated by Bertolacci et al. (2019) from the Australian Bureau of Meteorology using Bertolacci's `R` package `bomdata`. Although the dataset contains measurements for 17,606 different *sites* (locations), we focus on four major dam sites: Woronora, Wellington, Serpentine, and North Dandalup. Data is missing for some days, while for others zero rainfall was observed. We ignore missing data and, since the gamma distribution has *strictly* positive support, we also ignore days with zero rainfall. This leaves 1066, 1698, 1798, and 6064 observations at each site, respectively.

We show posterior predictive density plots for each dataset-model pair. The PPD is the distribution of a new, unseen observation. We derive it by marginalising $p(\tilde{x} \mid \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$

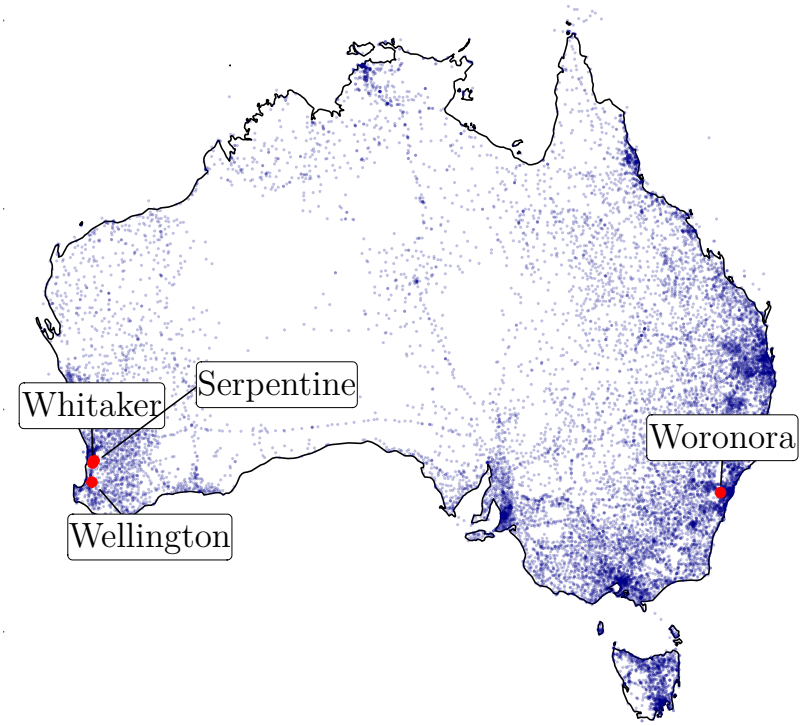



Figure 4.2: Sites recorded in the BOM rainfall dataset.

over the parameters $\boldsymbol{\pi}, \boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ [2]. Denoting the parameter space for a gamma mixture model by $\Theta$, we can approximate the PPD of a new observation $\tilde{x}$ using the Monte Carlo integral

$$\begin{aligned} p(\tilde{x} \mid \boldsymbol{x}) &= \int_{\Theta} p(\tilde{x} \mid \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta} \mid \boldsymbol{x}) d\boldsymbol{\pi} d\boldsymbol{\alpha} d\boldsymbol{\beta} \\ &\approx \frac{1}{T} \sum_{t=1}^{T} p(\tilde{x} \mid \boldsymbol{\pi}^{[t]}, \boldsymbol{\alpha}^{[t]}, \boldsymbol{\beta}^{[t]}) \end{aligned}$$

where $\boldsymbol{\pi}^{[t]}, \boldsymbol{\alpha}^{[t]}$, and $\boldsymbol{\beta}^{[t]}$ are samples from the posterior $p(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta} \mid \boldsymbol{x})$ for $t = 1, .., T$. For MCMC, we obtain these samples directly from the output of the algorithm. For `VI-1` and `VI-2`, we draw independent samples from the variational distributions in eqs. (3.26) to (3.28) for `VI-1`, and eqs. (3.45) to (3.47) for `VI-2`. Since the variational distribution for $\alpha_j$ is normal, it has support on the entire real line. This is problematic as the parameters of a gamma distribution are required to be strictly positive. We instead sample $\alpha_j$ from a positive *truncated* normal distribution.

We also present 90% posterior predictive intervals (PPI). The PPI is the interval in which the probability of $\tilde{x}$ occurring is 90%, and the probabilities of $\tilde{x}$ begin above or below the interval are both 5%. In practise, we approximate it by calculating the probability

2. In probability and statistics, to marginalise out a variable means to 'integrate it out' over its distribution.

density $p(\tilde{x} \mid \boldsymbol{\pi}^{[t]}, \boldsymbol{\alpha}^{[t]}, \boldsymbol{\beta}^{[t]})$ for $t = 1, ..., T$ and a given $\tilde{x}$, then sorting these values, and selecting the 5th and 95th percentiles.

For the synthetic datasets, we have the benefit of knowing the parameters of the underlying data-generating distributions. We can, therefore, measure the quality of each posterior predictive distribution using the integrated absolute difference between itself and the true (data-generating) distribution. This value is called the integrated absolute error (IAE) and it is given by

$$\text{IAE} = \int_0^\infty |p(\tilde{x} \mid \boldsymbol{x}) - p(\tilde{x} \mid \boldsymbol{\pi}_{\text{true}}, \boldsymbol{\alpha}_{\text{true}}, \boldsymbol{\beta}_{\text{true}})| d\tilde{x}$$

where $p(x \mid \boldsymbol{\pi}_{\text{true}}, \boldsymbol{\alpha}_{\text{true}}, \boldsymbol{\beta}_{\text{true}})$ is the density of the true distribution. We approximate this integral using Riemann integration. Another common error metric is the integrated squared error ISE which replaces the absolute value above with a power of two. The ISE penalises large deviations more harshly than the IAE, but the latter has the advantage of being easily interpretable as the area between the density curves. See fig. 4.3 for a visual illustration.

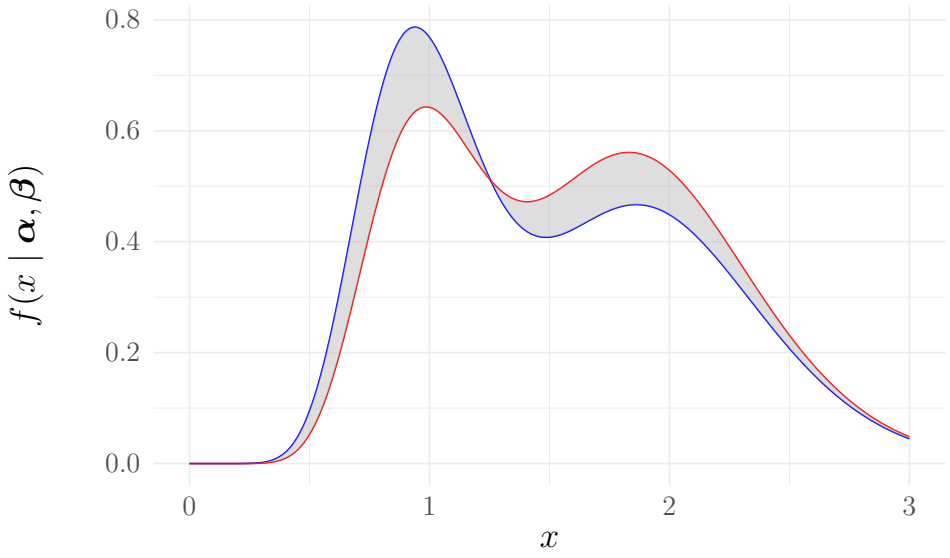



Figure 4.3: Examples probability density functions for two arbitrary mixtures of gamma distributions: one in red and one in blue. The area of the shaded region is equal to the integrated absolute error (IAE) between these distributions.

In MCMC algorithms, it is common to practise *burn-in* (also called warmup) by discarding a number of initial samples. This effectively restarts the chain with the better (converged) initial parameters found at the end of the burn-in phase. For all our trials, we burn-in the Gibbs sampler for 2000 iterations. All runtimes listed include this burn-in time.

## 4.2 Results

### 4.2.1 Examination of posterior inference on synthetic dataset with $K_{\text{data}} = 2$

First, we examine posterior inference on synthetic dataset generated from the simplest possible mixture of $K_{\text{data}} = K_{\text{model}} = 2$ components with evenly-spaced means at positive-integer values, variances equal to 0.05, and mixing weights equal to $K_{\text{data}}^{-1}$. Figure 4.5 displays a correlogram of the MCMC output. The most strongly correlated variables are $\alpha_k$ and $\beta_k$ which have correlation 0.991 and 0.998 for $k = 1$ and $k = 2$ respectively[3].

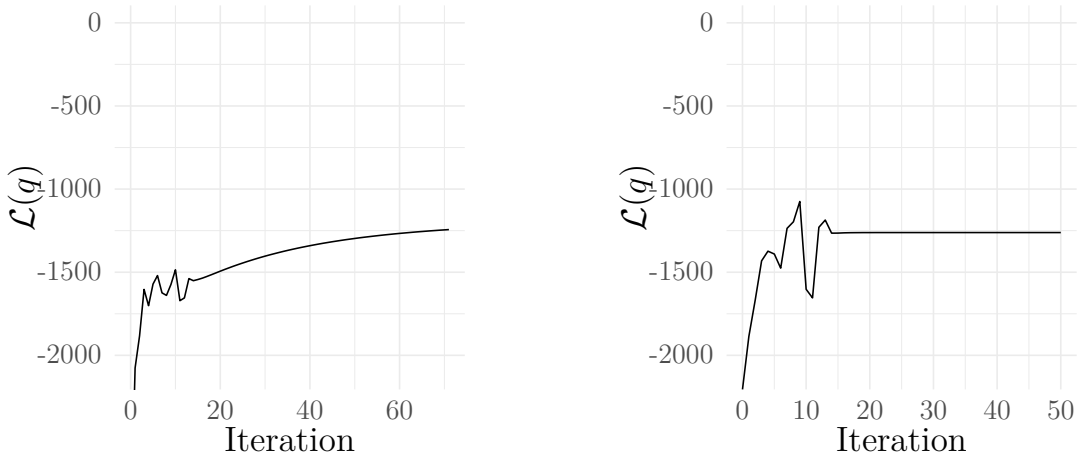
The posterior densities for each of the parameters are displayed in figs. 4.6 and 4.7. A visual inspection of subfigures 4.6a and 4.6b supports the assertion in section 3.1.3 that, for large $N$, the posterior distribution of $\alpha_k$ and $\beta_k$ approaches a multivariate normal distribution with a non-diagonal covariance matrix. Similarly, subfigures 4.7a and 4.7b provide mild support to the hypothesis that the posterior of $\mu_k$ and $\alpha_k$ approaches a (independent) multivariate normal distribution with a diagonal covariance matrix; however we can see that $\mu_k$ and $\alpha_k$ are not *quite* independent - the joint density has an 'egg' shape. As a result, both `VI-1` and `VI-2` underestimate of the variance of the posteriors of $\alpha_k$ and $\beta_k$ relative to the Gibbs sampler; however, `VI-1` underestimates it far more severely than `VI-2`. For example, by MCMC we find the variance of $\alpha_1$ to be 0.466. `VI-2` estimates it at 0.303, but `VI-1` estimates 0.0024 - far lower than the MCMC benchmark. This effect is also visible in the form of narrower posterior predictive intervals (the blue region) in figs. 4.11 to 4.13. Furthermore, a visual inspection of figs. 4.6 and 4.7 shows that `VI-2` captures the relationship between $\alpha_k$ and $\beta_k$ far more faithfully than `VI-1`. This confirms our hypothesis that the mean-shape parameterisation results in a superior variational approximation of the posterior.

Recall that a higher (less negative) ELBO indicates a better fit. Referring to fig. 4.4, we can see that both `VI-1` and `VI-2` converge to almost identical ELBOs after approximately 60 iterations. However, while the ELBO of `VI-1` rises rapidly within the first 10 iterations, it then very slowly converges to its final value over the next 60 iterations as if approaching an asymptote. In contrast, `VI-2` takes just 14 iterations to reach ELBO, after which it immediately stabilises. This shows that `VI-2` not only approximates the posterior better than `VI-1`, but it also converges significantly faster.

We produce the traces in figs. 4.9 and 4.10 by taking the means of each of the parameters

3. Technically, the most strongly correlated variables are $\pi_1$ and $\pi_2$ with correlation $-1$, but this is not of interest since for $K = 2$ they are, by definition, linked directly by the constraint $\pi_1 + \pi_2 = 1$.

$\pi_k$, $\alpha_k$, and $\beta_k$ for $k = 1, ..., K$ under their respective variational distributions. We omit the traces of the the variational parameters $\hat{\alpha}_k$, $\sigma_k^2$, $\gamma_k$, and $\lambda_k$ as these neither have a clear interpretation nor a counterpart in MCMC.




(a) (b)

Figure 4.4: Traces for the evidence lower bounds in eqs. (3.21) and (3.40) of `VI-1` (left) and `VI-2` (right) respectively.

### 4.2.2 Results for synthetic datasets with more components

From table 4.1, we see that `VI-2` is superior to `VI-1` and the Gibbs sampler in terms of both the IAE and execution time. For the case where $K_{\text{model}} = K_{\text{data}} = 14$, `VI-2` achieves a $\times 64$ speedup over the 2000-iteration Gibbs sampler (after the 2000 burn-in iterations).

Subfigure 4.13b reveals a flaw in `VI-2` where multiple components in the approximating density converge into a single component. We term this phenomenon *component degeneracy*, and it means that that the model is not fully exploring the parameter space, but instead getting stuck in a local minimum. The stochastic nature of the Gibbs sampler gives it a small chance of 'jumping out' of such a minimum. The same is not true in variational inference, which lacks any way to 'push' degenerate components apart. Even in stochastic variational inference where the data is randomly sampled at each step, the parameter updates are entirely deterministic. As a consequence, when two components converge to exactly the same parameters, they will never diverge. We discuss potential remedies to this problem in chapter 5.

Interestingly, while `VI-1` fits the PPD much more poorly overall, it does appear to successfully locate all modes and with no degeneracy. However, `VI-1` underestimates the variance of each subsequent component increasingly, severely overestimating the variance for the left-most component, and severely underestimating it for the right-most one. The

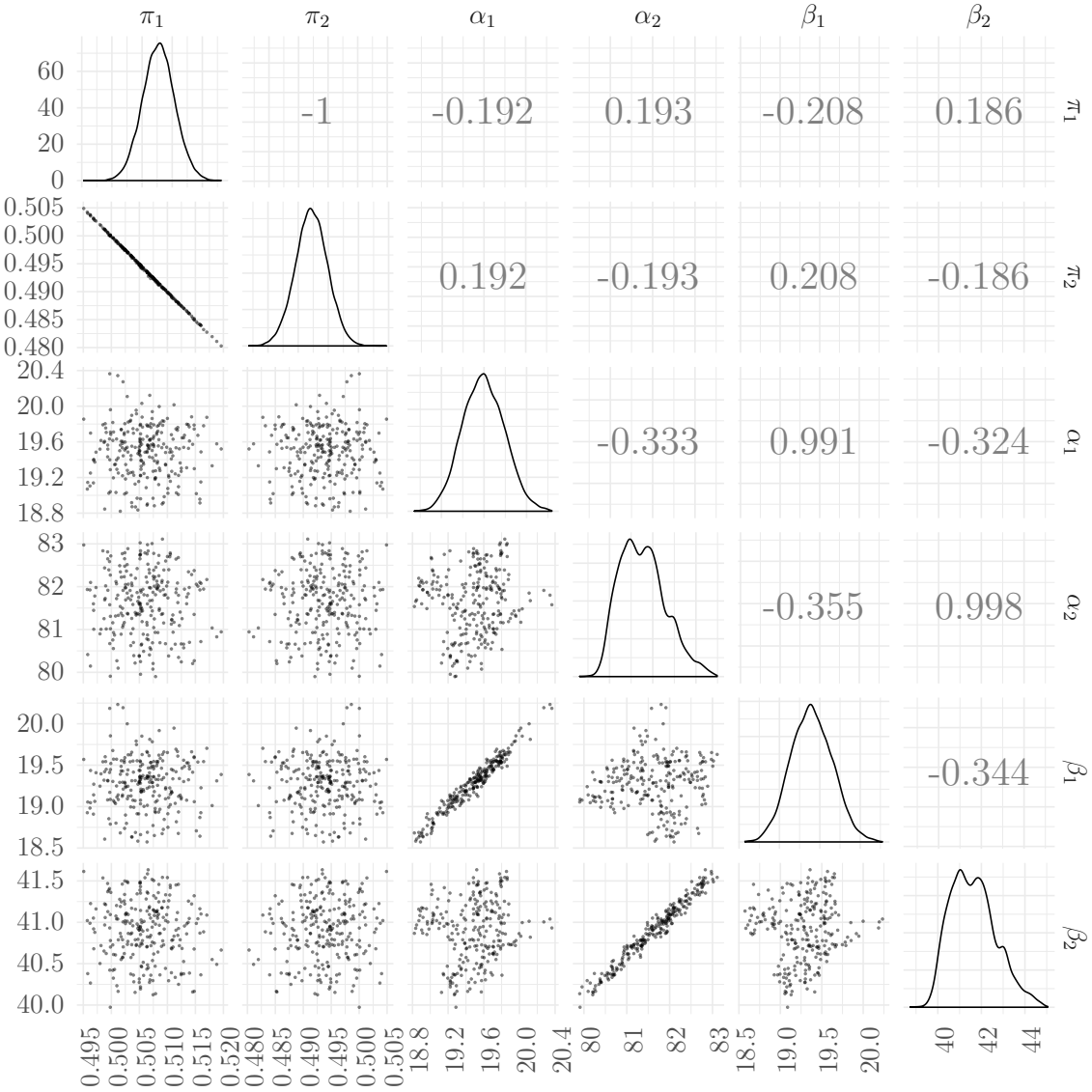



Figure 4.5: Corellogram of samples from the mixture of gammas posterior given by the Gibbs sampler in algorithm 3 for $K_{\text{model}} = K_{\text{data}} = 2$. The upper triangle shows the correlation between parameters, the lower triangle shows the 2D density, and the diagonal shows the marginal density.

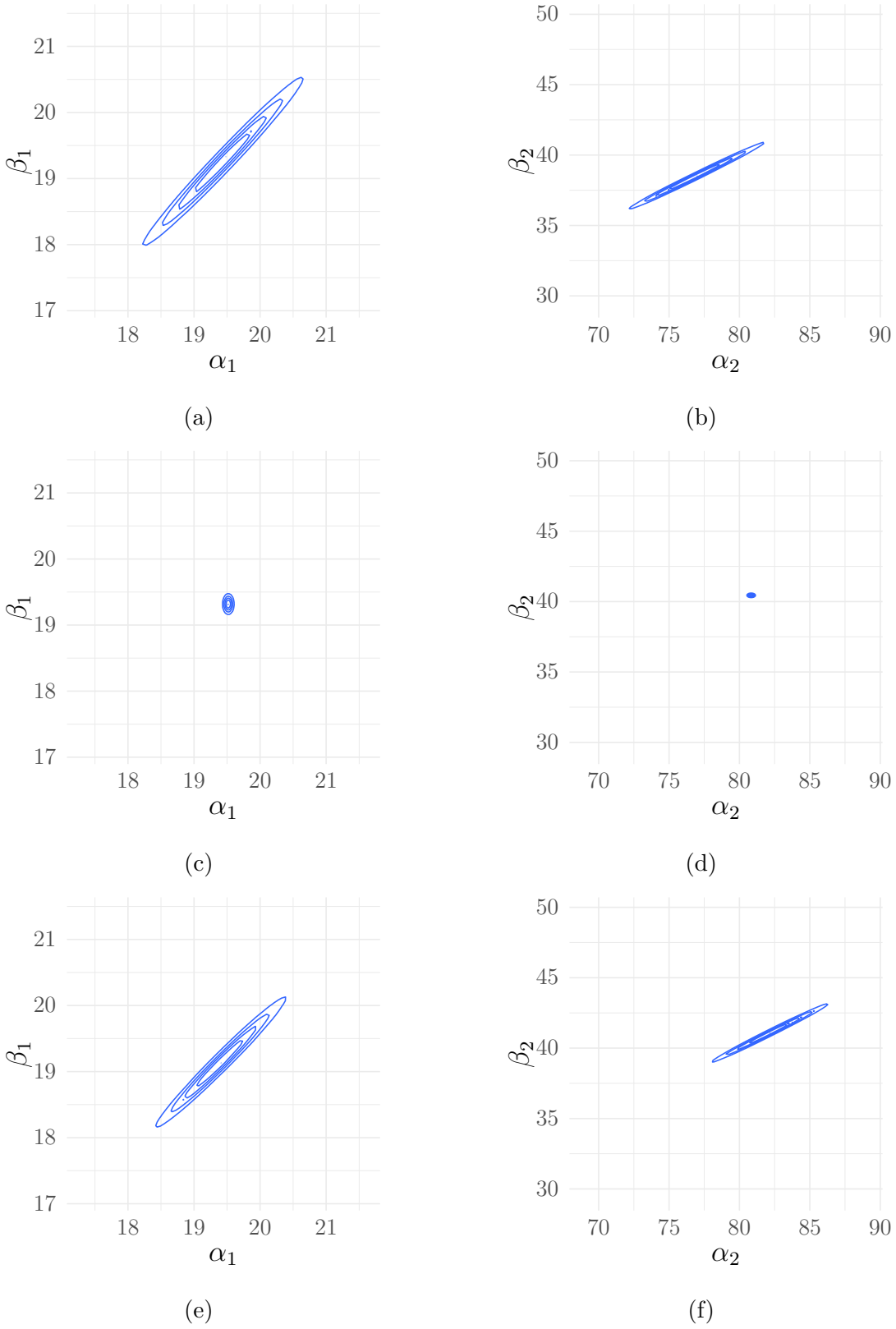



Figure 4.6: Posterior distributions for MCMC (top), `VI-1` (middle), and `VI-2` (bottom) in the shape-rate space. Each level of the contour contains one-fifth of the probability mass. The parameters used to generate the data are $(\alpha_1, \beta_1) = (20, 20)$ and $(\alpha_2, \beta_2) = (80, 40)$.

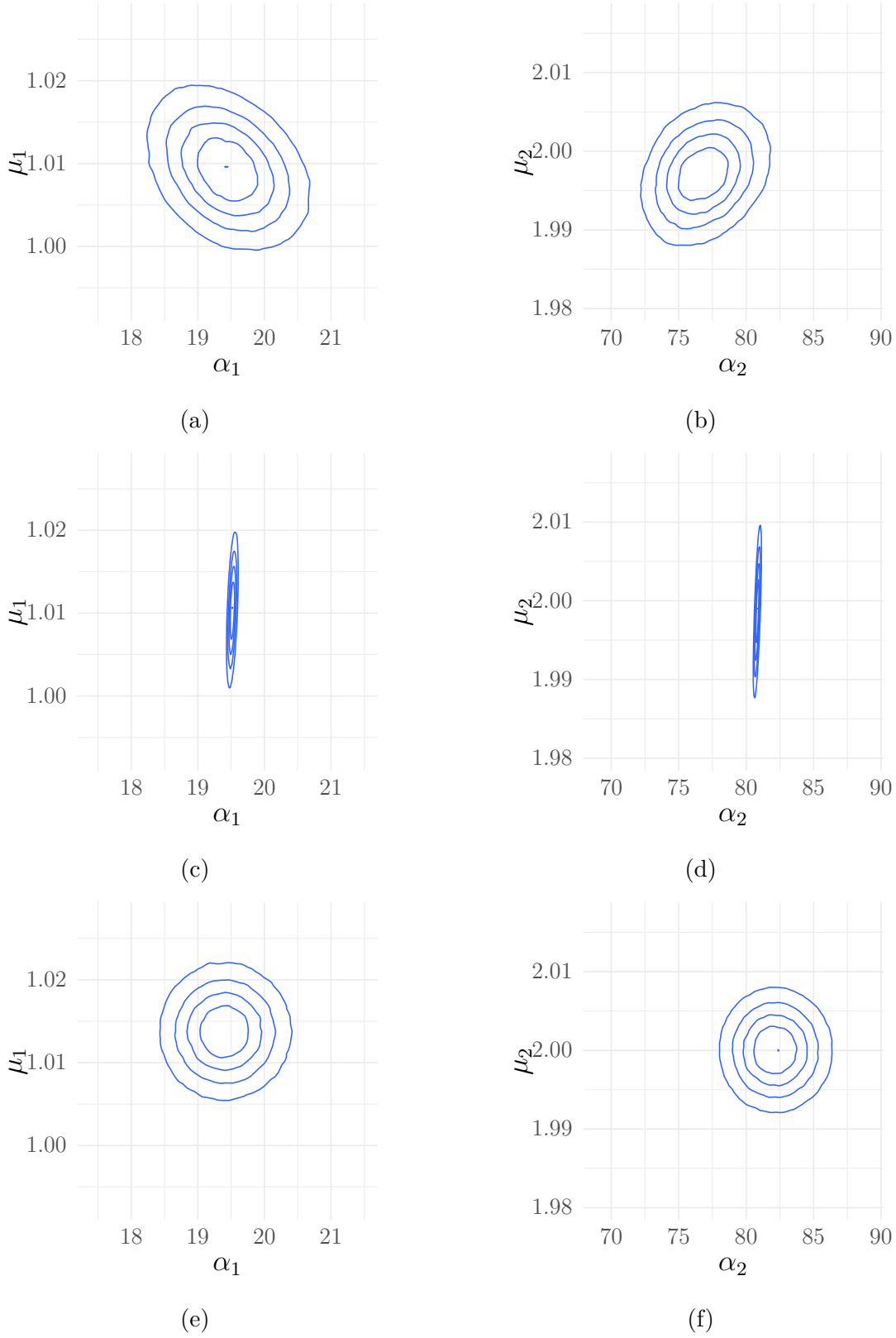



Figure 4.7: The same posterior distributions as in fig. 4.6 for MCMC (top), `VI-1` (middle), and `VI-2` (bottom) but in the shape-mean space. The parameters used to generate the data are $(\mu_1, \alpha_1) = (1, 20)$ and $(\mu_2, \alpha_2) = (2, 40)$.

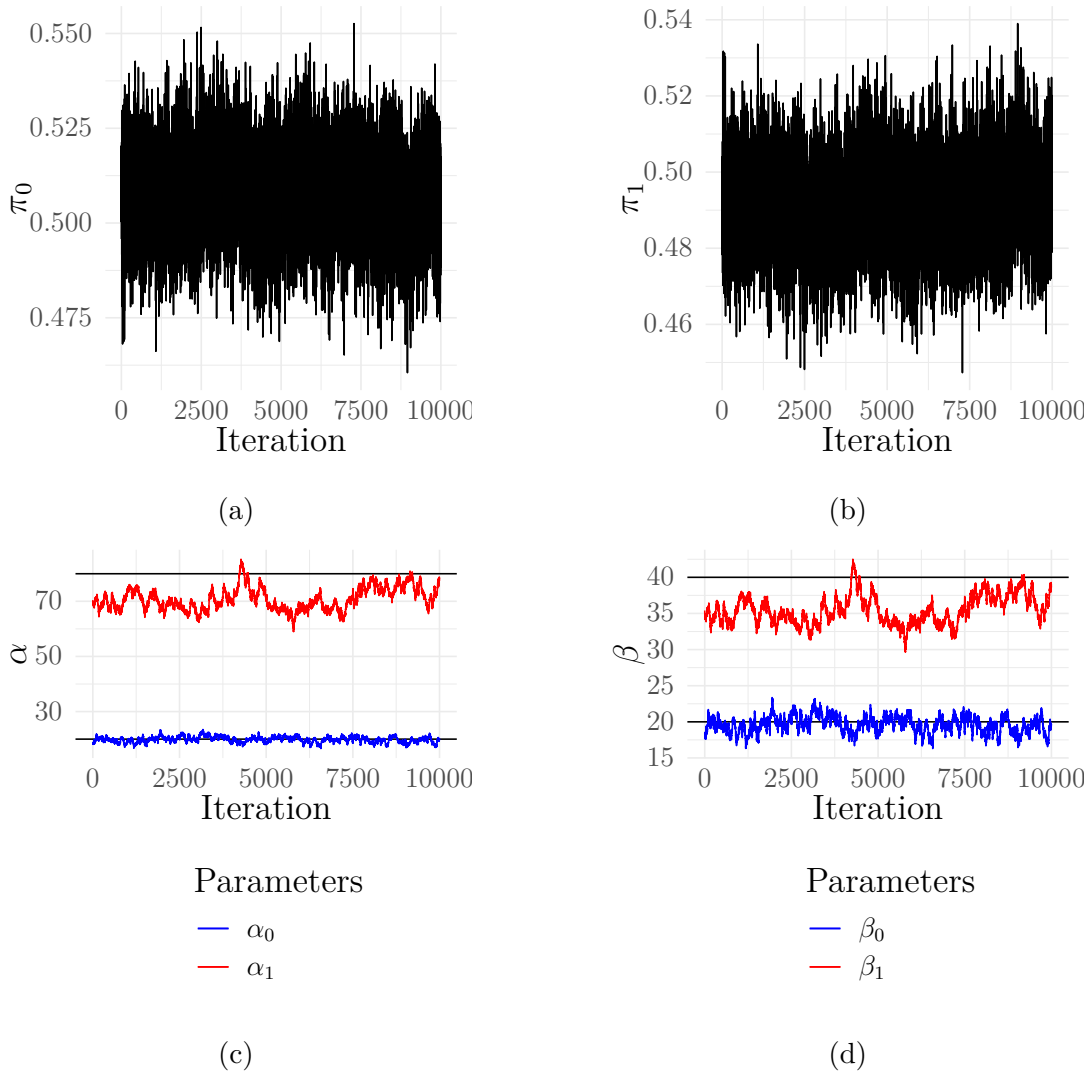



Figure 4.8: Trace plots for the Gibbs sampler.

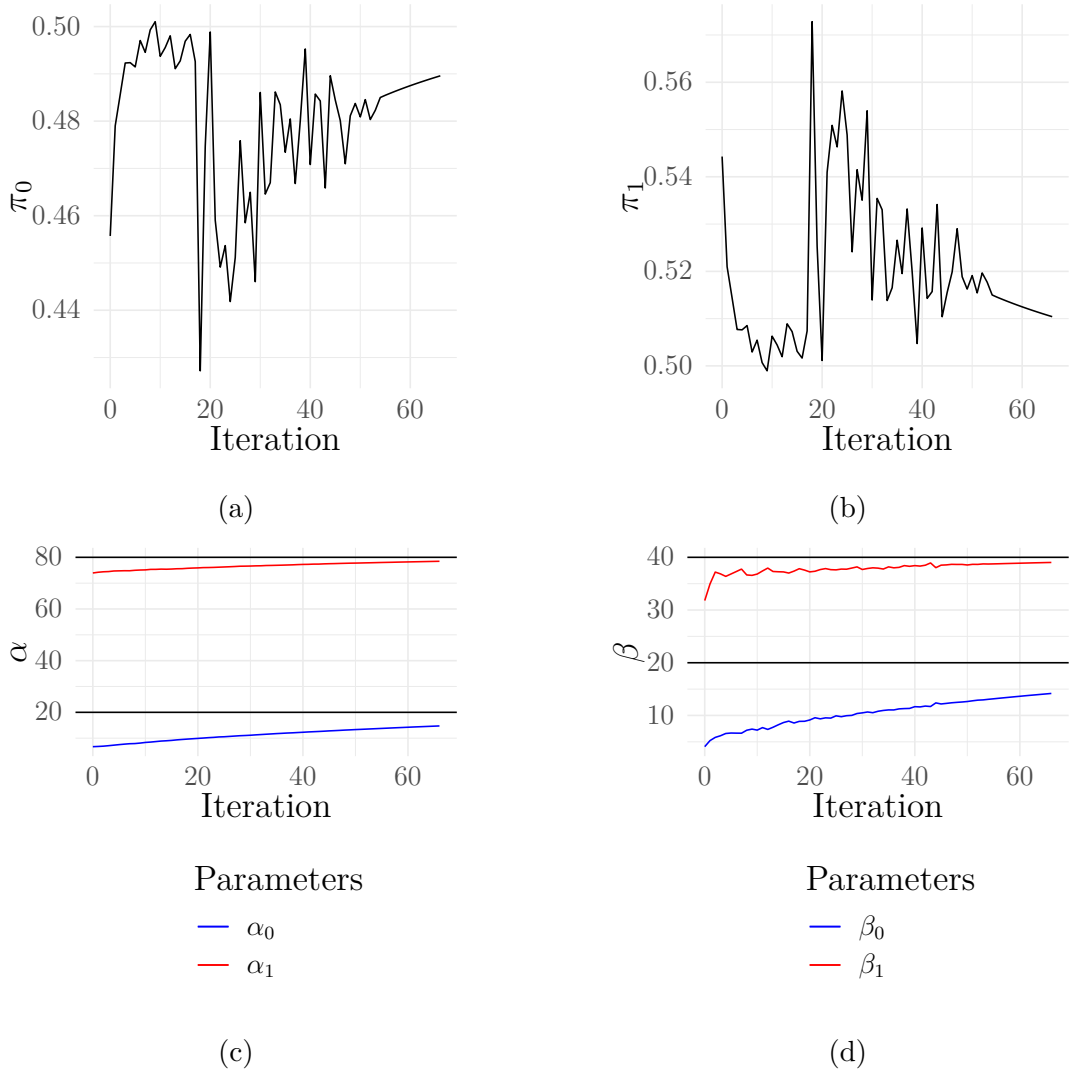



Figure 4.9: Traces for `VI-1`. The true values for $\alpha_k$ and $\beta_k$ are represented are represented by black horizontal lines. The true values for $\pi_1$ and $\pi_2$ are both 0.5.

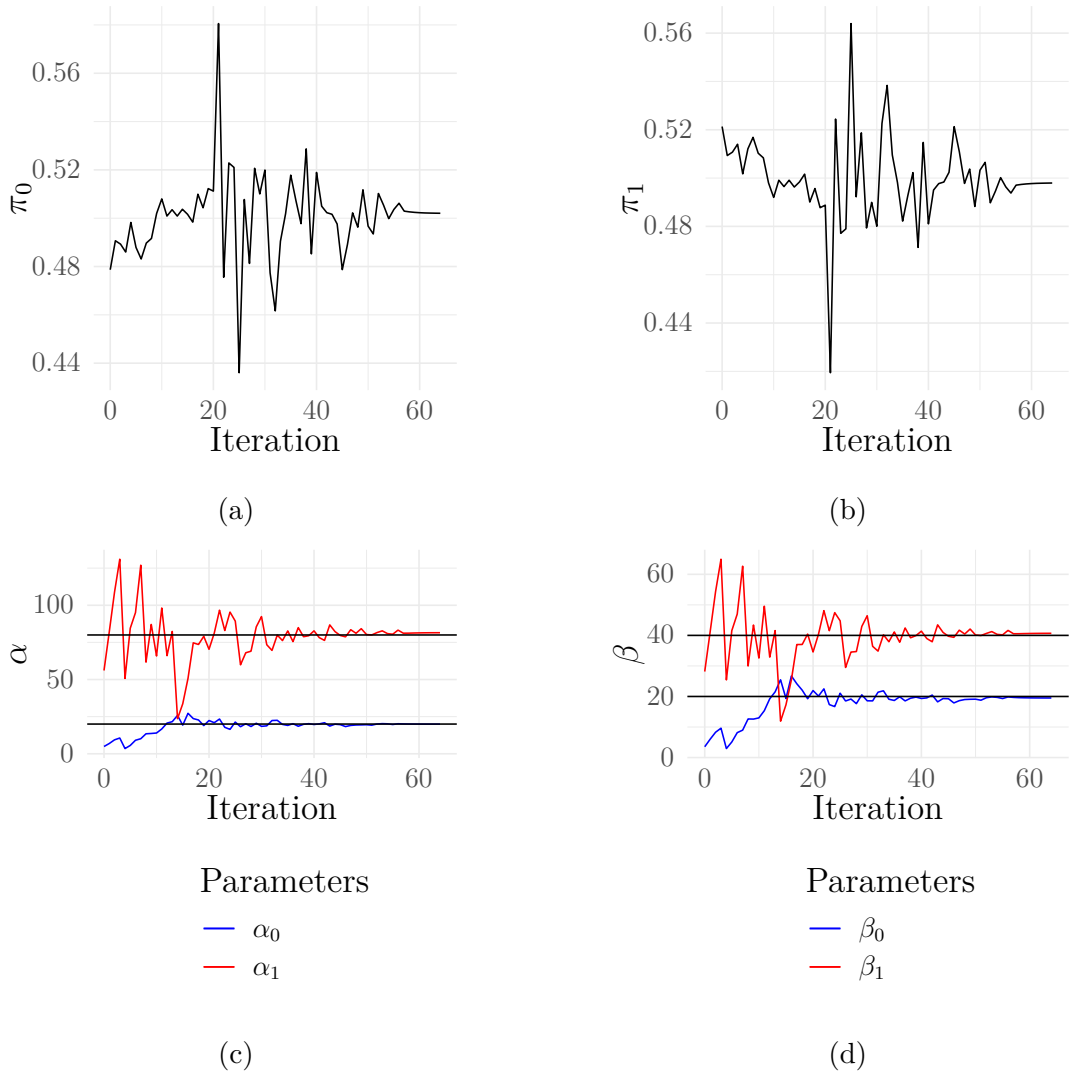
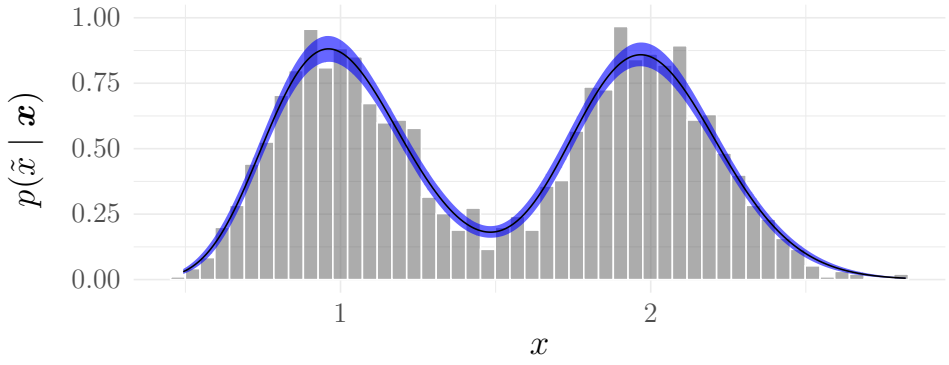



Figure 4.10: The same traces as in fig. 4.9 but for `VI-2`.

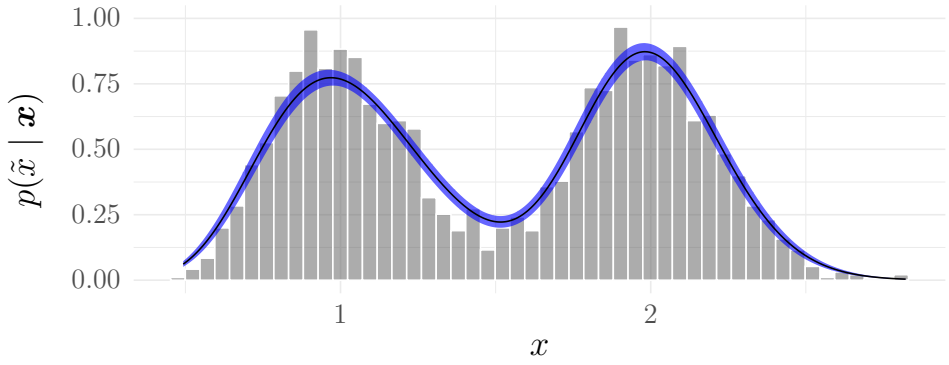
Gibbs sampler could not be tested for models with more than 14 components due to a fatal initialisation bug in the implementation we employed.

| Model | $K_{\text{data}}$ | $K_{\text{model}}$ | Time ($s$) | Iterations | ELBO | IAE |
|---|---|---|---|---|---|---|
| Gibbs | 2 | 2 | 1.292 | 2000 | | 0.063 |
| | 4 | 4 | 3.207 | 2000 | | 0.140 |
| | 6 | 6 | 6.491 | 2000 | | 0.266 |
| | 8 | 8 | 11.226 | 2000 | | 0.365 |
| | 10 | 10 | 17.241 | 2000 | | 0.536 |
| | 12 | 12 | 24.163 | 2000 | | 0.482 |
| | 14 | 14 | 31.380 | 2000 | | 0.491 |
| `VI-1` | 2 | 2 | 0.143 | 110 | -1272 | 0.032 |
| | 4 | 4 | 0.219 | 98 | -5456 | 0.123 |
| | 6 | 6 | 0.182 | 71 | -10815 | 0.157 |
| | 8 | 8 | 0.214 | 31 | -16804 | 0.172 |
| | 10 | 10 | 0.296 | 21 | -23311 | 0.183 |
| | 12 | 12 | 0.388 | 23 | -30255 | 0.186 |
| | 14 | 14 | 0.519 | 22 | -37564 | 0.192 |
| | 16 | 16 | 0.543 | 23 | -45047 | 0.194 |
| | 18 | 18 | 0.835 | 30 | -52903 | 0.195 |
| | 20 | 20 | 0.640 | 41 | -61449 | 0.213 |
| `VI-2` | 2 | 2 | 0.105 | 23 | -1341 | 0.042 |
| | 4 | 4 | 0.179 | 23 | -5362 | 0.028 |
| | 6 | 6 | 0.205 | 20 | -10438 | 0.042 |
| | 8 | 8 | 0.246 | 24 | -16087 | 0.048 |
| | 10 | 10 | 0.308 | 24 | -22239 | 0.042 |
| | 12 | 12 | 0.384 | 29 | -28971 | 0.035 |
| | 14 | 14 | 0.493 | 32 | -35940 | 0.046 |
| | 16 | 16 | 0.608 | 38 | -43197 | 0.046 |
| | 18 | 18 | 0.568 | 47 | -50746 | 0.039 |
| | 20 | 20 | 0.645 | 56 | -59837 | 0.102 |

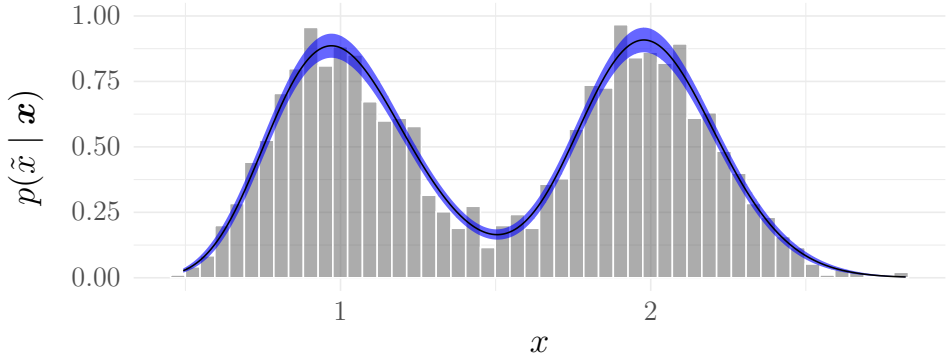
Table 4.1: Performance comparison between the Gibbs sampler, `VI-1`, and `VI-2`. Note that some entries are missing for the Gibbs sampler since the software we employed failed to initialise for $K_{\text{model}} > 14$.




(a) Gibbs

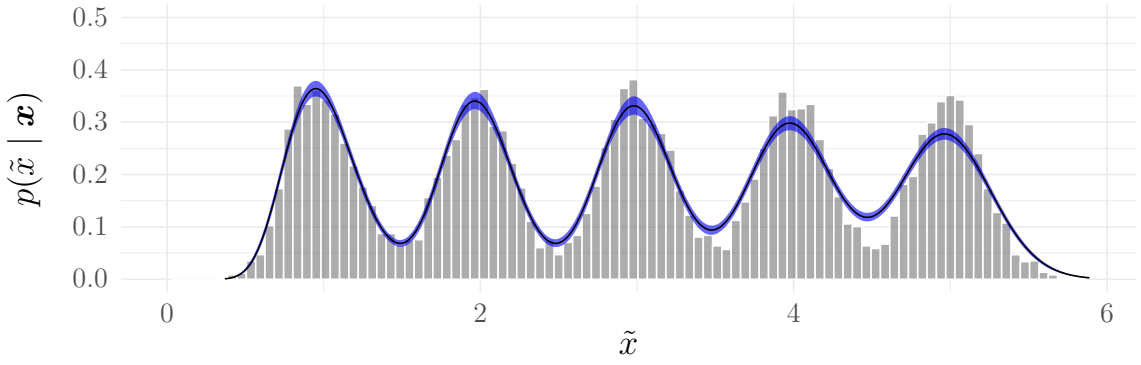



(b) `VI-1`

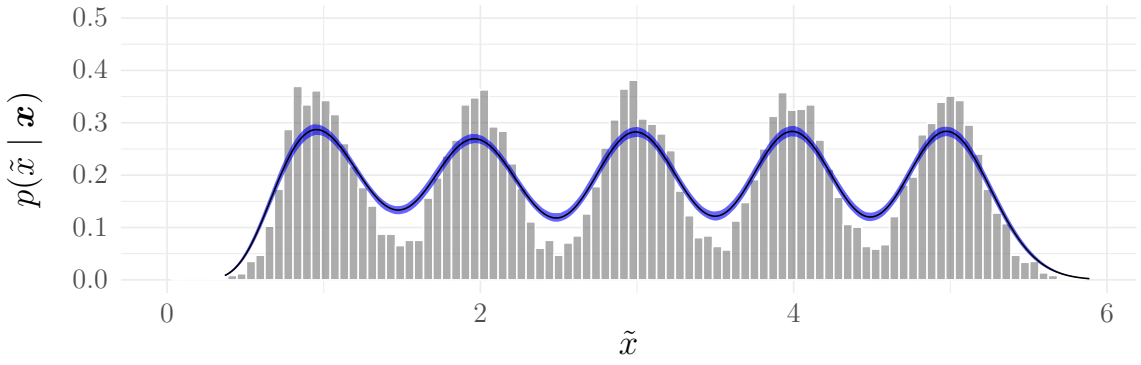



(c) `VI-2`

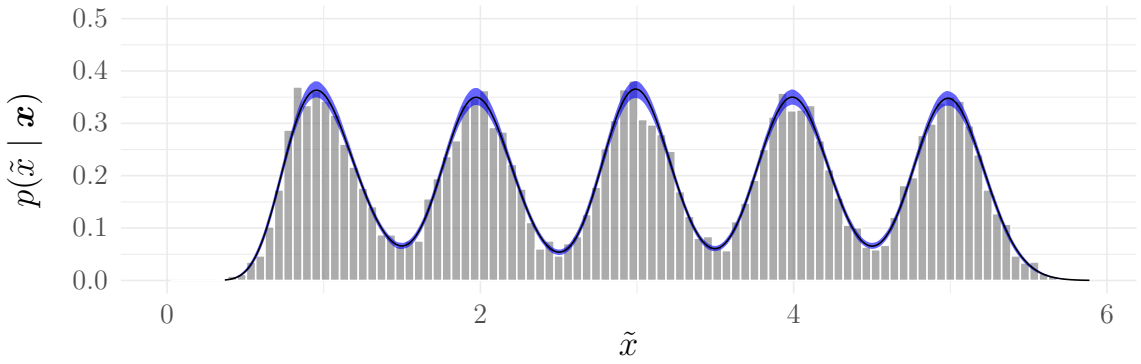
Figure 4.11: The posterior predictive densities from MCMC (top), `VI-1` (middle), and `VI-2` (bottom) for a synthetic dataset with $K_{\text{model}} = K_{\text{data}} = 2$. The black lines represents the PPDs while the blue regions represent the 90% posterior predictive intervals. The data is displayed in the background as a histogram.




(a) Gibbs

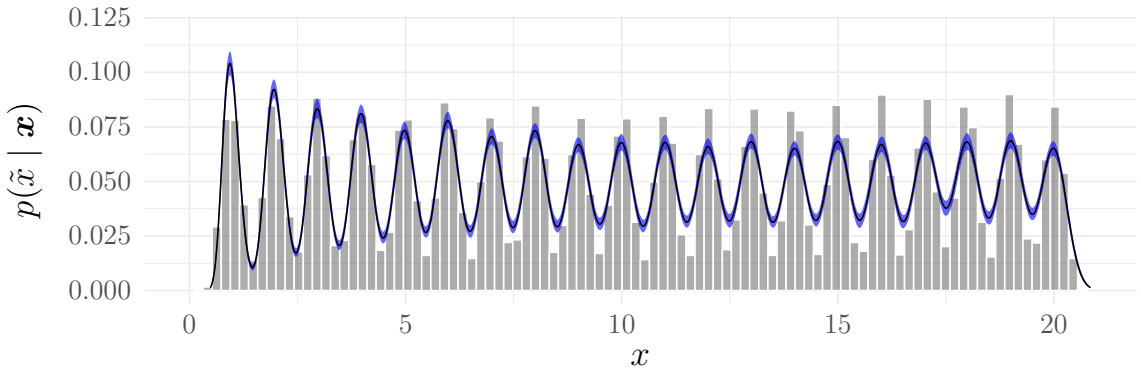



(b) `VI-1`

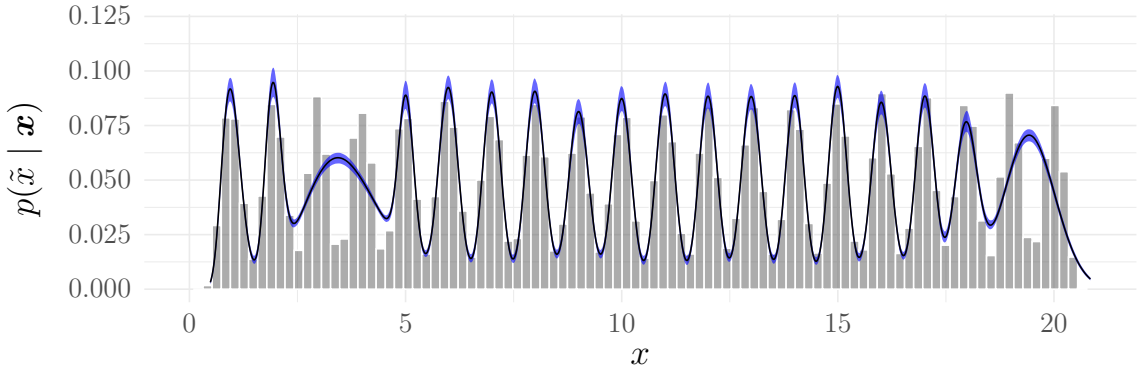



(c) `VI-2`

Figure 4.12: Results from fitting a dataset with $K_{\text{model}} = K_{\text{data}} = 5$.




(a) `VI-1`




(b) `VI-2`

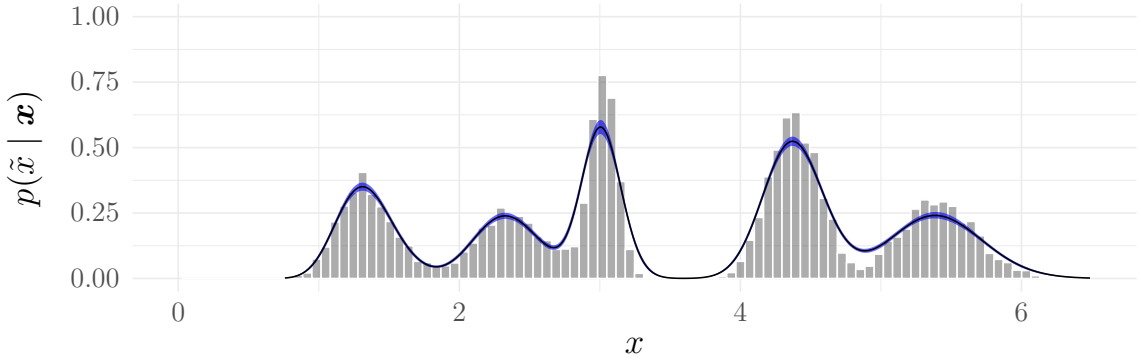
Figure 4.13: Results from fitting a dataset with $K_{\text{model}} = K_{\text{data}} = 20$.

### 4.2.3 Results for irregular synthetic datasets

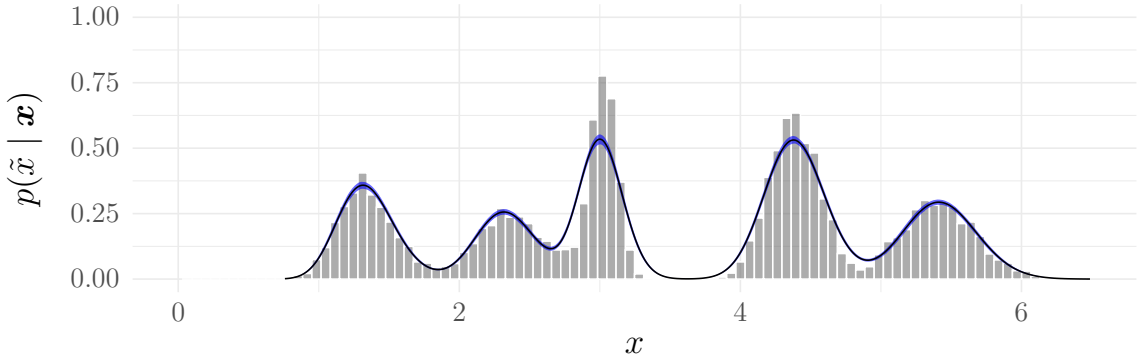
In this section, we evaluate our models on two synthetic datasets with irregular variances, mixing weights, and spacings between modes. In both cases, `VI-2` significantly outperforms `VI-1`. While were not able to test the Gibbs sampler on more than 14 components, for the 5-component case in fig. 4.14 it is clear that, similarly to `VI-1`, the Gibbs sampler fits the right-most modes (with higher mean) more poorly than the left-most modes (with lower mean).

### 4.2.4 Application to rainfall data

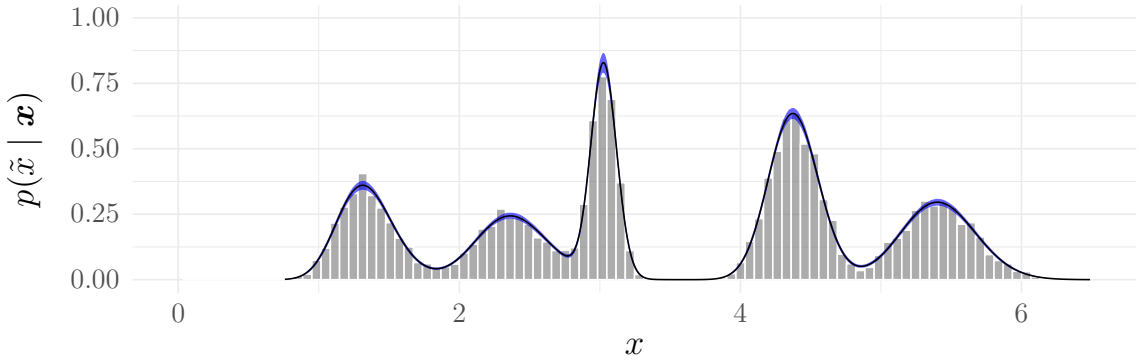
We now apply our models to Australian rainfall data from the Woronora, Wellington, Serpentine, and North Dandalup dams. Figures 4.16 and 4.17 display the PPDs. The rainfall data appears to be multimodal with one dominant mode and a number of minor ones. In all cases, the algorithms are able to successfully capture the dominant mode; however, `VI-1` consistently fails to capture the minor ones. `VI-2` appears to do better than both `VI-1` and the Gibbs sampler in this respect. This is especially noticeable in the




(a) Gibbs



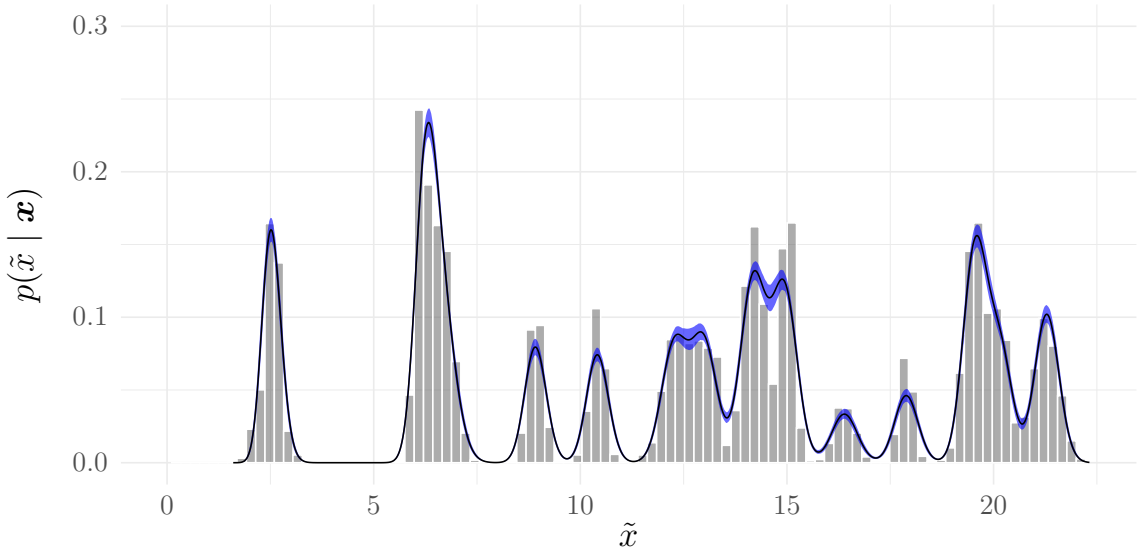

(b) `VI-1`



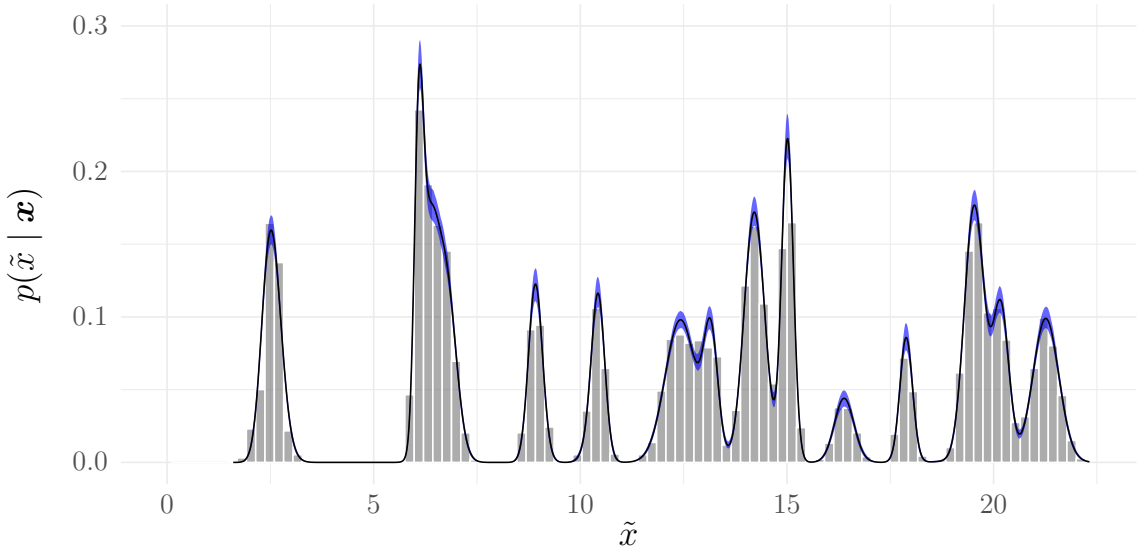

(c) `VI-2`

Figure 4.14: The posterior predictive densities of MCMC (top), `VI-1` (middle), and `VI-2` (bottom) on an irregular synthetic dataset with $K_{\text{model}} = K_{\text{data}} = 5$. The average execution time on this dataset was 561 ms for `VI-1` and 569 ms for `VI-2`.

case of the Wellington dam (subfigure 4.16f) where `VI-2` clearly achieves higher fidelity and successfully locates three distinct modes while Gibbs and `VI-1` only locate one. Also, recall that a key challenge in rainfall modelling is capturing the heavy tail. Upon visual inspection, it appears that all three of our models succeed in this respect.

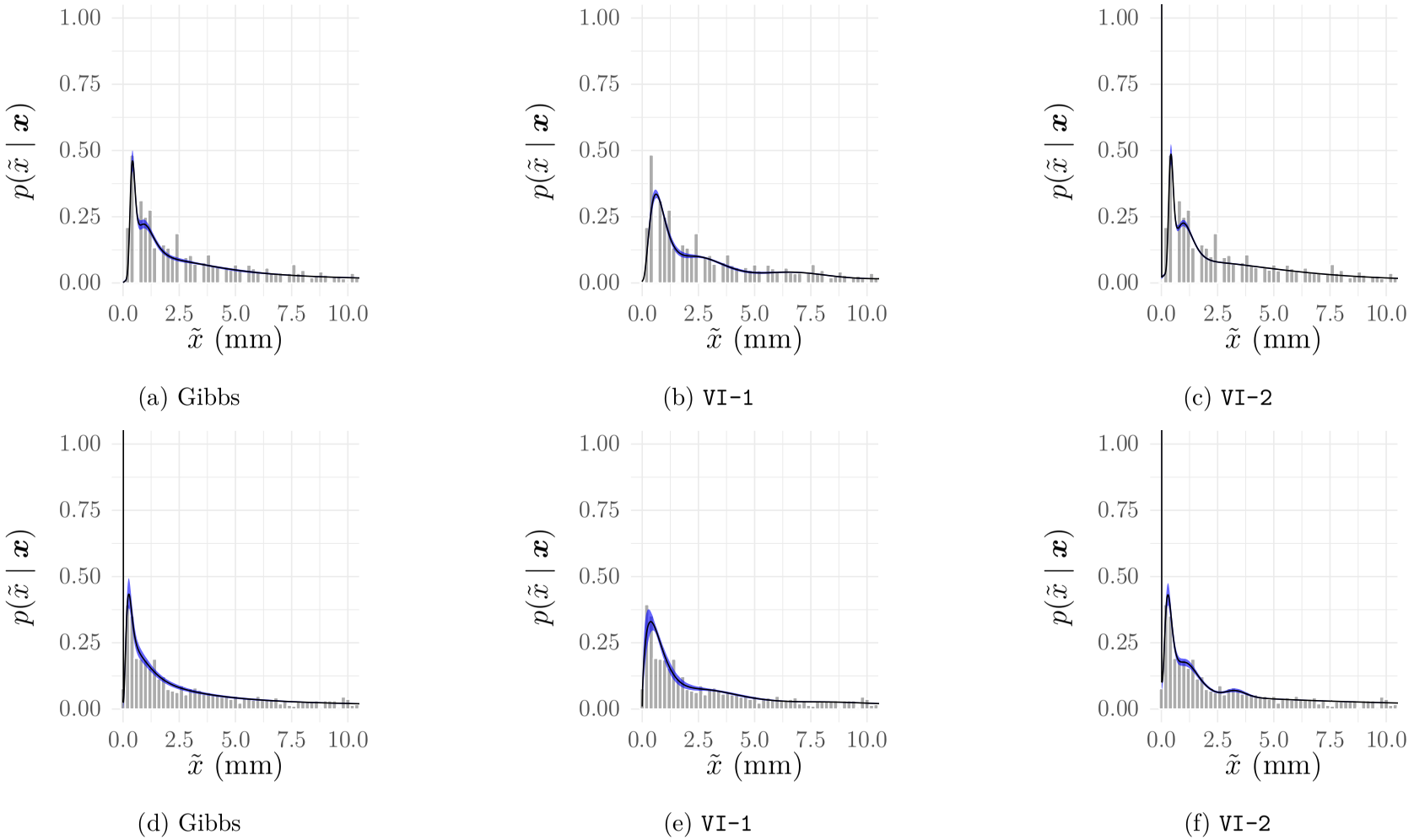



(a) `VI-1`




(b) `VI-2`

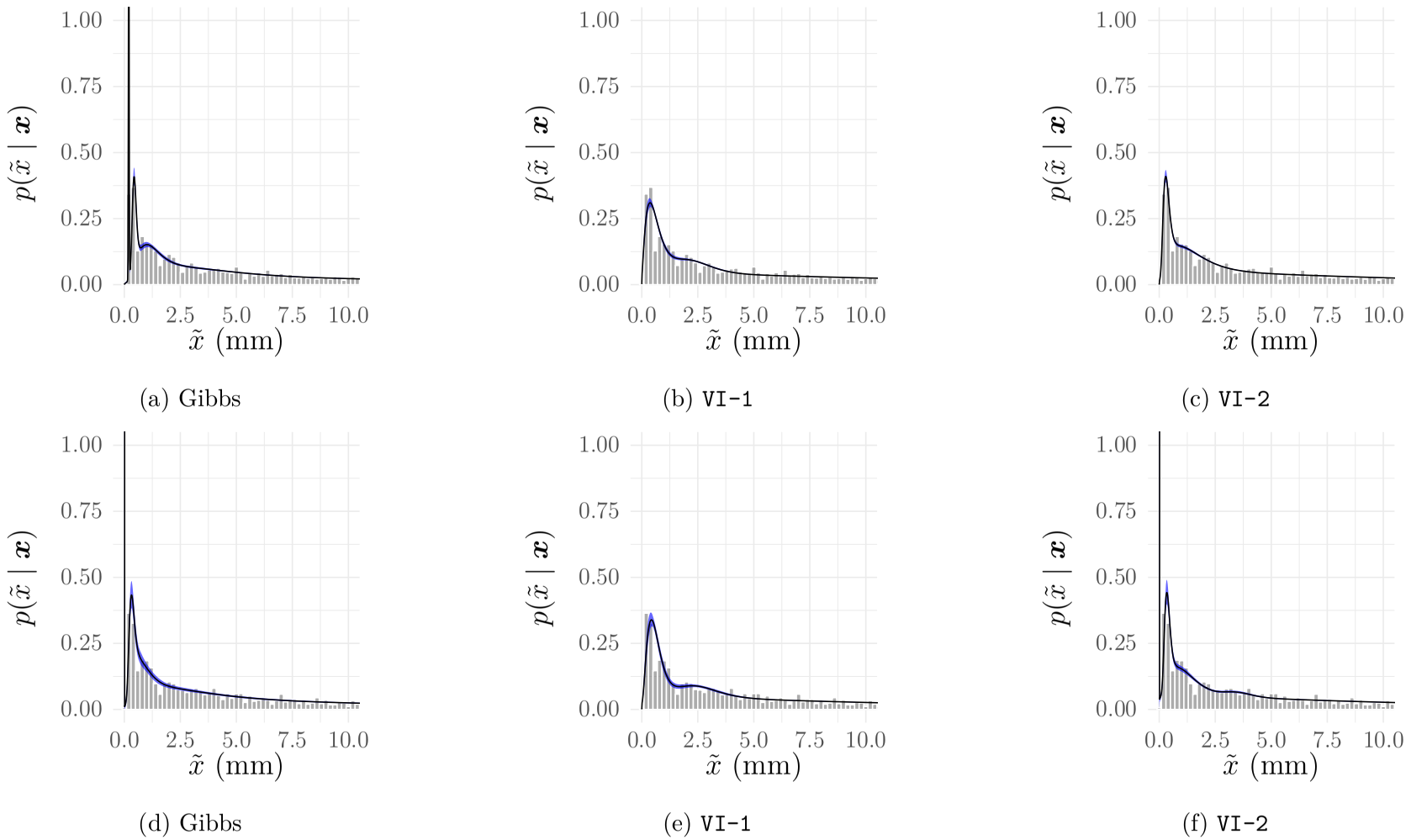
Figure 4.15: Similar to fig. 4.14 but for $K_{\text{model}} = K_{\text{data}} = 20$.




(a) Gibbs (b) `VI-1` (c) `VI-2`

(d) Gibbs (e) `VI-1` (f) `VI-2`

Figure 4.16: Posterior predictive densities and intervals for daily rainfall at the Woronora (top) and Wellington (bottom) dams.




(a) Gibbs (b) `VI-1` (c) `VI-2`

(d) Gibbs (e) `VI-1` (f) `VI-2`

Figure 4.17: Posterior predictive densities and intervals for daily rainfall at the Serpentine Main (top) and North Dandalup (bottom) dams.

# Chapter 5

# Conclusion and Extensions

We have constructed two algorithms for performing variational inference on mixtures of gamma distributions. Our results show that the orthogonal mean-shape parameterisation of the gamma distribution used in `VI-2` yields a mean-field approximation that is superior to shape-rate parameterisation in `VI-1` and consistent with a converged Gibbs sampler. We also found that, for more than two components, `VI-2` generally returns a more plausible posterior predictive distribution than a Gibbs sampler parameterised by the shape and rate, and it does so at a fraction of the computational cost. Due to an implementation bug, we were not able to test the Gibbs sampler for mixtures with more than fourteen components. Even so, the results we have indicate that the advantages of `VI-2` over the Gibbs sampler grow in both speed and accuracy with the number of components in the mixture.

There are many exciting directions that future research may take. One problem inherit to variational inference with the KL divergence is that it systematically underestimates the posterior variance. Promising research by Wang, Liu, and Liu (2018) and others finds that other $\alpha$-divergences (of which KL is a special case) may overcome this 'mass-covering' deficiency of the KL divergence and hence deliver a better posterior approximation. (Note that $\alpha$-divergence has no relation to the shape parameter $\alpha_k$.)

Visually, it is clear from in figs. 4.6 and 4.11 that `VI-2` produces a superior posterior approximation to `VI-1`. However, we did not evaluate this the quality of this approximation numerically. We calculated IAE between the posterior *predictive* distributions, but these 'throw away' information about the posterior distributions. Using a numerical measure of divergence between posterior densities would allow for a more systematic comparison of the performance of MCMC, `VI-1`, and `VI-2`.

We showed that a reparameterisation which transforms the parameters of the gamma

distribution into a space for which the posterior distribution is weakly dependent vastly the improves posterior inference of VI. Since the Fisher information matrix is diagonal, we call this an orthogonal reparameterisation. This raises an interesting question: could we *automatically* derive orthogonal parameterisations or something to the same effect? More generally, is it possible to transform the parameters in such a way that their posterior distribution becomes weakly dependent? Recall from section 3.1.3 that the posterior parameterised by the shape and rate approaches a multivariate normal distribution as $N \to \infty$. Indeed, this is supported by the plots in fig. 4.6. This implies that there exists a *linear* transformation from the shape-rate parameter space into a space which is weakly dependent or independent. If we could automate the process of finding this transformation, that would vastly reduce the amount of effort required by researchers and practitioners to search for parameterisations that are compatible with the mean-field approximation.

Gibbs sampling is an effective and widely-practised technique for Bayesian inference on mixture models; however, there are more modern approaches to MCMC that promise faster performance with reduced need for model-specific reconfiguration. One such method that has seen significant interest recently is Hamiltonian Monte Carlo (HMC). HMC uses Hamiltonian physics to exploit the geometry of the posterior distribution. This enables it to generate candidates which have a near-100% acceptance ratio in a Metropolis step, facilitating efficient exploration of the parameter space. Furthermore, since the Hamiltonian equations are derivative-based, probabilistic programming libraries such as `Stan` are able to leverage automatic differentiation to automatically derive the Hamiltonian dynamics, requiring the practitioner only to specify the model (Carpenter et al. 2017). Almond (2014) find that, for hierarchical mixture models, HMC performs 60% faster than a popular implementation of the random walk Metropolis algorithm (which is closely related to Gibbs sampling) from the statistical package `JAGS`. For a more comprehensive comparison, future works might compare the performance of VI not only to a Gibbs sampler, but HMC and other MCMC methods as well.

Another direction for future research is to reframe in a variational perspective the work of Bertolacci et al. (2019) on MCMC methods for modelling Australian rainfall. In particular, they augment a mixture of gamma distributions with time- and space-dependent mixture weights. They also add a Dirac delta component $\delta(x)$ to the mixture to represent days of zero rainfall; in contrast, we simply ignored zero-rainfall days and time-dependency. Bertolacci et al. also provide a careful treatment of 'missingness': the absence of any recorded observation. Although comprehensive, their model takes a long time to compute, and we postulate that variational methods would reduce this time drastically.

There are many improvements that we could make to the computational efficiency of our VI algorithms. One promising avenue for improvement is parallelisation, which

both `VI-2` and `VI-1` are naturally suited to. Although we fully parallelised most of the computational *operations* in our implementation, we have not yet explored the possibility of running multiple *instances* of VI in parallel. This would enable us to run VI on multiple datasets simultaneously in a computationally efficient manner by harnessing the ability of modern numerical computation libraries to perform large tensor operations much more efficiently than many smaller operations. Parallelisation would also lend itself to GPU devices, taking inspiration from the field of deep learning where GPUs enable the training of very wide and deep neural networks when it would be prohibitive to do so on a CPU.

Lastly, future work could address ways to avoid the component degeneracy observed in fig. 4.15. One approach may be to introduce a joint prior on $\boldsymbol{\mu}$ that 'forces' the component means apart. A highly informative Dirichlet prior on the distances between successive component means $\mu_k - \mu_{k-1}$ would might be suitable (where $\mu_k$ is the nearest neighbour of $\mu_{k-1}$). One would need to take an expectation on the logarithm of its density function in the $z_{i,j}$ step, but this could be approximated using a Taylor series.

In summary, this thesis provides a starting point for work on fast, accurate variational inference on mixtures of gamma distributions. Our Python package allows practitioners to perform positive-valued inference with minimal requirement for background technical knowledge. In a single line of code, our function returns the parameters to a closed-form posterior approximation that is remarkably close to that produced by a Gibbs sampler and does so in a fraction of the time.

# Appendix A

# Appendix

## A.1 Update Equation for Mean-Field Coordinate-Ascent Variational Inference

Beginning with the definition of the ELBO in eq. (2.10), the full derivation for the mean-field coordinate ascent variational inference update equation proceeds as follows.

$$
\begin{aligned}
\mathcal{L}(q) &= \mathbb{E}_q\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right] - \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right] \\
&= \mathbb{E}_q\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right] - \mathbb{E}_q\left[\log \prod_{j=1}^{M} q_j(\boldsymbol{\theta}_j)\right] \\
&= \mathbb{E}_{q_i}[\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right]] + \sum_{j=1}^{M} \mathbb{E}_{q_j}\left[\log q_j(\boldsymbol{\theta}_j)\right] \\
&= \mathbb{E}_{q_i}[\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right]] - \mathbb{E}_{q_i}\left[\log q_i(\boldsymbol{\theta}_i)\right] - \sum_{j:j\neq i}^{M} \mathbb{E}_{q_j}\left[\log q_j(\boldsymbol{\theta}_j)\right] \\
&= \mathbb{E}_{q_i}[\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right] - \log q_i(\boldsymbol{\theta}_i)] - \sum_{j:j\neq i}^{M} \mathbb{E}_{q_j}\left[\log q_j(\boldsymbol{\theta}_j)\right] \\
&= \mathbb{E}_{q_i}\left[\log \frac{\exp\left(\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right]\right)}{q_i(\boldsymbol{\theta}_i)}\right] - \sum_{j:j\neq i}^{M} \mathbb{E}_{q_j}\left[\log q_j(\boldsymbol{\theta}_j)\right] \\
&= -\mathrm{D}_{\mathrm{KL}}\left(q_i(\boldsymbol{\theta}_i) \| c^{-1} \exp\left(\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right]\right)\right) - \log c - \sum_{j:j\neq i}^{M} \mathbb{E}_{q_j}\left[\log q_j(\boldsymbol{\theta}_j)\right]
\end{aligned}
$$

where $c = \int_{\boldsymbol{\Theta}_i} \exp\left(\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right]\right) d\boldsymbol{\theta}_i$ is a normalisation constant. Clearly, the maximum of eq. (2.13) occurs where $q_i(\boldsymbol{\theta}_i) = c^{-1} \exp\left(\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right]\right)$. Therefore,

$$\begin{aligned} q_i^*(\boldsymbol{\theta}_i) &\propto \exp\left(\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right]\right) \\ &\propto \exp\left(\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta} \mid \boldsymbol{x}) + \log p(\boldsymbol{x})\right]\right) \\ &\propto \exp\left(\mathbb{E}_{q_j:j\neq i}\left[\log p(\boldsymbol{\theta} \mid \boldsymbol{x})\right]\right). \end{aligned}$$

## A.2 The Newton-Raphson Algorithm

The Newton-Raphson algorithm is a root-finding algorithm. It uses the first derivative of the objective function to find the root of its first-order Taylor expansion. First, consider the univariate case. Suppose that we have some function $f(\theta)$ with a non-zero first derivative and we want to find the value $\theta^*$ such that $f'(\theta^*) = 0$. The Newton-Raphson algorithm iteratively updates the best guess for $\theta^*$ at step $t+1$ using the equation

$$\theta_{t+1} = \theta_t - \frac{f'(\theta_t)}{f''(\theta_t)}$$

The extension to the multivariate case is similar. Suppose we want to find an extremum of $f(\boldsymbol{\theta})$ with respect to some vector $\boldsymbol{\theta} = (\theta_1, ..., \theta_p)$. Define the gradient vector and Hessian matrix respectively as

$$f'(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial f}{\partial \theta_1} \\ \vdots \\ \frac{\partial f}{\partial \theta_p} \end{bmatrix} \qquad f''(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial^2 f}{\partial \theta_1^2} & \cdots & \frac{\partial^2 f}{\partial \theta_1 \partial \theta_p} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial \theta_p \partial \theta_1} & \cdots & \frac{\partial^2 f}{\partial \theta_p^2} \end{bmatrix}.$$

The multivariate update equation is

$$\boldsymbol{\theta}^{[t+1]} = \boldsymbol{\theta}^{[t]} - f''\left(\boldsymbol{\theta}^{[t]}\right)^{-1} f'\left(\boldsymbol{\theta}^{[t]}\right).$$

## A.3 Equivalent Priors

In order to ensure a fair comparison, we must set equivalent priors on all parameters. Both parameterisations share the same priors on $\alpha_k$ and $\boldsymbol{\pi}$:

$$p(\alpha_k) \propto \frac{\mathrm{e}^{r_k \alpha_k}}{\Gamma(\alpha_k)^{s_k}}$$

$$p(\boldsymbol{\pi}) = \text{Dirichlet}(\boldsymbol{\omega})$$

for $r_k, s_k > 0$, $\boldsymbol{\omega} = (\omega_1, ..., \omega_K)$, and $\omega_k > 0$. The prior on $\beta_k$ is

$$p(\beta_k) = \text{Gamma}(c_k, d_k).$$

We can try to derive an expression for the prior on $\mu_k = \frac{\alpha_k}{\beta_k}$ using the Mellin transform:

$$\begin{aligned} p(\mu_k) &= \int_0^\infty p_{\beta_k}(y) p_{\alpha_k}(\mu_k y) y dy \\ &\propto \int_0^\infty y^{c_k} \mathrm{e}^{-d_k y} \frac{\mathrm{e}^{r_k \mu_k y}}{\Gamma(\mu_k y)^{s_k}} dy \\ &\propto \frac{1}{\mu_k^{c_k+1}} \int_0^\infty \frac{u^{c_k} \mathrm{e}^{-\frac{u}{\mu}(d_k - r_k \mu)}}{\Gamma(u)^{s_k}} du \end{aligned}$$

where we have used the substitution $u = \mu_k y \Rightarrow dy = \frac{du}{\mu_k}$. However, this integral is intractable. Instead, we set an Inverse-Gamma prior on $\mu_k$:

$$p(\mu_k) = \text{Inv-Gamma}(\xi_k, \tau_k)$$

where $c_k, d_k, \xi_k, \tau_k > 0$. We have $\mu_k = \frac{\alpha_k}{\beta_k}$ and seek $\xi_k, \tau_k$ such that the prior on $\mu_k$ is as close as possible to the prior on $\frac{\alpha_k}{\beta_k}$, which is

$$\begin{aligned} p\left(\frac{\alpha_k}{\beta_k}\right) &= p(\alpha_k) p\left(\frac{1}{\beta_k}\right) \\ &\propto \frac{\mathrm{e}^{r_k \alpha_k}}{\Gamma(\alpha_k)^{s_k}} \text{Inv-Gamma}(\beta_k \mid c_k, d_k) \end{aligned}$$

For notational purposes, define $q(\mu_k) = p(\mu_k) = \text{Inv-Gamma}(\mu_k \mid \xi_k, \tau_k)$. We seek values $\xi_k^*, \tau_k^*$ that minimise the KL divergence:

$$\begin{aligned} \xi_k^*, \tau_k^* &= \underset{\xi_k, \tau_k}{\text{argmin}} \, \mathrm{D}_{\mathrm{KL}} \left( p\left(\frac{\alpha_k}{\beta_k}\right) \middle\| q\left(\mu_k\right) \right) \\ &= \underset{\xi_k, \tau_k}{\text{argmin}} \left( \mathbb{E}_p \left[ \log p\left(\frac{\alpha_k}{\beta_k}\right) \right] - \mathbb{E}_p[\log q(\mu_k)] \right) \end{aligned}$$

We can find these values using the Newton-Raphson algorithm. Let $f(\xi_k, \tau_k) = \mathrm{D}_{\mathrm{KL}} \left( p\left(\frac{\alpha_k}{\beta_k}\right) \middle\| q\left(\mu_k\right) \right)$. Define the gradient vector and Hessian matrix, respectively, as

$$f'(\xi_k, \tau_k) = \begin{bmatrix} \frac{\partial f}{\partial \xi_k} \\ \frac{\partial f}{\partial \tau_k} \end{bmatrix} \qquad f''(\xi_k, \tau_k) = \begin{bmatrix} \frac{\partial^2 f}{\partial \xi_k^2} & \frac{\partial^2 f}{\partial \tau_k \partial \xi_k} \\ \frac{\partial^2 f}{\partial \xi_k \partial \tau_k} & \frac{\partial^2 f}{\partial \tau_k^2} \end{bmatrix}.$$

Let $\boldsymbol{\theta} = (\xi_k, \tau_k)$. The Newton-Raphson update equation is

$$\boldsymbol{\theta}^{[t+1]} = \boldsymbol{\theta}^{[t]} - f''\left(\boldsymbol{\theta}^{[t]}\right)^{-1} f'\left(\boldsymbol{\theta}^{[t]}\right).$$

## A.4 Adaptive Rejection Sampling

### A.4.1 Non-adaptive rejection sampling

Suppose that we wish to sample a random variable $X$ whose density is known to be proportional to $g(x)$. Then (non-adaptive) rejection sampling draws proposals from a

density $g_v(x)$ which, when multiplied by some constant $M$, envelops $g(x)$. We can find $M$ by solving

$$M = \max_x \frac{g(x)}{g_v(x)}.$$

The accept-reject algorithm then proceeds as follows.

1. Sample $x$ from $g_u(x)$.
2. Sample $u$ from Uniform$(0, 1)$.
3. If $u < \frac{g(x)}{Mg_v(x)}$ then *accept* $x$, otherwise *reject* it.
4. Repeat steps 1 to 3 for as many samples as desired.

### A.4.2 Adaptive rejection sampling

Adaptive rejection sampling (ARS) enables us to sample from log-concave densities that are known only to a normalising constant (Gilks and Wild 1992). Unlike plain rejection sampling, however, ARS does not require a proposal density. Instead, it constructs an envelope of $g(x)$ that is updated to more closely resemble the target density each time a sample is rejected. There are two main formulations of ARS: one which requires derivatives and one which does not. We use the former. The following is a summary of the work of Gilks and Wild (1992).

First, define $h(x) = \log g(x)$ and assume that $h(x)$ is concave (so $h''(x) < 0$ for all $x$). ARS constructs the envelope using tangents to $g(x_m)$ at points $x_1 \leq x_2 \leq ... \leq x_M$. The tangents intersect at

$$z_m = \frac{h(x_{m+1} - h(x_m) - x_{m+1}h'(x_{m+1}) + x_m h'(x_m)}{h'(x_m) - h'(x_{m+1})}$$

and, therefore, the piecewise-linear tangent function between points $x_m$ and $x_{m+1}$ is

$$u_M(x) = h(x_m) + (x - x_m)h'(x_m)$$

for $x \in [z_m, z_{m+1})$ and $m = 0, ..., M + 1$ where we define $z_0$ as $-\infty$ and $z_{M+1}$ as $\infty$. $u_M(x)$ forms an *upper hull* of $h(x)$. Define the proposal density as

$$g_v(x) = \frac{\exp u(x)}{\int_{\mathcal{X}} \exp u(x')dx'}$$

where $\mathcal{X}$ is the probability space of $X$. Lastly, for define the lower hull as the piecewise linear function

$$l_M(x) = \frac{(x_{m+1} - x)h(x_m) + (x - x_m)h(x_{m+1})}{x_{m+1} - x_m}$$

for $x \in [x_m, x_{m+1})$ and $m = 0, ..., M + 1$ where, similarly, we define $x_0$ as $-\infty$ and $x_{M+1}$ as $\infty$. ARS proceeds with the following steps.

**Initialisation step**

Initialise $x_1, ..., x_M$. If $\mathcal{X}$ is unbounded on the left then choose $x_1$ such that $h'(x_1) > 0$, and if $\mathcal{X}$ is unbounded on the right then choose $x_1$ such that $h'(x_1) < 0$.

**Sampling step**

Sample a proposal $x^*$ from $g_v(x)$ and sample $u$ from Uniform$(0, 1)$. If $u \leq \exp\left(l_M(x^*) - u_M(x^*)\right)$ then accept $x^*$. This is called the squeeze test. Otherwise, run the updating step below and perform the following rejection test: if $u \leq \exp\left(h(x^*) - u_M(x^*)\right)$ then accept $x^*$. Otherwise, reject $x^*$.

**Updating step**

If the squeeze test fails then set $x_{M+1} \leftarrow x^*$ followed by $M \leftarrow M + 1$. Then, reconstruct the functions $u_M, l_M$, and $g_v$ and repeat the sampling step until the desired number of samples are accepted.

# Bibliography


Almond, Russell G. 2014. "A comparison of two MCMC algorithms for hierarchical mixture models." *CEUR Workshop Proceedings* 1218:1–19. ISSN: 16130073.

Altosaar, Jaan, and David M Blei. 2018. "Proximity Variational Inference." *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS) 2018* 84.

Arellano, Manuel. 2016. *Bayesian analysis.*

Ayanlade, Ayansina, Maren Radeny, John F. Morton, and Tabitha Muchaba. 2018. "Rainfall variability and drought characteristics in two agro-climatic zones: An assessment of climate change challenges in Africa." *Science of the Total Environment* 630:728–737. ISSN: 18791026. doi:10.1016/j.scitotenv.2018.02.196. https://doi.org/10.1016/j.scitotenv.2018.02.196.

Bamler, Robert, Cheng Zhang, Manfred Opper, and Stephan Mandt. 2017. "Perturbative black box variational inference." *Advances in Neural Information Processing Systems* 2017-Decem (Nips): 5080–5089. ISSN: 10495258. arXiv: arXiv:1709.07433v2.

Baydin, Atılım Günes¸, Güne¸s Baydin, Barak A Pearlmutter, and Jeffrey Mark Siskind. 2018. "Automatic Differentiation in Machine Learning: a Survey." *Journal of Machine Learning Research* 18:1–43. http://www.jmlr.org/papers/volume18/17-468/17-468.pdf.

Belikov, Aleksey V. 2017. "The number of key carcinogenic events can be predicted from cancer incidence." *Scientific Reports* 7 (1): 1–8. ISSN: 20452322. doi:10.1038/s41598-017-12448-7. http://dx.doi.org/10.1038/s41598-017-12448-7.

Bertolacci, Michael, Edward Cripps, Ori Rosen, John Lau, and Sally Cripps. 2019. "Climate Inference on Daily Rainfall Across the Australian Continent, 1876 – 2015." 13 (2): 683–712.

Bishop, Christopher M. 2006. "Machine Learning and Pattern Recognition." In *Information Science and Statistics.* ISBN: 9780387310732.

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe. 2017. "Variational Inference: A Review for Statisticians." *Journal of the American Statistical Association* 112 (518): 859–877. ISSN: 1537274X. doi:10.1080/01621459.2017.1285773. arXiv: arXiv:1601.00670v9.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. "Latent Dirichlet allocation." *Journal of Machine Learning Research* 3 (4-5): 993–1022. ISSN: 15324435.

Blundell, Charles, and Yee Whye Teh. 2013. "Bayesian hierarchical community discovery." In *Advances in Neural Information Processing Systems.*

Bottou, Léon, and Olivier Bousquet. 2009. "The tradeoffs of large scale learning." *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference.*

Carpenter, Bob, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus A. Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. 2017. "Stan: A probabilistic programming language." *Journal of Statistical Software.* ISSN: 15487660. doi:10.18637/jss.v076.i01.

Chaney, Allison J B. 2015. "A Guide to Black Box Variational Inference for Gamma Distributions." *Technical Report:* 1–8. http://ajbc.io/resources/bbvi%7B%5C_%7Dfor%7B%5C_%7Dgammas.pdf.

Clarke, L. E., and Patrick Billingsley. 1980. *Probability and Measure.* doi:10.2307/3616746.

Cotter, Andrew. 2013. "Stochastic Optimization for Machine Learning." arXiv: 1308.3509. http://arxiv.org/abs/1308.3509.

Damsleth, Elvind. 1975. "Conjugate Classes for Gamma Distributions." *Scandinavian Journal of Statistics* 2 (2): 80–84.

Dean, Jeffrey. 2016. "Tensorflow: Large-Scale Deep Learning For Building Intelligent Computer Systems." doi:10.1145/2835776.2835844.

DeVore, Ronald A., and George G. Lorentz. 1993. *Constructive Approximation.* Springer-Verlag. ISBN: 0-387-50627-6.

Elouissi, Abdelkader, Mohammed Habi, Boumedienne Benaricha, and Sid Ahmed Boualem. 2017. "Climate change impact on rainfall spatio-temporal variability (Macta watershed case, Algeria)." *Arabian Journal of Geosciences* 10 (22). ISSN: 18667538. doi:10.1007/s12517-017-3264-x.

Euler, Leonhard. 1744. "Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes sive solutio problematis isoperimetrici latissimo sensu accepti." ISSN: 0048-7333. doi:983498498./c33333.

Frey, Brendan J, and Geoffrey E. Hinton. 1999. "Variational learning in nonlinear gaussian belief networks." *Neural Computation* 11 (1): 193–213. http://www.scopus.com/inward/record.url?eid=2-s2.0-0032603958%7B%5C&%7DpartnerID=40%7B%5C&%7Dmd5=06a497bad64eb41cfc40e76f86175e48.

Futami, Futoshi, Issei Sato, and Masashi Sugiyama. 2017. "Variational Inference based on Robust Divergences." 84. arXiv: 1710.06595. http://arxiv.org/abs/1710.06595.

Gelfand, Alan E., and Adrian F.M. Smith. 1990. "Sampling-based approaches to calculating marginal densities." *Journal of the American Statistical Association* 85 (410): 398–409. ISSN: 1537274X. doi:10.1080/01621459.1990.10476213.

Geman, Stuart, and Donald Geman. 1984. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6 (6): 721–741. ISSN: 01628828. doi:10.1109/TPAMI.1984.4767596.

Gilks, W. R., and P. Wild. 1992. "Adaptive Rejection Sampling for Gibbs Sampling." *Applied Statistics* 41 (2): 337. ISSN: 00359254. doi:10.2307/2347565.

Hastings, W, K. 1970. "Monte Carlo Sampling Methods Using Markov Chains and Their Applications." *Biometrika* 57 (1): 97–109.

Hoffman, Matthew D., and David M. Blei. 2015. "Structured stochastic variational inference." *Journal of Machine Learning Research* 38:361–369. ISSN: 15337928.

Hoffman, Matthew D., David M. Blei, Chong Wang, and John Paisley. 2013. "Stochastic variational inference." *Journal of Machine Learning Research* 14:1303–1347. ISSN: 15324435.

Howden, Matthew, Rohan Nelson, and Kirk Zammit. 2018. "Australian agriculture overview." *Agricultural Commodities* 8 (2): 11–19. ISSN: 18395627.

Jaynes, E. T. 1957. *Information theory and statistical mechanics. II.* doi:10.1103/PhysRev.108.171.

Jones, P. W. 1989. *Multi-armed bandit allocation indices,* 40:1158–1159. 12. ISBN: 9780470670026. doi:10.1057/jors.1989.200.

Jordan, Michael I., Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. 1999. "Introduction to variational methods for graphical models." *Machine Learning* 37 (2): 183–233. ISSN: 08856125. doi:10.1023/A:1007665907178.

Kemp, Charles, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. 2006. "Learning systems of concepts with an infinite relational model." In *Proceedings of the National Conference on Artificial Intelligence*. ISBN: 1577352815.

Knowles, David A. 2015. "Stochastic gradient variational Bayes for gamma approximating distributions": 1–14. arXiv: 1509.01631. http://arxiv.org/abs/1509.01631.

Kullback, S., and R. A. Leibler. 1951. "On Information and Sufficiency." *The Annals of Mathematical Statistics*. ISSN: 0003-4851. doi:10.1214/aoms/1177729694.

Landau, L. 1936. "The theory of phase transitions." *Nature* 138 (3498): 840–841. ISSN: 00280836. doi:10.1038/138840a0.

Leisink, M. A.R., and H. J. Kappen. 2001. "A tighter bound for graphical models." *Neural Computation* 13 (9): 2149–2171. ISSN: 08997667. doi:10.1162/089976601750399344.

Li, Meng, Kun Xie, Hui Kuang, Jun Liu, Deheng Wang, Grace E Fox, Wei Wei, et al. 2018. "Spike-timing pattern operates as gamma-distribution across cell types, regions and animal species and is essential for naturally-occurring cognitive states." *bioRxiv*. http://biorxiv.org/content/early/2018/02/09/145813.abstract.

Llera, A., D. Vidaurre, R. H. R. Pruim, and C. F. Beckmann. 2016. "Variational Mixture Models with Gamma or inverse-Gamma components." arXiv: 1607.07573. http://arxiv.org/abs/1607.07573.

Madras, Neal, and Deniz Sezer. 2010. "Quantitative bounds for Markov chain convergence: Wasserstein and total variation distances." *Bernoulli* 16 (3): 882–902. ISSN: 13507265. doi:10.3150/09-BEJ238.

Marron, J. S., and M. P. Wand. 1992. "Exact Mean Integrated Squared Error." *The Annals of Statistics*. ISSN: 0090-5364. doi:10.1214/aos/1176348653.

Metropolis, Nicholas, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. "Equation of state calculations by fast computing machines." *The Journal of Chemical Physics* 21 (6): 1087–1092. ISSN: 00219606. doi:10.1063/1.1699114.

Mohammadi, A., M. R. Salehi-Rad, and E. C. Wit. 2013. "Using mixture of Gamma distributions for Bayesian analysis in an M/G/1 queue with optional second service." *Computational Statistics* 28 (2): 683–700. ISSN: 09434062. doi:10.1007/s00180-012-0323-3.

Nicholls, Neville, Wasyl Drosdowsky, and Beth Lavery. 1997. "Australian rainfall variability and change." *Weather* 52 (3): 66–72. ISSN: 14778696. doi:10.1002/j.1477-8696.1997.tb06274.x.

Pendergrass, Angeline G., Reto Knutti, Flavio Lehner, Clara Deser, and Benjamin M. Sanderson. 2017. "Precipitation variability increases in a warmer climate." *Scientific Reports* 7 (1): 1–9. ISSN: 20452322. doi:10.1038/s41598-017-17966-y. http://dx.doi.org/10.1038/s41598-017-17966-y.

Peterson, C., and James R. Anderson. 1987. "A mean field theory learning algorithm for neural networks." *Complex Syst.* 1:995–1019.

R Development Core Team, R. 2011. *R: A Language and Environment for Statistical Computing.* ISBN: 3900051070. doi:10.1007/978-3-540-74686-7.

Ranganath, Rajesh, Sean Gerrish, and David M. Blei. 2014. "Black box variational inference." *Journal of Machine Learning Research* 33:814–822. ISSN: 15337928.

Robert, Christian, and George Casella. 2011. "A short history of Markov Chain Monte Carlo: Subjective recollections from incomplete data." *Statistical Science* 26 (1): 102–115. ISSN: 08834237. doi:10.1214/10-STS351. arXiv: arXiv:0808.2902v7.

Saul, Lawrence K., Tommi Jaakkola, and Michael I. Jordan. 1996. "Mean field theory for sigmoid belief networks." *Journal of Artificial Intelligence Research* 4:61–76. ISSN: 10769757.

Sklar, A. 1959. "Fonctions de Répartition à n Dimensions et Leurs Marges." *Publications de L'Institut de Statistique de L'Université de Paris.*

Tran, Dustin, David M. Blei, and Edoardo M. Airoldi. 2015. "Copula variational inference." *Advances in Neural Information Processing Systems* 2015-Janua:3564–3572. ISSN: 10495258.

Tran, Viet Hung. 2018. "Copula Variational Bayes inference via information geometry." 2018:1–23. arXiv: 1803.10998. http://arxiv.org/abs/1803.10998.

Tsvetkov, Dimiter, Lyubomir Hristov, and Ralitsa Angelova-Slavova. 2013. "On the convergence of the Metropolis-Hastings Markov chains": 1–14. arXiv: 1302.0654. http://arxiv.org/abs/1302.0654.

VanDerwerken, Douglas. 2017. "Not every Gibbs sampler is a special case of the Metropolis–Hastings algorithm." *Communications in Statistics - Theory and Methods.* ISSN: 1532415X. doi:10.1080/03610926.2016.1228961.

Wang, Dilin, Hao Liu, and Qiang Liu. 2018. "Variational inference with tail-adaptive f-divergence." *Advances in Neural Information Processing Systems* 2018-Decem (1): 5737–5747. ISSN: 10495258.

Williams, Christopher K.I., and Geoffrey E. Hinton. 1991. *Mean field networks that learn to discriminate temporally distorted strings.* doi:10.1016/b978-1-4832-1448-1.50008-1.