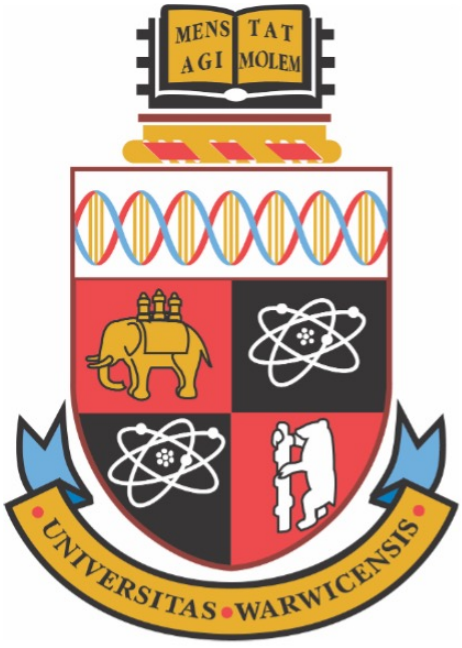
Federated Learning under Misspecification: A Partitioned Generalised Variational Inference Approach

A Third Year Discrete Mathematics Project

Terje J. Mildner

Supervised by Prof. Theodoros Damoulas
Co-supervised by Oliver Hamelijnck



Department of Computer Science
University of Warwick
Coventry, CV4 7AL, UK
April 2024

Contents

The sections marked by the ⋆ symbol contain core contributions of this thesis.

List of Figures

List of Tables

List of Algorithms

The different colours represent different implementations of the respective algorithm.

Acknowledgements

I would like to especially thank my supervisor Theo Damoulas, without whose help and knowledge this thesis would not have been possible. Much of the insights I gained throughout this project are due to his questions that challenged my knowledge and lead me to further delve into the topics discussed throughout this thesis. I am also grateful for introducing me to many fascinating topics including Federated Learning and Model Misspecification.

I would also like to thank Ollie Hamelijnck, who acted as a second supervisor throughout this project. I am particularly grateful for being able to ask all kind of questions, even if we had already discussed these before, and always getting helpful replies. His help was instrumental in developing the necessary knowledge to implement the algorithms and discover, and fix the bugs in my code.

This thesis would not have been possible without the many insightful discussions we had as a group.

I would also like to thank my family for supporting me throughout this project, and university in general. Thank you for always encouraging me to do my best.

Federated Learning under Misspecification: A Partitioned Generalised Variational Inference Approach

Terje J. Mildner
Department of Computer Science
University of Warwick
Coventry, CV4 7AL, UK

Supervised by: Theo Damoulas and Ollie Hamelijnck

Abstract

Federated Learning (FL) is the collaborative training of a single model through a federation of clients with individual, and decentralised data sets. This emerging framework enables accessing previously inaccessible, and sensitive data sets, whilst offering increased privacy guarantees. We investigate Partitioned Variational Inference (PVI), a Bayesian approach to FL; these are able to capture data and model uncertainty, but have received little attention in the literature. Due to the intractability of exact Bayesian posteriors, PVI uses a variational approximation to the Bayesian posterior by combining Variational Inference and Expectation Propagation, which are frequently used in traditional problem settings. In practice, however, Bayesian methods like PVI suffer from misspecification, where the data generating process is not known or available, nor is the prior distribution always suitable and well–specified. To mitigate misspecification in federated learning, we propose Partitioned Generalised Variational Inference (PGVI), a robust approach to FL, in which we leverage recent advances in robust approximate Bayesian inference. We demonstrate increased robustness, better uncertainty quantification, and improved accuracy through toy experiments with Gaussian Mixture Models, and classification of real data through Bayesian Neural Networks.

Keywords: Federated Learning, Variational Inference, Model Misspecification, Robustness, Generalised Variational Inference, Gaussian Mixture Models, Bayesian Neural Networks

Chapter 1

Introduction

> *"Federated Learning is a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider. Each client's raw data is stored locally and not exchanged or transferred; instead focused updates intended for immediate aggregation are used to achieve the learning objective."* — Kairouz et al. (2021)

In a world where consumers are increasingly concerned with keeping their proprietary data private and not sharing it with data collecting servers, but personal computing devices proliferate in daily life for both commercial and private purposes, there is a mismatch between data that exists, and data that is available to be used. Federated learning (FL) attempts to bridge this gap through keeping data local and training a model in a decentralised manner, where the server never sees any client data, McMahan et al. (2017). Consider, for instance, a federation of hospitals that have patient data that is confidential, see figure 1.1 below.

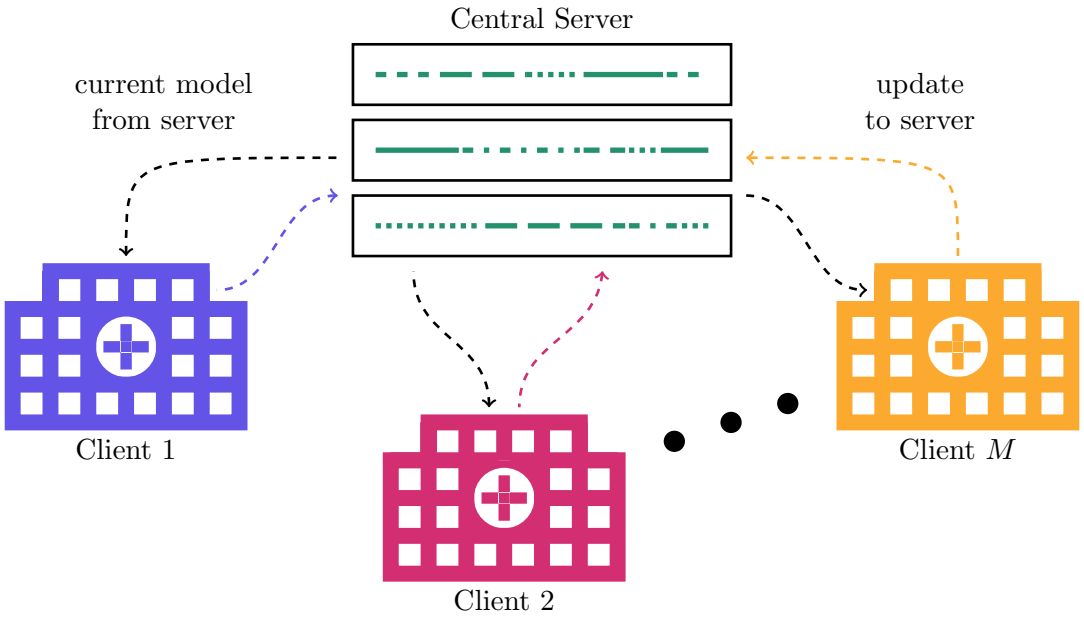


Figure 1.1: Graphical visualisation of **Federated Learning**.

If these hospitals wanted to train a model on their combined data, then they could do this via federated learning without sharing confidential patient information with a central

server; in theory, FL achieves this by having individual clients retrieve the current model from the server, optimising this locally based on the individual data, and sending an update, that does not contain data, for aggregation back to the server, Kairouz et al. (2021).

We could also consider a model in which we wanted to train a handwriting recognition model, that is able to read handwriting and transfer this to electronic text. For example, we can imagine modern cars, that have touchscreens on which address lines are written by hand, as a set of clients. This is privacy sensitive information that could be used by malicious actors to identify common behavioural patterns, however the data is of high quality since users directly verify what they wanted to write by selecting the correct address. Training such a classification model allows the improvement of services by the company, as well as having further applications to handwriting recognition.

Federated learning, is distinctly different to both global learning, where the entire data set is known and we effectively have only a single client, as well as distributed learning. Distributed learning has received much attention in the literature, even before the introduction of FL in the seminal paper of McMahan et al. (2017); but is concerned with only a restricted case of federated learning, where we have a data centre setting, and the client splits originate by homogeneously (evenly) partitioning a large data set across individual machines (Mesquita et al., 2020; Ahn et al., 2014; Kairouz et al., 2021). In federated learning, we place no such restriction on the data partitioning nor the client heterogeneity, and hence have unique challenges that FL needs to address (Kairouz et al., 2021).

The most immediate of these challenges is having an algorithmic framework that is able to learn from decentralised and distributed data[1] without loosing much accuracy in predictions. To this end McMahan et al. (2017) introduced the first such framework, called Federated Averaging which is based on Stochastic Gradient Descent (Murphy, 2022). This method, as well as most of the FL literature, is concerned with a frequentist approach to machine learning, where the aim is to investigate how parameters would change if the data changed, Murphy (2022). It does not treat this parameter as a random variable, but rather as a fixed quantity, where the data is a random variable and not fixed, Murphy (2022).

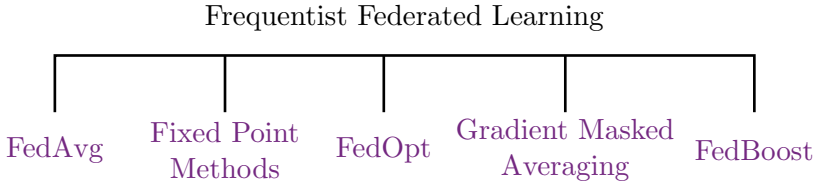


Figure 1.2: Tree showing a selection of Frequentist **Federated** Learning algorithms.

Such approaches to federated learning have been widely investigated for instance in McMahan et al. (2017), Malinovsky et al. (2020), Reddi et al. (2021), Tenison et al. (2023) and Hamer et al. (2020), as shown in figure 1.2 above (corresponding to each from left to right). These algorithms present interesting ideas that each contribute to the FL literature

1. We highlight the difference between data being distributed, i.e. split across some decentralised framework, and distributed learning, where we require this distributed framework to be restricted as described above.

by proposing different ways of performing the aggregation of client updates. Especially the Gradient Masked Averaging (GMA) approach of Tenison et al. (2023) is able to address data heterogeneity, a challenge that arises in federated learning due to clients having different data generating processes, for instance because different people have different handwriting.

Further challenges in federated learning can be seen in figure 1.3, as well as approaches that aim to overcome these, Kairouz et al. (2021). These include, but are not limited to: communication efficiency (McMahan et al., 2017; Hamer et al., 2020) and how informative updates are; data heterogeneity (Tenison et al., 2023), when the amount, quality or distribution of data varies across clients; client heterogeneity (Tziotis et al., 2023), when clients have different computational availability and power; and data privacy (Chen et al., 2022; Zhu et al., 2019; Zhao et al., 2023).

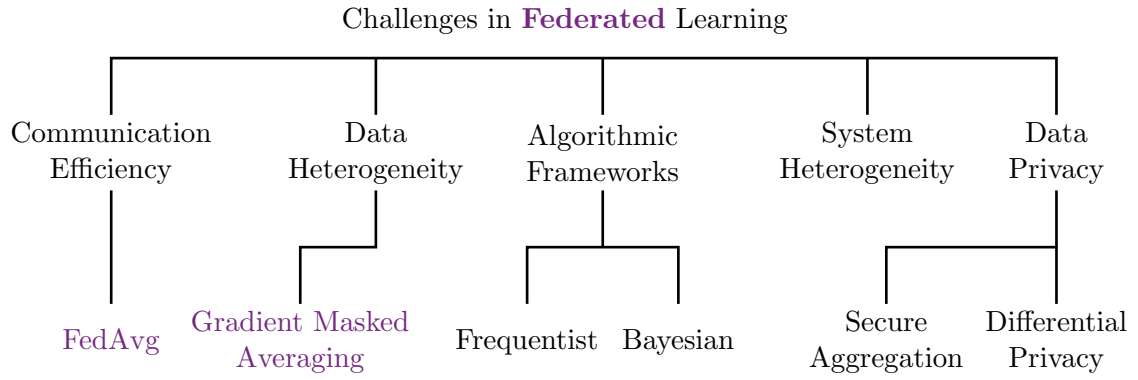


Figure 1.3: Tree showing a selection of challenges in **Federated** Learning. We further show a selection of methods that attempt to deal with these challenges.

However, all these frequentist approaches to federated learning, and their extensions, have one major drawback: they are unable to quantify uncertainty efficiently, especially concerning epistemic, model, uncertainty, Kassab and Simeone (2022) and Murphy (2023). Furthermore, frequentist statistics treats data as variable in general, which causes paradoxes in certain scenarios (Murphy, 2023), and is inconsistent with the notion that observed data is fixed.

Bayesian statistics treat parameters as unknown random variables that we attempt to infer from the underlying data through Bayes' theorem, Knoblauch et al. (2022). Bayes' theorem is commonly known for inferring the probability of an event A given that another event B has occurred, through the following equation (Bernardo and Smith, 2000):

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B)} \tag{1.1}$$

In Bayesian statistics we utilise this through placing a posterior distribution over the possible parameters given that we have observed some data, which corresponds to $\mathbb{P}(A|B)$. This can be calculated through the likelihood of the data occurring given these possible parameters $\mathbb{P}(B|A)$, as well as a prior distribution of these parameters $\mathbb{P}(A)$. The marginal likelihood term, $\mathbb{P}(B)$, normalises the result and makes sure that it is in fact a proper probability

distribution. However, exact computation of this term is generally not possible, even if we have all the data available, Rogers and Girolami (2016). Since exact inference is not possible, we require approximate inference that calculates the Bayesian posterior; we can do this via three main methods: Laplace approximation, deterministic approximate inference, and stochastic approximate inference, Rogers and Girolami (2016).

The Laplace approximation is motivated through the Bernstein–von Mises theorem, that the posterior converges, under regularity conditions and asymptotically for infinite data, to a multivariate Gaussian distribution, Kleijn and van der Vaart (2012); Rogers and Girolami (2016). However, in practice this turns out to not be a good approximation in most cases, Murphy (2023). Therefore making deterministic and stochastic approximate inference frameworks more appealing in practice. Stochastic approximate inference requires the computation of Markov Chains in a repetitive Monte Carlo fashion, through sampling from the true posterior, Rogers and Girolami (2016). This is typically known as Markov Chain Monte Carlo (Bishop, 2006) and includes approaches such as Gibbs sampling (Rogers and Girolami, 2016) and Stochastic Gradient Langevin Dynamics (Murphy, 2023; Ahn et al., 2014). Deterministic approximate inference uses a variational approximation to the Bayesian posterior, through the calculus of variations, Bishop (2006). This includes the frameworks of Variational Inference (VI) (Zhang et al., 2019; Blei et al., 2016) and Expectation Propagation (EP) (Minka, 2001b; Vehtari et al., 2020), where we aim to approximate the posterior from a set of simpler distributions (Pinski et al., 2015). Regardless of the approximate inference method chosen, this allows for principled uncertainty quantification through the placement of a probability distribution over the parameters.

This naturally extends to the distributed and the federated learning settings, and a selection of these approaches is shown in figure 1.4.

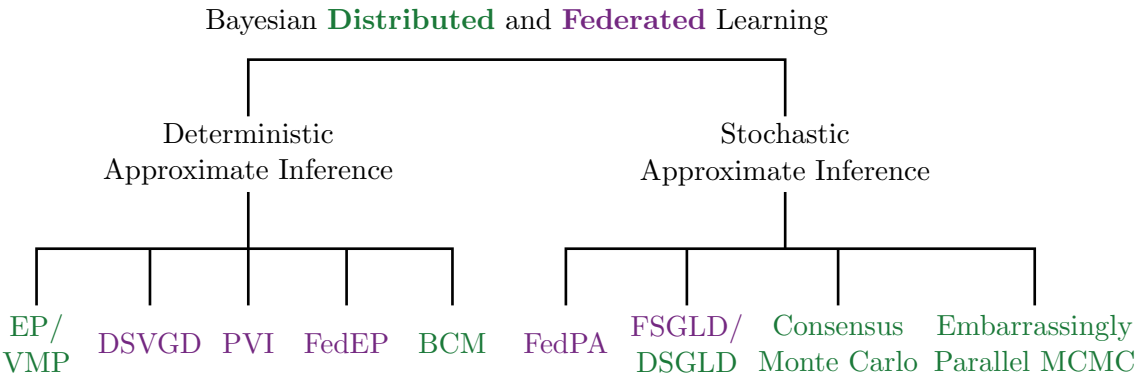


Figure 1.4: Best viewed in colour. Tree showing a selection of Bayesian **Distributed** and **Federated** Learning algorithms. We split these into stochastic and deterministisc approximate inference frameworks

The FL literature, has so far focused on the frequentist approaches, with a few works now also considering Bayesian approaches to federated learning, as seen in the figure above. For stochastic approximate inference, the main federated learning approaches are Federated Posterior Averaging (FedPA) (Al-Shedivat et al., 2021), and Federated Stochastic Gradient Langevin Dynamics (FSGLD) (Mekkaoui et al., 2021); and for deterministic approximate

inference there is Federated Expectation Propagation (FedEP) (Guo et al., 2023), Distributed Stein Variational Gradient Descent (DSVGD) (Kassab and Simeone, 2022), as well as Partitioned Variational Inference (PVI) (Bui et al., 2018; Ashman et al., 2022). However, as MCMC based methods are computationally and communication wise expensive due to having to sample extensively from the true distribution, and communicating Markov chains, this can violate federated learning assumptions. In this thesis we will be primarily concerned with Bayesian federated learning, due to the lack of research in this area (Kairouz et al., 2021), in which we consider the Partitioned Variational Inference (PVI) framework of Bui et al. (2018), which builds on VI and EP.

Similar to frequentist approaches having unique challenges, the same can be said for Bayesian approaches. In particular, these assume that the likelihood function (or the data generating process) are known and well specified, Jewson et al. (2018). In practice this is typically not the case and the likelihood function will not represent the data generating process (DGP) (Berk, 1966; Bernardo and Smith, 2000; Kleijn and van der Vaart, 2006; Shalizi, 2009; Walker, 2013; Miller, 2021) or is too complicated to work with (Shalizi, 2009). This is known as model misspecification, and the assumption that the model is incorrect is called the $\mathcal{M}$–open assumption, Bernardo and Smith (2000). However, it is known, and will be shown by us, that even under misspecification, we can still efficiently learn through Bayesian approaches (Miller, 2021; Knoblauch et al., 2022).

In this thesis, we will argue, along the lines of Knoblauch et al. (2022) and Jewson et al. (2018), that it is not sensible to mindlessly target the posterior as specified by Bayes' theorem (Bissiri et al., 2016; Miller, 2021), but rather one that is useful to a decision maker, depending on what she might want to target and use the model for.

This leads us to develop the novel federated learning algorithm, we term Partitioned Generalised Variational Inference, that we will show empirically to be robust to model misspecification, and leads to desirable uncertainty quantification.

1.1 Core Contributions

The core contribution of this thesis, which are marked by the $\star$ symbol in the table of contents, is the development of the novel federated learning algorithm Partitioned Generalised Variational Inference. We propose this method as a robust approach to federated learning under model misspecification, connecting robust Bayesian inference and federated learning. We test this method by implementing it on both toy data sets, where we can control the environment, and real–world data sets, where we do not know the true data distribution.

Additionally, we improve the theoretical understanding of the existing Partitioned Variational Inference framework of Bui et al. (2018) by elaborating on mathematical ideas used in parallel updating of PVI. We further implement PVI from scratch and verify the validity of this approach in comparison to global Variational Inference empirically on toy data sets. To demonstrate PGVI on real world data, we have first fixed the outdated and erroneous

code base of Ashman et al. (2022), and we then implement PGVI through modifying this PVI code base.

1.2 Structure

The thesis has the following structure. Chapter 2 explores the mathematical background of Bayesian machine learning, where we consider Bayes' theorem as an optimisation problem. Discussing how we can efficiently solve this optimisation problem through Variational Inference and Expectation Propagation builds the foundation for Partitioned Variational Inference. To demonstrate how PVI and PGVI compare with the existing literature of federated learning, we review a selection of frequentist and Bayesian approaches to FL, and how these address challenges in federated settings. Information geometry allows us to understand optimisation problems by considering their underlying geometry. This is particularly useful in probabilistic machine learning through divergence measures and by considering the statistical manifold of possible probability distributions. We use this to illustrate model misspecification in the $\mathcal{M}$–open framework and why the Bayesian posterior still accumulates around some optimal value.

Chapter 3 introduces Partitioned Variational Inference and it's relationship with Expectation Propagation, and Variational Inference. We discuss how different update schedules enable different properties of PVI, and we develop more mathematical justification for these. Furthermore, since the idea of combining EP and VI is not novel to PVI, but was already used in Hensman et al. (2014), we discuss the differences of these two approaches, and explore why PVI enables federated learning. We evaluate the effectiveness of PVI by implementing it from scratch for Gaussian Mixture Models (Attias, 1999) to conduct toy data experiments in settings where we can control all variables.

Chapter 4 introduces the framework of Generalised Variational Inference (Knoblauch et al., 2022), which is robust to model misspecification. We use this to develop a robust version of Partitioned Variational Inference, we term Partitioned Generalised Variational Inference. We explore how this can address robustness in federated learning, and how this links to frequentist approaches to FL. We further explore the effectiveness of PGVI in terms of model misspecification, robustness to outlier data points, and uncertainty quantification on toy data sets.

In chapter 5 we further explore PGVI through implementing it on real world data sets through Bayesian Neural Networks (BNN), and demonstrate that this novel method can outperform global VI, and therefore PVI, in classifying labelled images.

We conclude the thesis by discussing our contributions and open challenges related to federated learning, PVI, and Partitioned Generalised Variational Inference, in chapter 6. In chapter 7, we discuss project management details as well as legal, social, and ethical considerations. In the appendix, we present some key measure theoretic ideas and provide lengthy proofs for theorems that would disrupt the flow of this project. We further provide proofs for the derivations of equations we use for implementing PVI and PGVI.

Chapter 2

Background

To begin, we set the mathematical foundations for Bayesian machine learning and introduce the foundations of Partitioned Variational Inference. We explore the Federated Learning literature, and current challenges.

Chapter 2.1 describes necessary mathematical foundations of Bayesian machine learning, characterising probability measures as their unique Radon-Nikodym derivatives. We reframe Bayes' theorem as an information processing rule and hence introduce Variational Inference and Expectation Propagation, which provide the foundation for federated Bayesian learning.

Chapter 2.2 Federated Learning poses several challenges and constraints. We explore the Federated Learning literature and algorithmic frameworks through the lens of frequentist approaches and the approximate Bayesian inference frameworks, investigating stochastic and deterministic approximate inference.

Chapter 2.3 When investigating probability measures, we can explore the underlying geometry of the probability measure thorough information geometry. Specifically, we investigate divergence measures, and their properties.

Chapter 2.4 Having introduced different divergence measures, we can minimise these in VI instead.

Chapter 2.5 Model misspecification is a challenge in global learning algorithms, and specifically in federated learning where we use approximations for different client data.

2.1 Mathematical Background

Since we are concerned with probabilistic machine learning we need to define how these probability distributions, characterised by their unique Radon-Nikodym derivative measures[1], are defined (Cohn, 2013). For this we borrow the notation of Pinski et al. (2015) to define the measure spaces for which the results hold[2].

1. Measure theoretic knowledge is assumed for most of this thesis. Further definitions of standard measure theoretic statements and well–known results are therefore deferred to Appendix A.
2. Stronger assumptions need to be made for parts of the proofs of certain results. These will be discussed where necessary.

Definition 1 *The Measure Space* $(\Theta, \mathcal{E}, \mathcal{P}(\Theta))$*, is defined over the Polish Space* Θ *with it's Borel Sigma Algebra* $\mathcal{E}$*, and the set of all probability measures on* Θ*,* $\mathcal{P}(\Theta)$*.*

This definition is more general than will be considered in the experiments, where we only have finite dimensional $\Theta \subseteq \mathbb{R}^n$. However, it can easily be seen that we require Θ to be infinite dimensional for instance for Bayesian inverse problems, Pinski et al. (2015).

When talking about probability distributions in $\mathcal{P}(\Theta)$, we mean the probability distributions as characterised by their unique Radon–Nikodym derivatives of probability measures in $\mathcal{P}(\Theta)$ with respect to some common dominating measure. The Radon–Nikodym theorem guarantees uniqueness of such distributions up to sets of measure 0, Kullback and Leibler (1951). We formalise this in the following definition, adapted from Kullback (1959), with potentially generalised distributions, not required to take values in $[0, 1]$.

Definition 2 (Probability distribution) *For a probability measure* $\mu \in \mathcal{P}(\Theta)$ *and some dominating measure* $\lambda \in \mathcal{P}(\Theta)$ *on the measurable space* $(\Theta, \mathcal{E})$*, then by the Radon–Nikodym theorem there exists a* λ*-measurable set function* $f(\theta)$ *unique up to sets of measure 0 in* λ*, such that for all sets* $E \in \mathcal{E}$

$$\mu(E) = \int_E f(\theta)\, d\lambda(\theta) \tag{2.1}$$

with $0 < f(\theta) < \infty$ *λ–almost–surely. We call* $f(\theta)$ *a probability distribution in* $\mathcal{P}(\Theta)$*.*

Our aim is to find some probability distribution $q_B^* \in \mathcal{P}(\Theta)$ that, given some observed data $x_{1:n}$ from a measurable space $(\Xi, \mathcal{X})$, models the data generating process, Shalizi (2009). We can infer such distributions through Bayes' theorem (Rogers and Girolami, 2016; Bishop, 2006), where we denote our prior knowledge about some parameter $\theta \in \Theta$ as a prior distribution $\pi(\theta)$, and connect this parameter θ to the data $x_{1:n}$ through the likelihood function $p_n(x_{1:n}|\theta)$ (Knoblauch, 2019)[3]. We further define the marginal likelihood $p(x_{1:n}) = \int_\Theta p(x_{1:n}|\theta) d\pi(\theta)$ which acts as a normalisation constant to ensure the posterior distribution $q_B^*(\theta|x_{1:n})$ is a proper probability distribution.

$$q_B^*(\theta|x_{1:n}) = \frac{\pi(\theta)\, p_n(x_{1:n}|\theta)}{p(x_{1:n})} \tag{2.2}$$

However, this is usually, i.e. for most interesting, practical applications, intractable to compute explicitly, making exact inference impossible (Rogers and Girolami, 2016). This intractability comes from the normalising constant $p(x_{1:n})$, also known as the model evidence. Due to typically having high dimensional data, and hence parameters, or large amounts of data, making the integral infeasible to compute. To this end, we first explore how we can reframe Bayes' theorem as an optimisation problem, and then how Variational Inference can serve as a tractable approximation to the Bayesian posterior.

2.1.1 Optimisation Centric view of Bayes' Theorem

In (Zellner, 1988), the author derives Bayes' theorem as an optimal "information processing rule" (IPR) by considering the information before observations and the information after

3. The subscript n for the likelihood indicates the size of the data, on which it depends (Miller, 2021).

observations. Assuming that we have some prior information on our parameter of interest, $\pi(\theta)$, and we know the likelihood of observing some data $x_{1:n}$, $p(x_{1:n}|\theta)$, then after applying some information processing rule, which we will show to be Bayes' theorem, then we can get the output information in terms of the marginal likelihood $p(x_{1:n})$ and some "post–data" distribution $q(\theta|x_{1:n})$ (Zellner, 1988). As a measure of information, Zellner (1988) chose the cross entropy (Kullback, 1959) between the "post–data" distribution and all terms, defined as $\mathcal{H}_q(\cdot) = \mathbb{E}_q[\log \cdot]$, where $\mathbb{E}_q[\cdot]$ is the expectation of something with respect to distribution q. And per the definition of an optimal IPR, the input information needs to equal the output information, Zellner (1988).

$$\mathcal{H}_{q(\theta|x_{1:n})}(\pi(\theta)) + \mathcal{H}_{q(\theta|x_{1:n})}(p(x_{1:n}|\theta)) = \mathcal{H}_{q(\theta|x_{1:n})}(p(x_{1:n})) + \mathcal{H}_{q(\theta|x_{1:n})}(q(\theta|x_{1:n}))$$

Rearranging this through the laws of expectation and the usual rules of logarithms, we get the following formulation:

$$\begin{aligned}
0 &= \mathbb{E}_{q(\theta|x_{1:n})}[\log \pi(\theta)] + \mathbb{E}_{q(\theta|x_{1:n})}[\log p(x_{1:n}|\theta)] \\
&\quad - \big(\mathbb{E}_{q(\theta|x_{1:n})}[\log p(x_{1:n})] + \mathbb{E}_{q(\theta|x_{1:n})}[\log q(\theta|x_{1:n})]\big) \\
0 &= \mathbb{E}_{q(\theta|x_{1:n})}[\log \pi(\theta)p(x_{1:n}|\theta) - \log p(x_{1:n})q(\theta|x_{1:n})] \\
0 &= \mathbb{E}_{q(\theta|x_{1:n})}\Big[\log \frac{\pi(\theta)p(D|\theta)}{p(x_{1:n})q(\theta|x_{1:n})}\Big] = \mathrm{KL}\Big(q(\theta|x_{1:n})\Big|\Big|\frac{\pi(\theta)p(x_{1:n}|\theta)}{p(x_{1:n})}\Big)
\end{aligned}$$

This result recovers something called the Kullback–Leibler Divergence, introduced in Kullback and Leibler (1951), a strictly non-negative functional, which equals zero if and only if the two inputs are equal. That is, the minimum 0 is obtained if and only if this "post–data" is equal to the Bayesian posterior, $q(\theta|x_{1:n}) = q^*_B(\theta|x_{1:n}) = \frac{\pi(\theta)p(x_{1:n}|\theta)}{p(x_{1:n})}$.

Definition 3 (Kullback–Leibler Divergence) *For probability measure spaces $(\Theta, \mathcal{E}, \mu_1)$ and $(\Theta, \mathcal{E}, \mu_2)$, with $\mu_1, \mu_2 \in \mathcal{P}(\Theta)$, and a dominating measure $\lambda \in \mathcal{P}(\Theta)$, let the probability distributions $f_i(\theta) = \frac{d\mu_i}{d\lambda}(\theta)$ for $i = 1, 2$ be characterised by their unique Radon–Nikodym derivatives, then the Kullback–Leibler divergence between f_1 and f_2 is defined as*

$$\mathrm{KL}(f_1||f_2) = \int_\Theta f_1(\theta) \log \frac{f_1(\theta)}{f_2(\theta)}\, d\lambda(\theta) \tag{2.3}$$

Theorem 4 *The cross–entropy term chosen as a measure of information with respect to the posterior distribution is equivalent to choosing the Kullback–Leibler Divergence between $q(\theta|x_{1:n})$ and the inputs, and outputs of the information processing rule.*

Proof We show that the KL Divergence decomposes into an entropy and a cross–entropy term

$$\begin{aligned}
\mathrm{KL}(f_1||f_2) &= \int f_1 \log \frac{f_1}{f_2} d\lambda = \int f_1 \log f_1 d\lambda - \int f_1 \log f_2 d\lambda \\
\mathrm{KL}(f_1||f_2) &= \mathbb{E}_{f_1}[f_1] - \mathbb{E}_{f_1}[f_2]
\end{aligned}$$

The entropy terms $\mathbb{E}_{f_1}[f_1]$ then cancel out since the number of inputs and outputs are equal and $f_1 = q(\theta|x_{1:n})$ remains the same in all terms.

$$\begin{aligned} &\mathrm{KL}(q(\theta|x_{1:n})||\pi(\theta)) + \mathrm{KL}(q(\theta|x_{1:n})||p(x_{1:n}|\theta)) \\ &= \mathrm{KL}(q(\theta|x_{1:n})||p(x_{1:n})) + \mathrm{KL}(q(\theta|x_{1:n})||q(\theta|x_{1:n})) \\ \iff &\mathcal{H}_{q(\theta|x_{1:n})}(\pi(\theta)) + \mathcal{H}_{q(\theta|x_{1:n})}(p(x_{1:n}|\theta)) = \mathcal{H}_{q(\theta|x_{1:n})}(p(x_{1:n})) + \mathcal{H}_{q(\theta|x_{1:n})}(q(\theta|x_{1:n})) \end{aligned}$$

Hence, showing that the KL divergence can be used instead. ■

The result of Zellner (1988) is significant, since it implies that we can reframe Bayesian updating as an infinite dimensional optimization problem—which was most likely known in different form at least in Csiszar (1975)—where we find the KL minimizer from the, perhaps infinite dimensional, set $\mathcal{P}(\Theta)$.

$$q_B^*(\theta|x_{1:n}) = \underset{q(\theta)\in\mathcal{P}(\Theta)}{\arg\inf} \left\{ \mathrm{KL}\Big(q(\theta) \Big|\Big| \frac{\pi(\theta) p_n(x_{1:n}|\theta)}{p(x_{1:n})} \Big) \right\} \tag{2.4}$$

This result is instrumental, not only as a justification of Bayes' theorem, but also establishes the foundation of Knoblauch et al. (2022), on which we build in this thesis to derive our novel federated learning algorithm, Partitioned Generalized Variational Inference.

2.1.2 Variational Inference

Since we cannot explicitly evaluate the marginal likelihood $p(x_{1:n})$, due to having to evaluate intractable, potentially high-dimensional, integrals over complicated likelihood functions $p_n(x_{1:n}|\theta)$ with respect to the prior measure $\pi(\theta)$, we are interested in bounding this probability, Blei et al. (2016), and Rogers and Girolami (2016). In order to do this we make use of Jensen's inequality, a standard result in measure theory, Rudin (1987).

Definition 5 (Jensen's Inequality) *For a probability measure space $(\Theta, \mathcal{E}, \mu)$, a μ measurable function $p : \Theta \to \mathbb{R}_{\geq 0}$, and a convex function $\phi : \mathbb{R} \to \mathbb{R}$, then for all $A \in \mathcal{E}$ the following inequality holds,*

$$\phi\Big(\int_A p(\theta)\, d\mu(\theta) \Big) \leq \int_A \phi(p(\theta))\, d\mu(\theta)$$

We can apply this inequality to lower bound the log marginal likelihood. In order to do this, we consider the marginal likelihood as the integral of the likelihood with respect to the prior distribution, $p(x_{1:n}) = \int_\Theta p_n(x_{1:n}|\theta)\, d\pi(\theta)$, and introduce an arbitrary probability distribution $q(\theta) \in \mathcal{P}(\Theta)$.

$$p(x_{1:n}) = \int_\Theta \frac{p_n(x_{1:n}|\theta)\, q(\theta)}{q(\theta)}\, d\pi(\theta)$$

Taking logarithms of both sides and applying Jensen's inequality yields the Evidence Lower Bound (ELBO), Rogers and Girolami (2016).

$$\log p(x_{1:n}) = \log \int_\Theta \frac{p_n(x_{1:n}|\theta)\pi(\theta)}{q(\theta)}\, dq(\theta) \geq \int_\Theta \log\Big(\frac{p_n(x_{1:n}|\theta)\pi(\theta)}{q(\theta)}\Big)\, dq(\theta) = \text{ELBO}(q)$$

The usage of the logarithm with Jensen's inequality and the reversal of the inequality sign is because log is a concave function and multiplying it by minus one makes it convex, Amari (2016), while reversing the inequality sign of Jensen's inequality. Additionally, an interesting connection arises in this derivation of Variational Inference (VI), since the ELBO is equivalent the negative Kullback–Leibler divergence.

$$\begin{aligned}\text{ELBO}(q) &= \int_\Theta \log\Big(\frac{p_n(x_{1:n}|\theta)\pi(\theta)}{q(\theta)}\Big)\, dq(\theta) = \int_\Theta \big(\log\left(p_n(x_{1:n}|\theta)\pi(\theta)\right) - \log q(\theta)\big)\, dq(\theta) \\ &= -\int_\Theta q(\theta) \log\Big(\frac{q(\theta)}{p_n(x_{1:n}|\theta)\pi(\theta)}\Big)\, d\theta = -\text{KL}(q(\theta)||p_n(x_{1:n}|\theta)\pi(\theta))\end{aligned}$$

Remark 6 *It is important to highlight the fact that the Kullback–Leibler divergence is not symmetric. This means that* $\text{KL}(q||p) \neq \text{KL}(p||q)$ *in general, Nielsen (2020).*

In VI we focus on minimising $\text{KL}(q||q_B^*)$ rather than $\text{KL}(q_B^*||q)$, which is explored in Expectation Propagation and we call the reverse KL divergence, see chapter 2.1.3 for details.

Maximising the ELBO, or alternatively minimising the positive KL divergence, yields an approximation to the Bayesian posterior, Zhang et al. (2019). This is due to the fact that the difference between the log marginal likelihood and the ELBO is equivalent to the KL divergence between $q(\theta)$ and $q_B^*(\theta)$. This relies on the property that $q(\theta)$ is a proper probability distribution, i.e. $\int_\Theta q(\theta)\, d\mu(\theta) = 1$, with respect to the dominating measure μ.

$$\begin{aligned}\log p(x_{1:n}) - \text{ELBO}(q) &= \log p(x_{1:n}) + \text{KL}(q(\theta)||p_n(x_{1:n}|\theta)\pi(\theta)) \\ &= \log p(x_{1:n}) + \int_\Theta q(\theta) \log\Big(\frac{q(\theta)}{p_n(x_{1:n}|\theta)\pi(\theta)}\Big)\, d\mu(\theta) \\ &= \log p(x_{1:n}) \int_\Theta q(\theta)\, d\mu(\theta) + \int_\Theta q(\theta) \log\Big(\frac{q(\theta)}{p_n(x_{1:n}|\theta)\pi(\theta)}\Big)\, d\mu(\theta) \\ &= \int_\Theta q(\theta) \log p(x_{1:n}) + q(\theta) \log\Big(\frac{q(\theta)}{p_n(x_{1:n}|\theta)\pi(\theta)}\Big)\, d\mu(\theta) \\ &= \int_\Theta q(\theta) \log\Big(\frac{p(x_{1:n})q(\theta)}{p_n(x_{1:n}|\theta)\pi(\theta)}\Big)\, d\mu(\theta) = \text{KL}\Big(q(\theta)\Big|\Big|\frac{p_n(x_{1:n}|\theta)\ \pi(\theta)}{p(x_{1:n})}\Big)\end{aligned}$$

This is equivalent to the formulation in Zellner (1988), where Bayesian updating is viewed as an infinite dimensional optimisation problem. However, VI has two key differences, and different motivation. The first being that we do not have to include the intractable marginal likelihood in our optimisation problem since the term does not depend on the choice of $q(\theta)$, and the second being the choice of $q(\theta)$. Previously, we have assumed to optimise over the entirety of $\mathcal{P}(\Theta)$, however Variational Inference aims to approximate an intractable posterior over a subset $\mathcal{Q} \subset \mathcal{P}(\Theta)$, Bishop (2006). Figure 2.1 illustrates

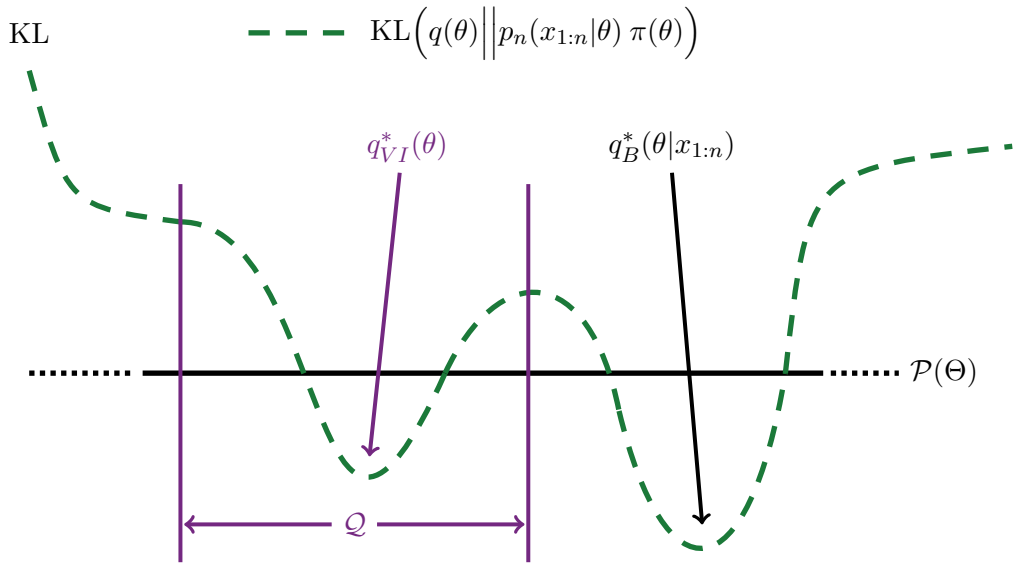


Figure 2.1: Visualisation of Variational Inference, reproduced from Knoblauch et al. (2022). The figure illustrates the **KL** divergence value over all permissible distributions, which is minimized at the **Bayesian** posterior q^*_B. The **VI** posterior q^*_{VI} is minimized over a restricted subset $\mathcal{Q}$ of tractable distributions and is closest to q^*_B in terms of KL divergence.

the approximation of Variational Inference; the VI posterior, from tractable set $\mathcal{Q}$, is the distribution which is closest to the Bayesian posterior in terms of KL divergence, inside $\mathcal{Q}$. The approximation might not be the exact posterior, however it is the best approximation inside some subset of possible distributions, and through the choice of $\mathcal{Q}$ it can be made an excellent approximation. Hence making computation significantly faster, or in some cases even possible, while still achieving high accuracy, Zhang et al. (2019), Blei et al. (2016). Therefore, VI is an optimisation problem and can be formulated as minimizing the following objective.

$$q^*_B(\theta|x_{1:n}) \approx q^*_{VI}(\theta) = \underset{q(\theta)\in\mathcal{Q}}{\arg\min} \Big\{ \mathrm{KL}\big(q(\theta)\big|\big|p_n(x_{1:n}|\theta)\pi(\theta)\big)\Big\} \tag{2.5}$$

Lower bounding the log marginal likelihood is not the only motivation or direction in which we can derive VI. Blei et al. (2016) views VI as a divergence minimisation problem, using the known Kullback–Leibler divergence, and minimising an approximation to the Bayesian posterior in the style of Zellner (1988), however restricting the space of possible distributions to optimize over. This approach effectively arrives at the ELBO maximization we used previously in reverse order.

Another different derivation of VI stems from statistical Physics, where we derive the ELBO(q) through something called the Variational Free Energy, originating from Gibbs' inequality, MacKay (2003). This states that the entropy term of one distribution is greater than or equal to the cross entropy to a different distribution, and formulated for discrete

distributions Q and P states.

$$\sum_x Q(x) \log Q(x) \geq \sum_x Q(x) \log P(x) \iff \mathrm{KL}(Q||P) \geq 0$$

Then, the variational free energy $\mathcal{F}(q)$, generalized to continuous distributions, and applicable to discrete distributions through choosing μ to be the counting measure, is the following:

$$\mathcal{F}(q) = \mathrm{KL}\big(q(\theta)||p_n(x_{1:n}|\theta)\pi(\theta)\big) = \int q(\theta) \log \frac{q(\theta)}{p_n(x_{1:n}|\theta)\pi(\theta)}\, d\mu(\theta) \geq 0$$

Even though we arrive at the same conclusion, VI is motivated here through minimising the free energy between some approximation to the Bayesian posterior (in the sense as before) through Gibbs inequality rather than as a lower bound on the marginal likelihood or as a KL divergence minimisation problem.

Since the likelihood function and the prior are typically not conjugate (Rogers and Girolami, 2016), computation can be significantly improved when rearranging the KL divergence as the following equation.

$$q^*_{VI}(\theta) = \underset{q(\theta)\in\mathcal{Q}}{\arg\min} \Big\{ \mathbb{E}_{q(\theta)}\big[-\log p_n(x_{1:n}|\theta)\big] + \mathrm{KL}\big(q(\theta)\big|\big|\pi(\theta)\big) \Big\} \tag{2.6}$$

This interpretation of VI allows us to highlight that if the data is independent and identically distributed (i.i.d.), exchangeable (Walker, 2013) , or has some multiplicative structure in the likelihood function, then the logarithm of the product of the individual likelihood terms decomposes into the sum of the log likelihoods of each data point. For instance, assume that the data is from some Markov Chain distribution, such as a time series, where the t^{th} element depends on all the previous terms, such that the likelihood function is of the following form

$$p_n(x_{1:n}|\theta) = \prod_{t=1}^{n} p_t(x_t|\theta, x_{t-1}, x_{t-2}, ..., x_1)$$

then, taking the logarithm in equation 2.6 replaces multiplication with summation.

$$\log \prod_{t=1}^{n} p_t(x_t|\theta, x_{t-1}, x_{t-2}, ..., x_1) = \sum_{t=1}^{n} \log p_t(x_t|\theta, x_{t-1}, x_{t-2}, ..., x_1)$$

However, in general the data generating process need not be i.i.d. or even Markovian, such as in pushdown automata, see Shalizi (2009) and references therein. In chapter 4 we will explore how assuming some simpler structure can be acceptable even without knowing the correct likelihood structure. We illustrate the VI framework in algorithm 1 for i.i.d. data.

To make Variational Inference an attractive approximate inference framework, we need a variational family $\mathcal{Q}$, that is sufficiently rich. In VI we consider this family to be a set of distributions on parameters $\theta \in \Theta$ that depend on some variational parameters κ, as given

Algorithm 1 Variational Inference

Inputs: Data $x_{1:n}$, prior $\pi(\theta)$, likelihood function $p_i(x_i|\theta)$, and variational family $\mathcal{Q}$

$$q(\theta) \leftarrow \underset{q(\theta)\in\mathcal{Q}}{\arg\min}\Big\{\mathbb{E}_{q(\theta)}\Big[\sum_{i=1}^{n}\log p_i(x_i|\theta)\Big] + \mathrm{KL}\big(q(\theta)\big|\big|\pi(\theta)\big)\Big\}$$

Output: $q(\theta)$

by the family of distributions we want to optimise over, Knoblauch (2019). An example of such families is the set of Multivariate Gaussian distributions with a mean vector and covariance matrix. Furthermore, each data point might depend on some hidden, unobserved latent variables, not necessarily the same across data points, Hoffman et al. (2013). For data point i, we denote the set of latent variables by $z_{s(i)}$. The dependencies of data points on latent variables and global parameters, as well as variational parameters, is illustrated in figure 2.2. The likelihood function then becomes $p_n(x_{1:n}, z_{s(1:n)}|\theta)$, Knoblauch (2019). The following example demonstrates latent variables as the mixture components of a mixture of Gaussian distributions, Attias (1999).

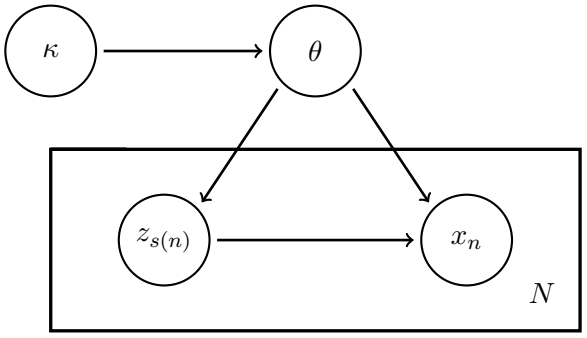


Figure 2.2: Illustrative example of a plate diagram of dependence in Variational inference of observed data $x_{1:n}$, latent variables $z_{s(1:n)}$, parameters θ and variational parameters κ. Each data point x_n, for $n \in [N]$, depends on a local set of latent variables $z_{s(n)}$, and global parameters θ. Reproduced with alterations from Hoffman et al. (2013).

Example 1 *Consider the mixture of M Multivariate Gaussian distributions, McLachlan and Peel (2000), each with distribution $m_i(x) \sim \mathcal{N}(\mu_i, \Sigma_i)$ and probability of being chosen $w_i \in (0,1)$ such that $\sum_i w_i = 1$. Then the mixture of Gaussians is defined as $p(x) = \sum_i w_i\, m_i(x)$. The unobserved latent variable w_i determines the parameters of the distribution of some data point, however is not directly observed. For instance, assume that we have two Gaussians in $\mathbb{R}^2$, one with mean $\mu_1 = (5,5)^\top$ and the other having mean $\mu_2 = (0,0)^\top$, with identity matrix covariance matrices and equal probability of being chosen. Then the latent variables, i.e. which mixture component was chosen to generate the data, determines the parameters of the distribution of each data point.*

To infer a dependent approximate posterior $q(\theta, z_{s(1:n)})$ about some parameters θ and latent variables $z_{s(1:n)}$ is computationally demanding, Rogers and Girolami (2016), Bishop (2006) and Blei et al. (2016). Therefore, we typically assume independence between parameters

and latent variables.

$$q(\theta, z_{s(1:n)}) = q_1(\theta) q_2(z_{s(1:n)})$$

This is called Mean–Field Variational Inference, where we assume some independence of parameters of our approximating distribution, Blei et al. (2016). This allows us to break the integral over $q(\theta, z_{s(1:n)})$ into a double integral over parameters and latent variables, and by Fubini's theorem (Rudin, 1987), we can choose the order of integration, making computation significantly easier. We could further assume that individual components of θ or of $z_{s(1:n)}$ are independent, and hence decompose $q_1(\theta)$ or $q_2(z_{s(1:n)})$ into further multiplications, however abstracting away more structure of the underlying model makes the approximation less accurate, Rogers and Girolami (2016). This implies that we have a trade off between computational complexity and accuracy in our posterior approximation when choosing the Mean–Field VI framework.

This also leads to Structured Variational Inference, where we do not assume that all parameters are fully independent but rather that there are substructures within parameters and latent variables, Murphy (2023), Aglietti et al. (2019). We want to exploit the underlying model structure to decompose our approximate posterior into independent structures that have some dependencies, with some latent variables potentially having no dependencies, whilst others might depend on these, as in Aglietti et al. (2019) for continuous cox process models. As an illustrating example—and without delving into what is actually occurring—we borrow the structure of the model in Aglietti et al. (2019), where we want to approximate the posterior distribution $p(\boldsymbol{u}, \boldsymbol{f}, M, \boldsymbol{y}_{1:n}, \lambda^* | D)$ over data D, by some distribution $q(\boldsymbol{u}, \boldsymbol{f}, M, \boldsymbol{y}_{1:n}, \lambda^*)$. Then, we can represent the structure of the underlying model as the structured VI objective.

$$q(\boldsymbol{u}, \boldsymbol{f}, M, \boldsymbol{y}_{1:n}, \lambda^*) = q(\boldsymbol{f}|\boldsymbol{u})\, q(\boldsymbol{y}_{1:n}|M)\, q(M|\boldsymbol{f}, \lambda^*)\, q(\boldsymbol{u})\, q(\lambda^*)$$

Aglietti et al. (2019) shows that this significantly aids computation, performs well in comparison to stochastic approximate inference techniques, and improves uncertainty quantification. For more details on improving traditional VI, for instance through tighter bounds on the log marginal likelihoods, see chapter 10 of Murphy (2023) and references therein.

In the remainder we consider $x_{1:n}$ to be representative of both the data and local latent variables, i.e. $x_i \equiv (x_i, z_{s(i)})$, extending the measurable space $(\Xi, \mathcal{X})$ to define $\mathcal{X} \equiv (\mathcal{X} \times \mathcal{Z}^{s(1:n)})$, where $\mathcal{Z}^{s(1:n)}$ is the space of latent variables, Knoblauch (2019). This is done in order to lighten notation and explain existing theory as well as theoretical contributions more clearly. We will make the latent variables explicit where necessary, and where they are a key part of inference.

2.1.3 Expectation Propagation

In Expectation Propagation (EP) our objective is to minimize the reverse KL divergence, $\mathrm{KL}(q_B^*(\theta|x_{1:n})||q(\theta))$, Minka (2001b). And in difference to VI we do this global optimisation in a distributed fashion, where we distribute computation across all data points. To this

end EP introduces approximate likelihood terms $t_i(\theta)$ for $i \in [1, ..., n]$ that model $p_i(x_i|\theta)$ respectively, Minka (2001a,b). This approach assumes that the data is i.i.d. or exchangeable and hence factorises over the n likelihood terms, such that we can approximate q_B^* using equation 2.7.

$$q_B^*(\theta|x_{1:n}) = \frac{1}{Z}\,\pi(\theta)\prod_{i=1}^{n} p_i(x_i|\theta) \approx \frac{1}{\tilde{Z}}\,\pi(\theta)\prod_{i=1}^{n} t_i(\theta) = q_{EP}^*(\theta) \tag{2.7}$$

where Z and $\tilde{Z}$ are the normalizing constants, i.e. marginal likelihoods, for the respective posteriors. Our aim is to minimize the following KL divergence:

$$\mathrm{KL}(q_B^*(\theta|x_{1:n})||q_{EP}^*(\theta)) = \mathrm{KL}\Big(\frac{1}{Z}\,\pi(\theta)\prod_{i=1}^{n} p_i(x_i|\theta)\Big|\Big|\frac{1}{\tilde{Z}}\,\pi(\theta)\prod_{i=1}^{n} t_i(\theta)\Big)$$

The distributed nature of EP becomes relevant when we consider the optimisation of an approximating likelihood $t_i(\theta)$. EP iterates through the data points x_i, for $i \in [n]$, either sequentially or synchronously, and refines the global posterior approximation $q_{EP}^*(\theta)$ locally through the use of the cavity distribution, Vehtari et al. (2020).

Definition 7 (Cavity Distribution) *Consider a set up as in equation 2.7, then the cavity distribution denoted $q^{\setminus k}(\theta)$, for data point x_k is given by*

$$q^{\setminus k}(\theta) \propto \frac{q_{EP}^*(\theta)}{t_k(\theta)} = \pi(\theta)\prod_{i\neq k} t_i(\theta)$$

The cavity distribution aims at modelling the impact of the remaining data, and prior, on the posterior without explicitly evaluating the likelihood function for all remaining data points. Locally, we then compute the KL divergence between the likelihood term of the local data point multiplied by the cavity distribution to a new approximation in the variational family $\mathcal{Q}$, preserving the structure of the EP posterior, Minka (2001b). After finding an updated global approximation through the local KL minimisation, we can recover the new local approximate likelihood term through dividing the new approximation by the cavity distribution, effectively taking out the prior as well as the other approximate likelihood factors from the local posterior.

Expectation Propagation is an attractive framework, although lacking in theoretical guarantees, mainly due to it's speed and empirical success. The speed of EP stems from the set of distributions we consider, and restrict ourselves to. In particular EP is restricted to use exponential family distributions, see Bishop (2006) or Murphy (2022, 2023).

Definition 8 (Exponential Family) *We say a probability distribution $\pi_\eta(\theta)$, over parameters θ and natural parameters η, is in the exponential family if it can be written as*

$$\pi_\eta(\theta) = \frac{1}{Z(\eta)} h(\theta)\exp\{\eta^\top \phi(\theta)\} \tag{2.8}$$

where $Z(\eta)$ is the log partition function, $h(\theta)$ is some scaling constant, and $\phi(\theta)$ is the set of sufficient statistics.

Algorithm 2 Expectation Propagation

Inputs: Data $x_{1:n}$, prior $\pi(\theta)$, likelihood function $p_i(x_i|\theta)$, and variational family $\mathcal{Q}$.

Set $k = 0$
Initialise: For each $i \in [n]$, set $t_i^{(k)}(\theta) := 1$, such that $q^{(k)}(\theta) := \pi(\theta) \prod_{i=1}^n t_i^{(k)}(\theta) = \pi(\theta)$

while Algorithm 2 has not converged **do**:
  for $i \in 1, ..., n$ **do**
    Compute the cavity distribution:

$$q^{\setminus i}(\theta) \leftarrow \frac{q^{(k)}(\theta)}{t_i(\theta)}$$

    Compute the new approximate posterior through the KL divergence:

$$q^{(k+1)}(\theta) \leftarrow \underset{q(\theta)\in\mathcal{Q}}{\arg\min}\, \mathrm{KL}\big(p_i(x_i|\theta)\, q^{\setminus i}(\theta)\big|\big|q(\theta)\big)$$

    Update the approximate likelihood for data point i:

$$t_i^{(k+1)}(\theta) \propto \frac{q^{(k+1)}(\theta)}{q^{\setminus i}(\theta)}$$

    $k \leftarrow k + 1$
  end for
end while

We typically consider $h(\theta)$ to be 1 or use it as a base measure for integration. Some common examples of exponential family distributions are the (multivariate) Gaussians, the exponential distribution, or the categorical distribution, Amari (2016). Not only do they encompass many widely used distributions, but also offer attractive properties that make them a nice family of distributions to study. In expectation propagation they enable fast computation of the minimiser of the Kullback–Leibler divergence in algorithm 2, as formalised in the theorem below, due to Minka (2004), see also Bishop (2006).

Theorem 9 *The Kullback Leibler minimisation in Expectation Propagation over exponential family distributions with the same sufficient statistics $\phi(\theta)$ reduces to moment matching of the common sufficient statistics.*

$$q^{\text{new}}(\theta) = \underset{q(\theta)\in\mathcal{Q}}{\arg\min}\, \mathrm{KL}\big(p_i(x_i|\theta)q^{\setminus i}(\theta)\big|\big|q(\theta)\big) \iff \mathbb{E}_{q^{\text{new}}(\theta)}[\phi(\theta)] = \mathbb{E}_{p_i(x_i|\theta)q^{\setminus i}(\theta)}[\phi(\theta)]$$

The proof, due to Herbich (2005), is somewhat tedious and long, and is therefore deferred to Appendix B, we give only an outline here. The proof relies on the facts that the gradient vector of partial derivatives with respect to the natural parameters η of the log partition function $Z(\eta)$ of $q_\eta^{\text{new}}(\theta)$ is equivalent to $\mathbb{E}_{q_\eta^{\text{new}}(\theta)}[\phi(\theta)]$, and that the KL divergence is minimised where it's gradient, with respect to η, is 0, Bishop (2006) and Amari (2016). We can then show that the this is a minimum by computing the Hessian matrix of second derivatives of the KL divergence, showing that this is equivalent to the covariance matrix

of $q_\eta^{\text{new}}(\theta)$, hence positive definite and therefore a minimum. We include the proof in the appendix since we have filled in some of the, non–obvious, details in Herbich (2005), and to illustrate the usefulness of the exponential family.

This result implies that for many useful distributions and models, we can effectively compute updates in at most $O(D^3)$ time, D being the dimension of the vector θ, Rasmussen and Williams (2006). It, however, also demonstrates potential weaknesses of EP, such as having to compute matrix inversions, which is not numerically stable, Hasenclever et al. (2017). EP using moment matching is demonstrated in figure 2.3 for a single update.

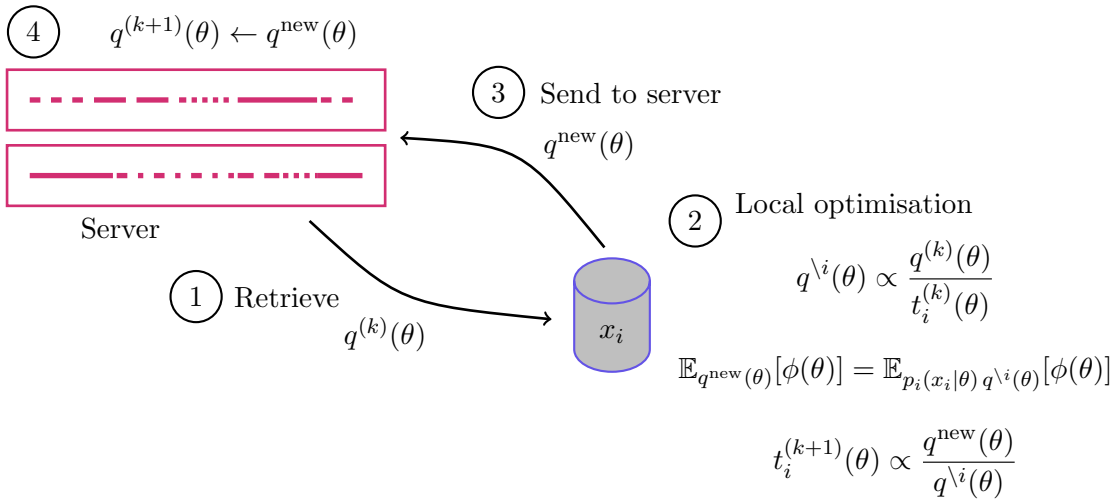


Figure 2.3: Visualisation of a single EP iteration with moment matching, as in theorem 9. We consider data point x_i at some iteration k.

Expectation Propagation as a framework has extensively been studied and extended, Minka (2001a, 2004, 2005), Vehtari et al. (2020), Heskes et al. (2005), Bishop (2006), Hasenclever et al. (2017) and Guo et al. (2023), to name a few. This is due to it being a fast and often highly accurate algorithm, if it converges. Vanilla EP is only guaranteed to converge to some local minimum of the loss landscape, however it might not converge to a global minimum, Minka (2001b). Nevertheless, extensions such as those in Hasenclever et al. (2017) and Vehtari et al. (2020) that improve on convergence guarantees of EP, it's distributed nature, and the empirical success in practice, motivate further study. It is the algorithm that Bui et al. (2018) used to motivate and derive Partitioned Variational Inference and subsequently this thesis, to develop our novel federated learning algorithm termed Partitioned Generalised Variational Inference.

2.2 Distributed and Federated Learning Literature Review

Federated Learning, as introduced in McMahan et al. (2017), is the decentralised but collaborative training of a global machine learning model over individual, private, and local data sets. Federated learning is distinctly different to distributed learning, since distributed computation typically assumes a massive data set that is distributed across machines in a data centre, where a single machine cannot store the entire data set, Mesquita et al. (2020). In federated learning we typically assume that the data is owned and stored across different

clients, which might not communicate directly, and cannot exchange data. We illustrate a general federated learning set–up in figure 2.4 where clients communicate with an organising server that does not see client data and only receives model updates and shares current models with clients. Clients locally optimise the model using their data sets without communicating data to the server or other clients.

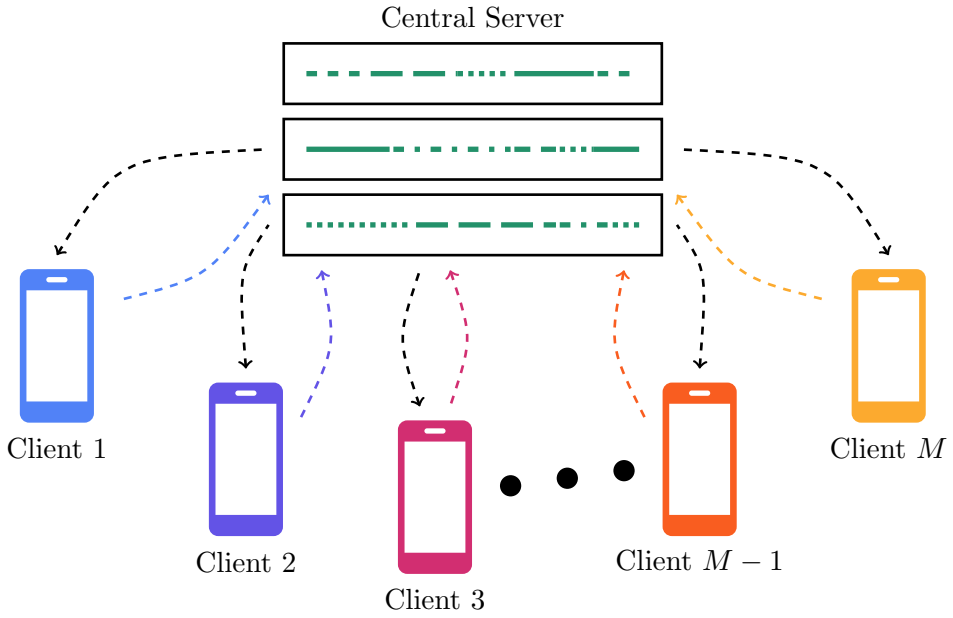


Figure 2.4: Graphical visualisation of **Federated Learning** of M clients scheduled by a central server. Dashed arrows indicate model updates being exchanged.

Since federated learning is primarily concerned with keeping data locally with clients, the immediate challenges are regarding algorithmic development, creating distributed algorithms that can recover global results consistently. However, doing this creates further key challenges that can occur in federated settings in order to keep data locally private and cope with challenging distributed systems. The key differences that research on federated learning addresses in comparison to distributed learning are shown across the top row of figure 2.5 and explored in chapter 2.2.1. We further review existing algorithmic approaches to distributed and federated learning, a selection of which is shown in figure 2.5, in order to understand how the Partitioned Variational Inference approach, which we introduce in chapter 3, compares to existing literature, while investigating algorithmic design choices for our development of Partitioned Generalised Variational Inference.

2.2.1 Challenges addressed by Federated Learning

The key challenges of federated learning, as first discussed in the seminal paper of McMahan et al. (2017), and further developed since, see for instance Kairouz et al. (2021), can be summarised as, but are not limited to, the following:

Algorithmic Frameworks We require algorithms that are able to use local data to infer a global model, such as Neural Networks, which perform as well as a global

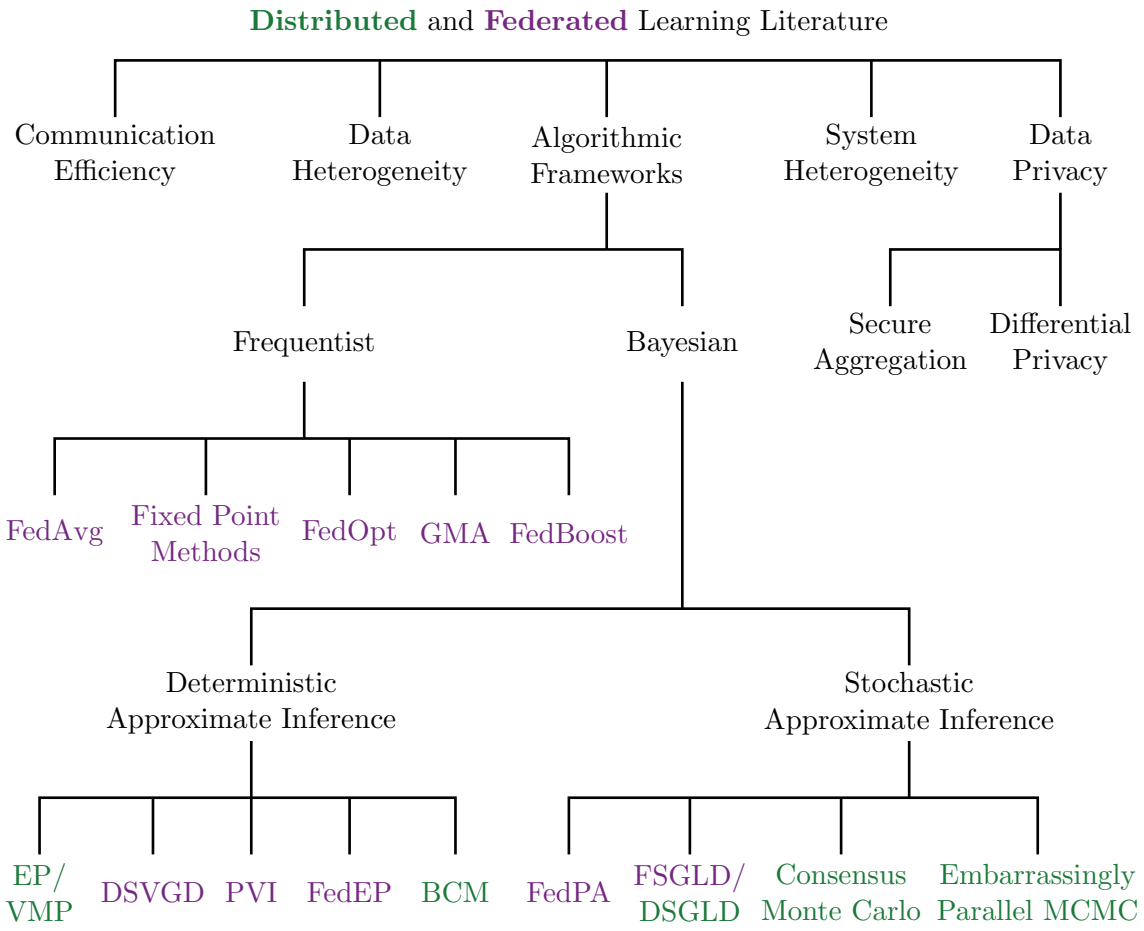


Figure 2.5: Best viewed in colour. Tree showing a selection of **Distributed** and **Federated** Learning algorithms. The top row represents key challenges that Federated Learning addresses. The algorithmic frameworks are split into the key approaches that are commonly used for Federated and Distributed Learning, namely Frequentist and Bayesian. The Bayesian Approaches are further split into Deterministic and Approximate Inference techniques.

approach in order to justify the use of federated learning in comparison to centralised data collection.

Theoretical Guarantees To this end, we require theory that explains why some federated algorithms perform well empirically and others do not. Proving consistency, coverage, and convergence, especially in probabilistic federated learning, is challenging and little work has yet examined this and concluded with rigorous results.

Client Heterogeneity Clients can be supercomputers of organisations or smartphones of individuals, hence potentially having limited computation capabilities, and also different amount of computation time, meaning updating models may take longer for some clients than others. Furthermore, clients might not always be available for computation.

Data Heterogeneity The data quality, amount, and distribution might vary across clients, meaning we cannot typically assume that the data is i.i.d.. For instance,

consider a handwritten number classification model, then clients might be individuals with different handwriting and different amounts of text written.

Distributed Systems The amount of clients can vary from very few, as in data centres or across organisations, to large numbers, such as all smart phones or computers that use certain applications. Therefore, we need to be able to cope with being massively distributed. Furthermore, we need not assume a central server exists but rather clients that communicate directly with each other, as in peer–to–peer federated learning, Kairouz et al. (2021).

Communication and Computation Local clients need to be able to communicate with a central server or each other. However, the communication bandwidth might be limited due to poor WiFi. Furthermore, they might be limited in their computational power making algorithms that require large time complexity intractable for some application domains, and potentially only suitable for corporations that have access to, e.g. GPU clusters.

Privacy FL aims to keep data locally private against malicious actors. Therefore we need to investigate both potential weaknesses, as in Zhu et al. (2019) or Zhao et al. (2023), as well as strategies to mitigate privacy risks such as state of the art methods of differential privacy and secure aggregation, Chen et al. (2022).

Trust How much trust can individual clients place in other clients or a central server? Servers should not be able to recover client data from communications, as with secure aggregation, Chen et al. (2022). We need to be able to mitigate the impact of updates from clients with erroneous or noisy data, who potentially want to impede the development of the global model or do not possess qualitative data.

In difference to what figure 2.5 suggests, research in federated learning is heavily concerned with the study of frequentist approaches to algorithmic development and hence most papers that deal with challenges of federated learning are concerned with improving these frequentist algorithms, in particular the federated averaging algorithm of McMahan et al. (2017), as well as it's generalisation as in Reddi et al. (2021). Therefore, we begin by reviewing frequentist approaches to federated learning and explore how these deal with these challenges listed above, before moving on to the relatively unexplored and underdeveloped Bayesian approaches.

For further details on challenges, advances, and current open problems in federated learning, see Kairouz et al. (2021).

2.2.2 Frequentist Approaches

Federated Stochastic Gradient Descent (FedSGD) and Federated Averaging (FedAvg) were first introduced in McMahan et al. (2017) and are based on local stochastic gradient descent. The SGD algorithm for finite-sum objectives, as assumed in FedAvg is of the following form,

as stated in McMahan et al. (2017) or Murphy (2023):

$$\min_{w \in \mathbb{R}^D} f(w) \qquad \text{where} \qquad f(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

In the frequentist setting, we typically consider $f_i(w)$ to be some loss function of a data point x_i potentially associated with some label y_i as in classification tasks, so we can define $f_i(w) := \ell(x_i, y_i; w)$. The objective $f(w)$ can then be considered as the average loss of our data with the parameters w. Stochastic gradient descent then minimizes this objective by differentiation of $f(w)$ with respect to w and updates this parameter iteratively according to some learning rate $\eta \in (0, 1]$ at time step t, Murphy (2023):

$$w_{t+1} \leftarrow w_t - \eta \nabla_w f(w)\big|_{w_t} \tag{2.9}$$

This can be interpreted as taking a step of size η in the direction of steepest descent at w_t, in order to avoid plateaus in the potentially non-convex loss landscape. In theory this learning rate should converge to 0 over time, but slowly enough to converge to the global optimum and not just a local minimum, Murphy (2023). In practice, however, this learning rate is usually fixed to some small enough value. For further details on traditional stochastic gradient descent, see Bishop (2006) or Murphy (2022, 2023).

Extending SGD to the so–called mini–batch setting, where we compute an update on only a subset of data, reveals the relationship with FL. Assume we have a subset of data points $\boldsymbol{x}_m$ of size $n_m = |\boldsymbol{x}_m|$, with indices $\mathcal{I}_m$, then the update becomes (Murphy, 2023)

$$w_{t+1} \leftarrow w_t - \eta \frac{1}{n_m} \sum_{i \in \mathcal{I}_m} \nabla_w f_i(w)\big|_{w_t}$$

McMahan et al. (2017) extends this notion of mini–batch updates, where we assume that there are M clients, each with their own data sets $\boldsymbol{x}_m$ for $m \in [M]$, corresponding sizes $n_m = |\boldsymbol{x}_m|$, and indexing sets $\mathcal{I}_m$, such that the global objective $f(w)$ becomes:

$$f(w) = \sum_{m=1}^{M} \frac{n_m}{n} F_m(w) \qquad \text{where} \qquad F_m(w) = \sum_{m \in \mathcal{I}_m} \frac{1}{n_m} f_i(w) \tag{2.10}$$

where n corresponds to the total number of data points across all clients. FedSGD makes each client, or potentially only a subset of clients, perform gradient computations locally, collecting these gradients at the server, aggregating them and take a step as per the specified learning rate as in equation 2.11 below. We denote a local gradient by $g_m = \nabla_w F_m(w)\big|_{w_t}$, and the parallel aggregation at the server is defined below.

$$w_{t+1} \leftarrow w_t - \eta \sum_{m=1}^{M} \frac{n_m}{n} g_m \tag{2.11}$$

This is the Federated Stochastic Gradient Descent algorithm. However, this also highlights that we potentially have a very high computation overhead, since it might take plenty

of updates at each client to converge globally. Considering the equivalent formulation of equation 2.11 can improve on this by doing more than one round of updating at each client locally, meaning we take several steps at each client and aggregate this at the server, McMahan et al. (2017).

$$w_{t+1}^m \leftarrow w_t - \eta g_m \,, \forall m \in [M] \qquad \text{then} \qquad w_{t+1} \leftarrow \sum_{m=1}^{M} \frac{n_m}{n} w_{t+1}^m \tag{2.12}$$

This shows that we can do several local steps, return this final update to the server and then aggregate the updates centrally. This builds the foundation of Federated Averaging. This algorithm still performs well even if only a subset of clients are available for computation at a time step t. This procedure is formalised in algorithm 3, and reduces to FedSGD if we only take a single step at each client update computation, McMahan et al. (2017). FedAvg assumes that the subset of clients S_t that are available for computation is either

Algorithm 3 Federated Averaging (McMahan et al., 2017)

Inputs: M clients indexed by m, local mini–batch size B, local update steps E, and learning rate η

Server:
- **Initialise:** w_0
- **for** each iteration $t = 1, 2, ...$ **do**
  - $S_t \leftarrow$ (set of indexes of available clients)
  - **for** each client $m \in S_t$ **do synchronously**
    - $w_{t+1}^m \,,\, n_m \leftarrow \text{ClientUpdate}(m, w_t, E, B, \eta)$
  - **end for**
  - $w_{t+1} \leftarrow \sum_{m=1}^{M} \frac{n_m}{n} w_{t+1}^m$

ClientUpdate(m, w, E, B, η)**:**
- let $n_m = |\boldsymbol{x}_m|$
- **for** each epoch i from 1 to E **do**
  - $\mathcal{B} \leftarrow$(sample mini–batch of size B from $\boldsymbol{x}_m$)
  - $w \leftarrow w - \eta \frac{n_m}{B} \sum_{b \in \mathcal{B}} \nabla_w f_b(w)$
- return w and n_m to server

all of them, or a fixed number of clients are randomly sampled at each iteration. We also need to carefully select the number of local update steps E, since too many steps could cause overfitting locally, and hence performs worse on heterogeneous data. This algorithm is quite simple, yet performs well in practice as shown in McMahan et al. (2017) for image classification using convolutional neural networks and language modelling, and is a frequent baseline in many frequentist federated learning papers such as Reddi et al. (2021). However, FedAvg and frequentist approaches in general are bad for uncertainty quantification and detecting one–off events, as they are overconfident in their predictions, however this works well for classification tasks with no overlaps in labels, however less well for complex next–character in language prediction, McMahan et al. (2017). Furthermore, FedAvg significantly outperforms FedSGD on almost all experiments in McMahan et al. (2017) with identical

learning rates and local batch sizes. Recent work on theoretical guarantees of Federated Averaging in the training of Neural Networks has been studied in Song et al. (2023), and references therein. However, the significant advantage of FedAvg over FedSGD empirically remains unclear in theory, Kairouz et al. (2021).

Adaptive Federated Optimization (FedOpt), as developed in Reddi et al. (2021), generalizes FedAvg to other optimisation schemes, such as Adam (Kingma and Ba, 2015), Adagrad, and Yogi, for situations where SGD does not perform well and does not converge, see further references in Reddi et al. (2021) for details on these optimisation procedures. They also extend this framework to consider global optimization steps at the server when combining the gradients. FedOpt, as shown in algorithm 4, can be shown to significantly outperform FedAvg on all considered experiments with the choice of the right optimisation method. Furthermore, they also consider the mathematical assumptions that FedAvg makes and provide some mathematical convergence analysis (however no convergence guarantees in general) of their proposed algorithms. We note that FedOpt is a general procedure, where client optimisation procedures and server optimisation procedures can be freely chosen, however we only show this for local SGD at clients as in FedAvg and for the Adagrad, Yogi, and Adam optimisers at the server in algorithm 4, improvements of these FedOpt and the FedAvg algorithms to heterogeneity concerns are considered in particular in the next section.

Algorithm 4 Federated Optimization with FedAdagrad, FedYogi and FedAdam (Reddi et al., 2021)

Input: w_0, $v_{-1} \geq \tau^2$, decay parameters $\beta_1, \beta_2 \in [0, 1)$, local learning rate η_l, server learning rate η and number of local epochs E

for iterations $t = 0, 1, ..., T-1$ **do**
  Sample a subset $\mathcal{S}_t$ of clients
  for each client $m \in \mathcal{S}_t$ **in parallel do**
    $w_{m,0}^t = w_t$
    for epoch $i = 0, ..., E-1$ **do**
      Compute unbiased estimate $g_{m,i}^t$ of $\nabla_w F_m(w_{m,i}^t)$
      $w_{m,i+1}^t = w_{m,i}^t - \eta_l g_{m,i}^t$
    end for
    $\Delta_m^t = w_{i,E}^t - w_t$
  end for
  if t=0 **then**
    $\Delta_{-1} = \sum_{m \in \mathcal{S}_t} \Delta_m^t$
  end if
  $\Delta_t = \beta_1 \Delta_{t-1} + (1 - \beta_1)\left(\sum_{m \in \mathcal{S}_t} \Delta_m^t\right)$
  $v_t = v_{t-1} + \Delta_t^2$ (FedAdagrad)
  $v_t = v_{t-1} - (1 - \beta_2)\Delta_t^2 \operatorname{sign}(v_{t-1} - \Delta_t^2)$ (FedYogi)
  $v_t = \beta_2 v_{t-1} + (1 - \beta_2)\Delta_t^2$ (FedAdam)
  $w_{t+1} = w_t + \eta \frac{\Delta_t}{\sqrt{v_t} + \tau}$
end for
Return: w_T

Federated Boosting (FedBoost), Hamer et al. (2020), further develops a communication efficient algorithm using ensembles of pre–trained models. Malinovsky et al. (2020) develop fixed point methods for frequentist federated learning in order to improve communication overhead as well, however suffering in accuracy as a result. Since we are mainly concerned with probabilistic federated learning, we do not cover these algorithms in detail here.

2.2.3 Revisiting Challenges in Federated Learning

Zhu et al. (2019) shows that the gradients that we communicate to the server, for instance in FedSGD, can be used to recover local data from clients. This raises privacy concerns in federated learning settings, since it was motivated in McMahan et al. (2017), and developed, as a secure and private way of learning a model without sharing data with a centralised server. However, the work of Zhu et al. (2019), illustrating potential data leakage from gradients, shows that exactly this set–up can be used to recover client data under malicious actors, including both third party actors and the central server. To this end we can consider differential privacy in order to obscure gradients in communication and at the server, since it aims to introduce artificial uncertainty about a clients data, Kairouz et al. (2021). Furthermore, secure aggregation strategies that ensure that the central server cannot observe individual contributions to model updates and only the entirety from all clients taken together can protect against malicious servers, Chen et al. (2022) and Kairouz et al. (2021). Further work on data leakage in federated learning is necessary in order to ensure privacy of client data against adversaries, such as the work of Zhao et al. (2023) in which there is data leakage when transmitting model weights, as in FedAvg or FedOpt. They directly implement their technique of data leakage from models on FedAvg and demonstrate the potential privacy concerns in FedAvg. However, privacy concerns, while integral to federated learning, are tangential to the current thesis and we refer the reader to Chen et al. (2022) for work on secure aggregation and differential privacy, and to Zhu et al. (2019) and Zhao et al. (2023) for work on data leakage from gradients and model respectively, and references therein.

Recent work on heterogeneity has also aimed at improving federated learning. Tenison et al. (2023) introduce a drop–in approach termed Gradient Masked Averaging (GMA) that aims to deal with data heterogeneity, such as out–of–distribution data, where local clients contain data that has different distributions,seen in many practical approaches. This is done through the mask m_τ that checks if gradients are "inconsistent in sign across [clients]" (Tenison et al., 2023). This means that if the sign of gradient components varies too much, as specified by some hyper–parameter $\tau \in [0, 1]$, then that part of the gradient is set to 0 and doesn't affect the optimisation step. This mask is calculated for a set of clients $\mathcal{S}$, the D–dimensional gradient vectors $\nabla F_m(w)$, and M, the total number of clients in the entire network, through the following indicator function:

$$[m_\tau]_{j=1}^{D} = \mathbb{1}\Big\{ \frac{1}{M} \Big| \sum_{m \in \mathcal{S}} \operatorname{sign}\big(\big[\nabla F_m(w)\big]_j\big) \Big| \geq \tau \Big\}$$

However, since we do not have direct access to gradients most of the time, we instead apply this mask to the change in the model parameters given by Δ_m^t for client m at time t, Tenison et al. (2023):

$$\left[\tilde{m}_\tau(\{\Delta_m^t\}_{m\in\mathcal{S}})\right]_{j=1}^{D} = \mathbb{1}\Big\{\frac{1}{M}\Big|\sum_{m\in\mathcal{S}}\operatorname{sign}(\left[\Delta_m^t\right]_j)\Big| \geq \tau\Big\} \tag{2.13}$$

These equations reveal, that hyper–parameter τ influences how many clients we need in order to make an update, since we average the signs over the total number of clients M. Tenison et al. (2023) directly apply GMA to FedAvg and FedOpt using their method, as shown in algorithm 5. Note that the symbol $\odot$ in algorithm 5 indicates the piecewise multiplication of vectors, i.e. we do not change the current estimate at some position in the parameter vector, if the clients are insufficient in number or are not in sufficient agreement. They show that GMA significantly outperforms the FedAvg and FedOpt counterparts without GMA. This improvement is especially prevalent on non–i.i.d. data, and tends to work well in experiments where only a subset of clients are available for computation at a time. However, using this approach to federated learning can lead to slower convergence of the global model, Tenison et al. (2023).

Furthermore, the work of Tziotis et al. (2023) has aimed at improving system heterogeneity challenges in federated learning, especially for robustness to straggling clients when learning representations.

2.2.4 Bayesian Approaches

Frequentist statistics are typically overconfident in their results,Kassab and Simeone (2022), and have therefore bad uncertainty quantification, even though confidence intervals can somewhat help mitigate this, (Murphy, 2023). In the frequentist sense, we treat the parameter θ that we want to infer as fixed but unknown and the data as random variables, however since we typically know the data, we can also do the opposite and treat the data as fixed and known and the parameter θ as a random variable, Murphy (2023). The latter is known as Bayesian approaches, also known as approximate inference techniques (Felekis et al., 2022), which we will now, and for the remainder of this thesis be focusing on.

We consider the training of a global posterior distribution over a set of M individual clients, with data sets $\boldsymbol{x}_m$ respectively. Furthermore, we assume that these clients have some likelihood function $p_m(\boldsymbol{x}_m|\cdot)$ and prior distribution $\pi(\cdot)$ over some common parameter $\theta \in \Theta$. We have already established that we cannot simply calculate an exact Bayesian posterior $q_B^* \in \mathcal{P}(\Theta)$, therefore we need approximation strategies. In probabilistic machine learning the two main approaches are deterministic, such as VI, or stochastic, such as Markov Chain Monte Carlo, in nature.

An intuitive way of performing distributed computation of a posterior distribution is the Bayesian Committee Machine (BCM) of Tresp (2000), in which each client locally infers a posterior distribution, which are aggregated at the server according to the amount of data

Algorithm 5 Gradient Masked Averaging (Tenison et al., 2023) for FedAvg, FedAdam, FedYogi (FedAdam and FedYogi)

Inputs: w_0, $v_0 \geq e^{-6}$, hyper–parameter τ, decay parameters $\beta_1, \beta_2 \in [0, 1)$, local learning rate η_l, mini–batch size B and server learning rate η

Server:

for each iteration $t = 1, 2, ..., T$ **do**

Sample a subset $\mathcal{S}_t$ of clients randomly

for each client $m \in \mathcal{S}_t$ **do in parallel**

$w_t^m = \text{ClientUpdate}(w_{t-1}, m, \eta_l, E)$

$\Delta_t^m = \frac{n_m}{\sum_{k \in \mathcal{S}_t} n_k}(w_t^m - w_{t-1})$

end for

$\Delta_t = \sum_{m \in \mathcal{S}_t} \Delta_t^m$

$m_t = \tilde{m}_\tau(\{\Delta_t^m\}_{m \in \mathcal{S}_t})$ (as in equation 2.13)

if $t = 1$ **then** $z_0 = \Delta_1$ (FedAdam and FedYogi)

$z_t = \beta_1 z_{t-1} + (1 - \beta_1)\Delta_t$ (FedAdam and FedYogi)

$v_t = v_{t-1} - (1 - \beta_2)\Delta_t^2 \text{sign}(v_{t-1} - \Delta_t^2)$ (FedYogi)

$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\Delta_t^2$ (FedAdam)

$\Delta_t = \frac{z_t}{\sqrt{v_t + e^{-3}}}$ (FedAdam and FedYogi)

$w_t = w_{t-1} + \eta * m_t \odot \Delta_t$

end for

Return: w_T

ClientUpdate(w, m, η_l, E)**:**

Let $w_0 = w$

for each local epoch $i = 0, 1, ..., E - 1$ **do**

Sample mini–batch $\mathcal{B}$ of size B uniformly at random from the client data set

$g_i = \frac{n_m}{B} \sum_{b \in \mathcal{B}} \nabla_w f_b(w_i)$

$w_{i+1} = w_i - \eta_l g_i$

end for

Return: w_E to the server

points of each client.

$$q_B^*(\theta|\boldsymbol{x}_{1:M}) \propto \prod_{m=1}^{M} \pi(\theta)^{n_m/n} p_m(\boldsymbol{x}_m|\theta) \approx \prod_{m=1}^{M} q_m(\theta)$$

This means, we compute a local posterior approximation $q_m(\theta) \in \mathcal{P}(\Theta)$ for each client based on the local likelihood and fraction of the prior. This updating can be done both in a VI manner, as well as using an MCMC approach which then becomes the Consensus Monte Carlo algorithm of Scott et al. (2016), where we sample from the likelihood function to infer a local posterior approximation.

However, this type of algorithm performs worse if the individual client data distributions vary, since the consensus MC approach limits itself to Gaussian distributions, Scott et al. (2016), and hence combining posteriors can have adverse effects, such as combining two Gaussians and finding as it's mean a distribution that does not cover any data.

The work on embarrassingly parallel MCMC of, for instance Mesquita et al. (2020), aims to improve on consensus MC by changing the aggregation strategy at the server by using importance sampling, Murphy (2023), when combining the approximations, and tries to mitigate the issue of heterogenous data.

Another interesting take on the consensus MC approach is the Federated Posterior Averaging (FedPA) algorithm of Al-Shedivat et al. (2021), who combine this with FedOpt through changing the SGD steps locally with an MCMC approach, and combine the client distribution parameters at the server according to some optimisation scheme. However, by doing this they limit themselves, not only to Gaussian approximations, for instance through Laplace approximations, but also to uniform prior distributions.

Further work on MCMC approaches to distributed learning include Ahn et al. (2014) who introduce Distributed Stochastic Gradient Langevin Dynamics (DSGLD), which takes a mini–batch approach traditional stochastic gradient Langevin dynamic, Murphy (2023, Chapter 12.7.1). This is similar in spirit to the SGD algorithm where we consider the local loss function to be the logarithm of the joint distribution of data and parameters, $p_m(\boldsymbol{x}_m|\theta)\pi(\theta)$, and add Gaussian noise to the updates in an MCMC fashion. The Markov Chains are then constructed by the last parameter of a chain as found by a client, which is then passed as the initial parameter chain to a random client to further build it based on that data set. This builds M Markov Chains in parallel, which can then be combined globally after a fixed number of iterations. However, it assumes data is distributed evenly among a small number of clients, ideally in a data centre.

Mekkaoui et al. (2021) extend on this by introducing the Federated Stochastic Gradient Langevin Dynamics (FSGLD) which aims to deal with federated learning challenges and provide mathematical rigour for DSGLD. The FSGLD algorithm not only computes local Markov Chains by the likelihood function, but also through "conducive gradients" (Mekkaoui et al., 2021) that model the data likelihood of different clients similarly to EP and multiply together to derive an approximation for the entire data set distribution.

However, MCMC based algorithms are computation heavy due to having to sample a large number of data points, and hence are time inefficient which makes their adaption to federated learning challenging in some practical situations where clients are available only a limited amount of time or have limited computation and storage allowances. Deterministic approximate inference methods can be significantly faster than stochastic versions, however trading some accuracy as a result.

Expectation Propagation as described in Vehtari et al. (2020) can also be seen as a distributed algorithm to deterministic approximate inference. As Vehtari et al. (2020) shows, we can allow clients to have more than a single data point at a client, however they use MCMC sampling to infer the local probability distributions that are send back to the client. (Hasenclever et al., 2017) expanded on a preprint of Vehtari et al. (2020) and proposed a convergent double–loop algorithm for power expectation propagation—power EP replaces the reverse Kullback Leibler divergence with Amari's alpha–Divergence (Amari, 2016), Minka (2004, 2005)—based on the exponential family and natural gradient descent (Amari, 2016).

Furthermore, (Guo et al., 2023) proposed Federated Expectation Propagation (FedEP) to make the work of Vehtari et al. (2020) scalable through a variation of Stochastic VI (Hoffman et al., 2013), and improves upon FedPA (Al-Shedivat et al., 2021) by taking approximate likelihood into account.

The Partitioned Variational Inference (PVI) framework of Bui et al. (2018), on which we build this thesis, is discussed in chapter 3 and is another deterministic approximate inference approach to federated learning. Distributed Stein Variational Gradient Descent (DSVGD) of Kassab and Simeone (2022) builds on PVI by proposing a particle algorithm for distributed VI using using arbitrary loss functions in the posterior and is not limited to log likelihoods, as in the generalised Bayesian posteriors of Bissiri et al. (2016) and Miller (2021).

In chapter 4 we propose Partitioned Generalised Variational Inference, a novel robust federated learning algorithm that extends PVI to use arbitrary divergence measures and loss functions, inspired by the global Generalised Variational Inference algorithm of Knoblauch et al. (2022).

2.3 Information Geometry

> *"Information geometry is a method of exploring the world of information by means of modern geometry."* — Amari (2016)

Information Geometry allows the study of closeness of probability distributions through divergence measures, such as the Kullback–Leibler divergence, through exploring statistical manifolds of probability distributions; these are characterised by their coordinates $\theta \in \Theta$, Nielsen (2020). We can therefore view $\mathcal{P}(\Theta)$ as a statistical manifold of probability distributions over a local coordinate chart Θ, such that each probability distribution is uniquely characterised by it's parameters. Therefore, we can characterise the discrepancy between any two points on the manifold through divergences.

Furthermore, since such manifolds are not Euclidean, but rather only dually flat under some Riemannian metric, optimisation methods that are proposed on Euclidean spaces are not necessarily optimal, Amari (2016). Optimisation methods such as Natural Gradient Descent or Bregman mirror descent can therefore overcome challenges in traditional Gradient Descent based schemes, Amari (2016). For instance, Amari (2016) shows that Natural Gradient Descent has no plateaus on which optimisation stalls when using vanilla SGD. We are particularly concerned with studying divergence measures, which are used for our novel Partitioned Generalised Variational Inference approach.

See Amari (2016) for an introduction to information geometry or Nielsen (2020) for a concise survey of differential geometry in information manifolds.

2.3.1 Divergence Measures

Divergence measures have been extensively studied in the literature and have many practical applications in information theory, e.g. Pardo Llorente (2006).

Definition 10 (Divergence) *A divergence $D : \mathcal{P}(\Theta) \times \mathcal{P}(\Theta) \to \mathbb{R}_{\geq 0}$ between p and q in the manifold $\mathcal{P}(\Theta)$ of probability distributions with local coordinates θ_p and θ_q satisfies*

1. $D(p||q) \geq 0$,
2. $D(p||q) = 0 \iff p(\theta) = q(\theta)$.

Technically, to be a divergence, this is not sufficient, and for close coordinates it also needs to satisfy certain conditions on the partial derivatives with respect to the coordinates, see Nielsen (2020) for details. A divergence D has a dual D^* such that $(D^*)^* = D$ and $D^*(p||q) = D(q||p)$, since divergences are not symmetric in general, $D(q||p) \neq D(p||q)$, they are not necessarily self–dual.

The most well known family of divergences are the f–divergences, which are given by a differentiable and convex function f such that $f(1) = 0$, Nielsen and Okamura (2024).

$$D_f(p||q) = \int p\, f\Big(\frac{q}{p}\Big)$$

From this it can be easily seen that the KL divergence is an f–divergence, where $f(u) = -\log u$, with its dual being $f^*(u) = u \log u$, leading to the reverse KL divergence, Amari (2016). Because of this, the KL divergence in VI is mode seeking (zero forcing) and the reverse KL divergence is mode avoiding (zero avoiding) (Minka, 2005). For $\mathrm{KL}(q||p)$, it is mode seeking ensures that $q(\theta) = 0$ whenever $p(\theta) = 0$; for the reverse $\mathrm{KL}(p||q)$, being mode avoiding ensures that $q(\theta) \neq 0$ whenever $p(\theta) \neq 0$, Amari (2016). Hence vanilla VI over–concentrates on a single mode of a distribution and vanilla EP tries to cover all modes.

Further work on divergence measures includes the $\alpha\beta\gamma$–divergences (Cichocki and Amari, 2010), the game–theoretic Hyvärinen divergence (Amari, 2016) which is half of the Fisher divergence (Nielsen, 2021), and the total variation distance (Jewson et al., 2018) which is used for convergence proofs.

2.4 Discrepancy Variational Inference

We typically use the KL divergence in VI since this is motivated through the ELBO, but not through the divergence minimisation view of VI. Having introduced a wealth of different divergences, we could therefore use other divergence measures instead to approximate the Bayesian posterior. We follow the formulation of Knoblauch et al. (2022) and call this discrepancy VI (DVI).

Definition 11 (Discrepancy Variational Inference) *The DVI posterior produced by some divergence $D \neq \mathrm{KL}$ is*

$$q^*_{DVI}(\theta) = \underset{q(\theta)\in\mathcal{Q}}{\arg\min}\, D\big(q(\theta)\big|\big|q^*_B(\theta|x_{1:n})\big)$$

This includes EP which is trying to minimise the reverse KL divergence, Rènyi divergence VI (Li and Turner, 2016), β–divergence VI (Futami et al., 2018), and other divergence measures

(Jewson et al., 2018). These methods can outperform vanilla VI empirically, however are suboptimal with respect to the Bayesian posterior of Zellner (1988), Knoblauch et al. (2022).

This empirical outperformance of some of these methods is a result of DVI posteriors encoding desirable properties, such as robustness to outliers (Futami et al., 2018), in their objectives. In general, we have that the DVI posteriors do not equal the usual VI posterior, however if a decision maker (DM) decides that she does not want to approximate the Bayesian posterior, which is defined through the KL divergence by Zellner (1988), but is rather interested in modelling out noise or outliers, then a DVI posterior can be more desirable than traditional VI. This relates to information geometry, since VI can have undesirable properties, such as being mode seeking or underestimating the variance of the target distribution, i.e. being over–confident. Therefore, choosing a different divergence can result in better uncertainty quantification. This will be useful in chapter 4, where we move away from the KL divergence.

2.5 Model Misspecification

Regardless of which divergence we choose, these rely on having a well specified model by virtue of including the Bayesian posterior q_B^*. This means that the parameter that we target and want to converge to, and which generated the data, θ^*, is contained within the parameter space Θ. This is known as the $\mathcal{M}$–closed assumption, Bernardo and Smith (2000). And hence, the likelihood function needs to be correctly specified. However, in practice this is not the case since we do not know the underlying data generating process (DGP), nor does it need to exist in any usable form; this is the $\mathcal{M}$–open assumption, Bernardo and Smith (2000). Figure 2.6 is inspired by Knoblauch et al. (2022), and illustrates that the data generating process $F^*(\cdot)$ does not lie on the statistical manifold of probability distributions parametrised by $\theta \in \Theta$.

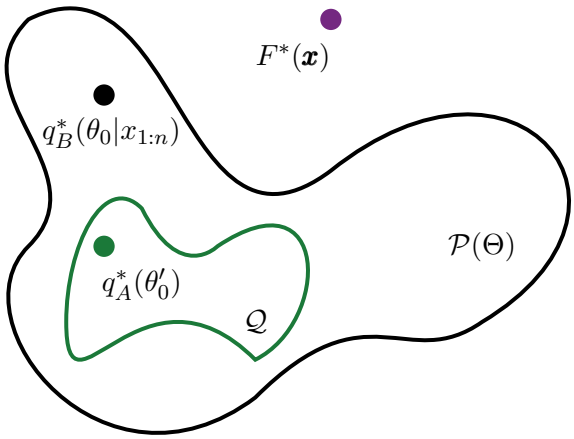


Figure 2.6: Best viewed in colour. Visualisation of model misspecification in the $\mathcal{M}$–open framework. The statistical manifold $\mathcal{P}(\Theta)$ represents all probability distributions parametrised by $\theta \in \Theta$. The data generating process $F^*(\cdot)$ does not lie on this manifold.

Acknowledging that the likelihood is likely misspecified, and that there exists no $\theta^* \in \Theta$ under which the data could be generated, requires thought about what we want to approximate. Walker (2013) argues that we want to target a value $\theta_0 \in \Theta$ which minimises the Kullback–Leibler divergence between our likelihood function and the true data generating process, $F^*(\boldsymbol{x})$.

$$\theta_0 = \arg\min_{\theta\in\Theta} \mathrm{KL}(F^*(\boldsymbol{x})||p_n(\boldsymbol{x}|\theta)) = \arg\min_{\theta\in\Theta} -\int_{\mathcal{X}} \log p_n(\boldsymbol{x}|\theta)\, dF^*(\boldsymbol{x})$$

We do not know this true DGP, however we can show that even under misspecification, the Bayesian posterior converges around θ_0. This optimal parameter is illustrated through the posteriors in figure 2.6, where we either consider the Bayesian posterior q_B^* or an alternative posterior within a subset $\mathcal{Q} \subset \mathcal{P}(\Theta)$, q_A^*.

2.5.1 Posterior Convergence under Model Misspecification

The work of Berk (1966) was instrumental in exploring posterior convergence under misspecification. It is the first work that examines concentration around some $\theta_0 \in \Theta$, not necessarily unique, under model misspecification, however under relatively strict assumptions such as i.i.d. data and Θ being a complete separable metric space. Furthermore, the work of Kleijn and van der Vaart (2006) extends this by considering more general Θ and by considering infinite dimensional models, such as Gaussian processes (Rasmussen and Williams, 2006). Shalizi (2009) further generalises the work by considering non-i.i.d. data, e.g. of stationary ergodic processes, and giving a proof based on a version of the Shannon-McMillan-Breiman theorem (Cover and Thomas, 2006). We consider here the proof of Miller (2021), which is not necessarily as strong as the one of Shalizi (2009), however makes fewer assumptions and is simpler to follow, as it builds on fundamental measure theory.

Theorem 12 *Let Π_n be a posterior after observing n data points $\{x_i\}_{i=1}^n$, and let Π be a prior distribution over the parameter space Θ. Further let $p_n(\cdot|\theta)$ be a likelihood function of some data, all taken with respect to some dominating measure. Then for a set $A \subset \Theta$*

$$\Pi_n(A) = \textstyle\int_A p_n(x_{1:n}|\theta)d\Pi(\theta) / \int_\Theta p_n(x_{1:n}|\theta)d\Pi(\theta)$$

assuming that $\int_\Theta p_n(x_{1:n}|\theta)d\Pi(\theta) < \infty$, $\forall n$. We define

$$f_n(\theta) := -\frac{1}{n}\log p_n(x_{1:n}|\theta)$$
$$f(\theta) := \lim_{n\to\infty} \mathbb{E}_{F^*}[f_n(\theta)] = \lim_{n\to\infty} -\frac{1}{n}\int_{\mathcal{X}} \log p_n(x_{1:n}|\theta)\, dF^*(x_{1:n})$$

Then for some $\theta_0 = \inf_{\theta\in\Theta} f(\theta)$ such that

$$\liminf_n \inf_{\theta\in A_\epsilon^c} f_n(\theta) > f(\theta_0)$$

where $A_\epsilon = \{\theta \in \Theta : f_n(\theta) < f(\theta) + \epsilon\}$ then, assuming that $\Pi(A_\epsilon) > 0$ for all $\epsilon > 0$, we have $\Pi_n(A_\epsilon) = 1$, F^*–almost surely.

Proof See Miller (2021) or appendix B, where we show that the posterior over the complement of A_ϵ, i.e. $\Pi_n(A_\epsilon^c)$, converges to 0 as n tends to infinity. ■

The result is surprisingly simple to proof, however it does not extend to rates of convergence as the work of Shalizi (2009) does.

Chapter 3

Partitioned Variational Inference

In this chapter, we discuss Partitioned Variational Inference (PVI), which combines federated learning with approximate Bayesian inference. PVI is a probabilistic federated learning algorithm, that takes ideas from Variational Inference and Expectation Propagation to develop a distributed variational algorithm. This idea was originally presented in Hensman et al. (2014) as a global VI approach using EP, termed Tilted Variational Bayes, and we belief independently re–invented in Bui et al. (2018) as a federated learning approach. Since our work focuses on Federated Learning, we mainly focus on Bui et al. (2018)–as well as Ashman et al. (2022) in which the authors of Bui et al. (2018) further develop PVI–since they explore this under the lens of distributed computation, in contrast to Hensman et al. (2014) whose aim is to simplify computation in a global model.

Chapter 3.1 explores PVI where we reframe the Evidence Lower Bound of Variational Inference as an optimisation problem over a partition of the entire data set. We introduce the algorithm and link VI to Expectation Propagation.

Chapter 3.2 investigates different server scheduling schemes in PVI, in order to address client and data heterogeneity in federated learning with PVI. We explore sequential, synchronous (parallel) and asynchronous (whenever clients are available) updating.

Chapter 3.3 PVI was developed independently of Tilted Variational Bayes, as far as we are aware, therefore we investigate the similarities and differences in these two approaches.

Chapter 3.4 Federated learning can have challenging problem domains, we review how challenges of federated learning can be addressed by PVI and where PVI falls short.

Chapter 3.5 shows that PVI encompasses variational continual learning by the nature of the initialisation of the approximate likelihood terms.

Chapter 3.6 Kullback–Leibler divergence minimisation using the ELBO is a non–convex objective, and hence requires optimisation procedures. We explore Adam as

a way of minimising the objective of PVI, and how we can approximate intractable expectations using Monte Carlo VI, Murphy (2023).

Chapter 3.7 We implement PVI from the definition and explore the effectiveness and drawbacks of PVI on toy data sets of mixtures of multivariate Gaussian distributions, McLachlan and Peel (2000); we compare these experiments with the EP framework.

3.1 Derivation of Partitioned Variational Inference

Partitioned Variational Inference, as in Bui et al. (2018) and Ashman et al. (2022), assumes that there are M clients, each with their unique data sets $\boldsymbol{x}_m$ for $m \in [M]$, and associated likelihood function $p_m(\boldsymbol{x}_m|\theta)$. These likelihood functions do not have to be common across clients, i.e. clients can have different data distributions, and the entire data doesn't need to be homogenously split across clients. We then assign each client an approximate likelihood term $t_m(\theta)$ which does not depend on the data explicitly. We further assume that the model has some global prior belief $\pi(\theta)$ about the distribution of the entire data set. This set–up is visualised in figure 3.1 for an arbitrary amount of clients. The Bayesian posterior $q_B^*(\theta|\boldsymbol{x}_{1:M})$ of these likelihood can be computed through the usual Bayes' theorem where we assume that the client data sets are independent from one another, Bui et al. (2018). Similar to Expectation Propagation, we assume that our approximate posterior approximation $q_{PVI}^*(\theta)$ factorises over the approximate likelihoods of the clients and the prior.

$$q_B^*(\theta|\boldsymbol{x}_{1:M}) = \frac{1}{Z}\pi(\theta)\prod_{m=1}^{M} p_m(\boldsymbol{x}_m|\theta) \approx \frac{1}{Z_q}\pi(\theta)\prod_{m=1}^{M} t_m(\theta) = q_{PVI}^*(\theta) \tag{3.1}$$

Z and Z_q represent the normalising constants of the Bayesian posterior and the approximate posterior respectively. In difference to EP we do not minimize the reverse KL divergence, but rather the one from VI, such that our aim is the minimisation of the following objective.

$$q_{PVI}^*(\theta) = \underset{q(\theta)\in\mathcal{Q}}{\arg\min}\,\mathrm{KL}\big(q(\theta)||q_B^*(\theta|\boldsymbol{x}_{1:M})\big) \tag{3.2}$$

To derive our local update steps, we consider the KL divergence in the global minimisation step, as in equation 3.2, and derive a sequence of local optimisation steps that are equivalent.

$$\mathrm{KL}\big(q(\theta)||q_B^*(\theta|\boldsymbol{x}_{1:M})\big) = \mathbb{E}_{q(\theta)}\big[\log q(\theta)\big] - \mathbb{E}_{q(\theta)}\big[\log \pi(\theta)\prod_{m=1}^{M} p_m(\boldsymbol{x}_m|\theta)\big] + \mathbb{E}_{q(\theta)}[\log Z]$$

Through rearranging this equation and by applying the the definition of $q_{PVI}(\theta)$, we derive:

$$\begin{aligned}
&= \mathbb{E}_{q(\theta)}\big[\log q(\theta)\big] - \mathbb{E}_{q(\theta)}\big[\log \pi(\theta)\big] - \mathbb{E}_{q(\theta)}\big[\sum_{m=1}^{M}\log p_m(\boldsymbol{x}_m|\theta)\big] + \log Z \\
&= \int_\Theta q(\theta)\log\frac{\pi(\theta)\prod_{m=1}^{M} t_m(\theta)}{Z_q\,\pi(\theta)}\,d\mu(\theta) - \sum_{m=1}^{M}\mathbb{E}_{q(\theta)}\big[\log p_m(\boldsymbol{x}_m|\theta)\big] + \log Z
\end{aligned}$$

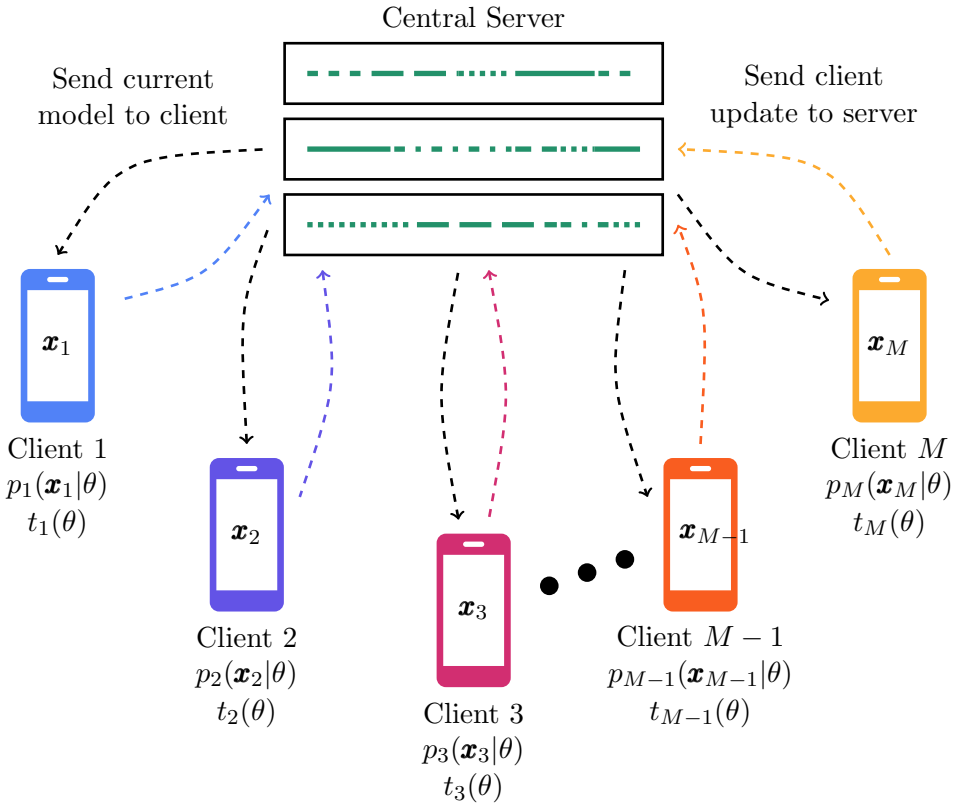


Figure 3.1: Graphical visualisation of the **Partitioned Variational Inference** set–up of M clients scheduled by a central server. Dashed arrows indicate model updates being communicated. Each client has a local likelihood of it's own data $p_m(\boldsymbol{x}_m|\theta)$, and an approximate likelihood term $t_m(\theta)$ assigned to it, which are aggregated at the server as shown.

$$
\begin{aligned}
&= \int_\Theta q(\theta) \log \prod_{m=1}^{M} t_m(\theta)\, d\mu(\theta) - \sum_{m=1}^{M} \int_\Theta q(\theta) \log p_m(\boldsymbol{x}_m|\theta)\, d\mu(\theta) + \log \frac{Z}{Z_q} \\
&= \sum_{m=1}^{M} \int_\Theta q(\theta) \log \frac{t_m(\theta)}{p_m(\boldsymbol{x}_m|\theta)}\, d\mu(\theta) + \log \frac{Z}{Z_q}
\end{aligned}
$$

By multiplying with $1 = q(\theta)/q(\theta)$ inside the logarithm, and noting that the cavity distribution from EP, Vehtari et al. (2020), is defined as $q^{\setminus m}(\theta) = \frac{q(\theta)}{t_m(\theta)} = \pi(\theta) \prod_{i \neq m} t_i(\theta)$, we arrive at our update steps for PVI, Ashman et al. (2022):

$$
\begin{aligned}
\mathrm{KL}\big(q(\theta)||q_B^*(\theta|\boldsymbol{x}_{1:M})\big) &= \sum_{m=1}^{M} \int_\Theta q(\theta) \log \frac{q(\theta)\, t_m(\theta)}{q(\theta)\, p_m(\boldsymbol{x}_m|\theta)}\, d\mu(\theta) + \log \frac{Z}{Z_q} \\
&= \sum_{m=1}^{M} \int_\Theta q(\theta) \log \frac{q(\theta)}{q^{\setminus m}(\theta)\, p_m(\boldsymbol{x}_m|\theta)}\, d\mu(\theta) + \log \frac{Z}{Z_q} \\
&= \sum_{m=1}^{M} \mathrm{KL}\big(q(\theta)\big|\big|q^{\setminus m}(\theta)\, p_m(\boldsymbol{x}_m|\theta)\big) + \log \frac{Z}{Z_q}
\end{aligned}
$$

And since we are minimizing this term we can safely ignore the constant Z, since it does not depend on our choice of $q(\theta)$, as in traditional VI (Blei et al., 2016), however Z_q does

depend on this choice.

$$\arg\min_{q(\theta)\in\mathcal{Q}} \mathrm{KL}\big(q(\theta)||q_B^*(\theta|\boldsymbol{x}_{1:M})\big) = \arg\min_{q(\theta)\in\mathcal{Q}}\Big\{\sum_{m=1}^{M}\mathrm{KL}\big(q(\theta)\big|\big|q^{\setminus m}(\theta)\,p_m(\boldsymbol{x}_m|\theta)\big) - \log Z_q\Big\} \quad (3.3)$$

The normalisation constant Z_q is required at the server in order to compute the new posterior for synchronous updating schemes, Ashman et al. (2022). Z_q does not decompose across clients, and therefore PVI aims to minimise equation 3.3 without the logarithm in a distributed fashion. We formalise this idea in algorithms 6 and 7.

Algorithm 6 Partitioned Variational Inference, Server (Bui et al., 2018)

Server:

Inputs: M clients, prior distribution $\pi(\theta)$, and variational family $\mathcal{Q}$

Initialise: $t_m^{(0)}(\theta) = 1,\ \forall m \in [M] \quad \implies \quad q^{(0)}(\theta) = \pi(\theta)$

until convergence for iteration $i = 1, 2, ...$ **do**

For client m in update schedule $\mathcal{B}_i$ **do in parallel**

$\Delta_m^{(i)}(\theta) \leftarrow \text{ClientUpdate}(q^{(i-1)}(\theta), \mathcal{Q})$ (# Algorithm 7)

end for

#Aggregate Client updates:

$$q^{(i)}(\theta) \propto q^{(i-1)}(\theta)\prod_{m\in\mathcal{B}_i}\Delta_m^{(i)}(\theta)$$

Return: $q(\theta)$

Algorithm 7 Partitioned Variational Inference, Client Update (Bui et al., 2018)

ClientUpdate$(q^{(i-1)}(\theta), \mathcal{Q})$**:**

Inputs: Current approximation $q^{(i-1)}(\theta)$ and variational family $\mathcal{Q}$

#Compute Cavity Distribution:

$$q^{\setminus m}(\theta) \propto \frac{q^{(i-1)}(\theta)}{t_m^{(i-1)}(\theta)}$$

#Compute local approximation:

$$q_m^{(i)}(\theta) \leftarrow \arg\min_{q(\theta)\in\mathcal{Q}}\Big\{\mathbb{E}_{q(\theta)}\big[-\log p_m(\boldsymbol{x}_m|\theta)\big] + \mathrm{KL}\big(q(\theta)\big|\big|q^{\setminus m}(\theta)\big)\Big\} \quad (3.4)$$

#Update local approximate likelihood:

$$t_m^{(i)}(\theta) \propto \frac{q_m^{(i)}(\theta)}{q^{\setminus m}(\theta)}$$

Return: $\Delta_m^{(i)}(\theta) := \frac{t_m^{(i)}(\theta)}{t_m^{(i-1)}(\theta)}$ to server

After initialising the approximate likelihoods to either be 1 or encode some sort of prior belief about the client distributions, then the initial approximate posterior is equal to our prior distribution. The server sends the current approximate posterior to a set of clients chosen according to some updating schedule, where this is either performed sequentially, i.e. one client at a time and in some order over all clients, synchronously, i.e. all clients update in parallel, or asynchronously, i.e. only available clients perform computations. Details on these schedules are discussed in the next section.

Proposition 13 *If all clients converge to a distribution $q^*_{PVI}(\theta)$ such that $\Delta^*_m(\theta) = 1$ for all subsequent client updates and $\forall m \in [M]$, then*

$$q^*_{PVI}(\theta) = \arg\min_{q(\theta)\in\mathcal{Q}} \mathrm{KL}\big(q(\theta)\big|\big|q^*_B(\theta|\boldsymbol{x}_{1:M})\big)$$

Proof This proposition is equivalent to Bui et al. (2018, Property 3) if $Z_q = 1$ and Ashman et al. (2022, Property 2.3) otherwise, and is proved there. It relies on computing the gradient vector and the Hessian matrix of second derivatives of the global KL divergence and showing this is equivalent to the sum of the gradients of the local KL divergences. Since the local KL divergences are minimal at some optimal solution on which all clients agree, the gradient is 0 (Bui et al., 2018) and therefore $q^*_{PVI}(\theta)$ minimizes equation 3.2. ■

If $\forall m \in [M]$, $\Delta^*_m(\theta) = 1$, then the distribution has converged and is therefore a fixed point, since the client update does not change the posterior approximation any longer. Furthermore, this proposition is significant, since it guarantees that if we converge to a single solution across clients, then we converge to a global solution that recovers the VI posterior approximation to the true posterior distribution, as in equation 3.2. However, this does not guarantee a global minimum being achieved and only a local minimum being found, due to the non-convexity of equation 3.2 in general, which VI also does not guarantee. Nevertheless, this result does not grantee convergence of the clients to a single solution, since we could potentially oscillate between solutions, Bui et al. (2018).

This result, as well as the interpretation of equation 3.3 as a way of deriving update equations, build the theoretical justifications for the use of PVI as a probabilistic inference scheme, Bui et al. (2018). Furthermore, the adoption of PVI in practice, such as in Heikkilä et al. (2023), Corinzia et al. (2021) and Kassab and Simeone (2022), further demonstrates the use of PVI as an algorithmic framework for deterministic approximate Bayesian inference.

PVI recovers Variational Inference if we only have a single client, i.e. $M = 1$. This means that the cavity distribution is the prior distribution and the local likelihood $p_m(\boldsymbol{x}_m|\theta)$ is the likelihood of the entire data set.

3.2 Server Update Scheduling

Bui et al. (2018) and Ashman et al. (2022) proposed different server scheduling approaches to PVI which we explore here.

3.2.1 Sequential

The first, perhaps most intuitive approach to scheduling PVI updates, and as motivated from the expectation propagation literature, is the sequential approach. During this, each client refines it's approximate likelihood term in sequence, based on the previous client update. Sequential PVI is therefore the most time inefficient version of PVI, since only a single client effectively does optimisation at a time. However, this procedure is also the most theoretically appealing approach to PVI, since we do not require the computation of a normalising constant Z_q, since $q_m(\theta)$ is a proper probability distribution and hence including $\Delta_m(\theta)$ into the global posterior one at a time restricts the global distribution to remain a normalised probability distribution, even without explicitly computing a normalising constant. We can show this as in Ashman et al. (2022) for some iteration i where we return $\Delta_m^{(i)}(\theta)$ for client m to the server. Then the new global approximation equals the local approximation computed at client m.

$$q^{(i)}(\theta) = \pi(\theta) \prod_{m=1}^{M} t_m^{(i)}(\theta) = t_m^{(i)}(\theta)\pi(\theta) \prod_{k \neq m} t_k^{(i-1)}(\theta)$$

$$= \pi(\theta) \frac{t_m^{(i)}(\theta)}{t_m^{(i-1)}(\theta)} \prod_{k=1}^{M} t_k^{(i-1)}(\theta) = \Delta_m^{(i)}(\theta) q^{(i-1)}(\theta) = \frac{q_m^{(i)}(\theta)}{q^{(i-1)}(\theta)} q^{(i-1)}(\theta) = q_m^{(i)}(\theta)$$

And since this local approximate posterior is required to a be proper probability distribution, the global approximate posterior will also be a normalised distribution. This holds by induction, since $\pi(\theta) = q^{(0)}(\theta)$ is normalised.

This sequential procedure, where we effectively pass the approximation of the previous client to the next client is illustrated in figure 3.2 and effectively lets the schedule $\mathcal{B}_i$ of algorithm 6 be a single client index at each iteration i, choosing a different client each subsequent iteration until all clients have updated.

The fact that we do not require an explicit normalising constant for sequential updates implies that we can fix $Z_q = 1$ and hence $\log Z_q$ in equation 3.3 is equal to 0. Therefore we have a rigorous link between the global VI approach to the PVI approach with more than one client.

However, even though this approach is appealing theoretically, this isn't very federated in nature due to almost all clients being idle at a time, and becomes particularly concerning under heterogeneity in both clients and data. We can avoid not receiving an update when a client isn't available at a time by simply passing the current posterior one client further, however if a single client takes particularly long to update, then the whole procedure becomes inefficient and slow. Additionally, we have observed in our experiments, as well as those in Bui et al. (2018) and Ashman et al. (2022), that if the clients have highly heterogenous data, such as in classification where clients contain only data of a single label of a multi–label classification data set, then the sequential approach fails to learn an accurate model.

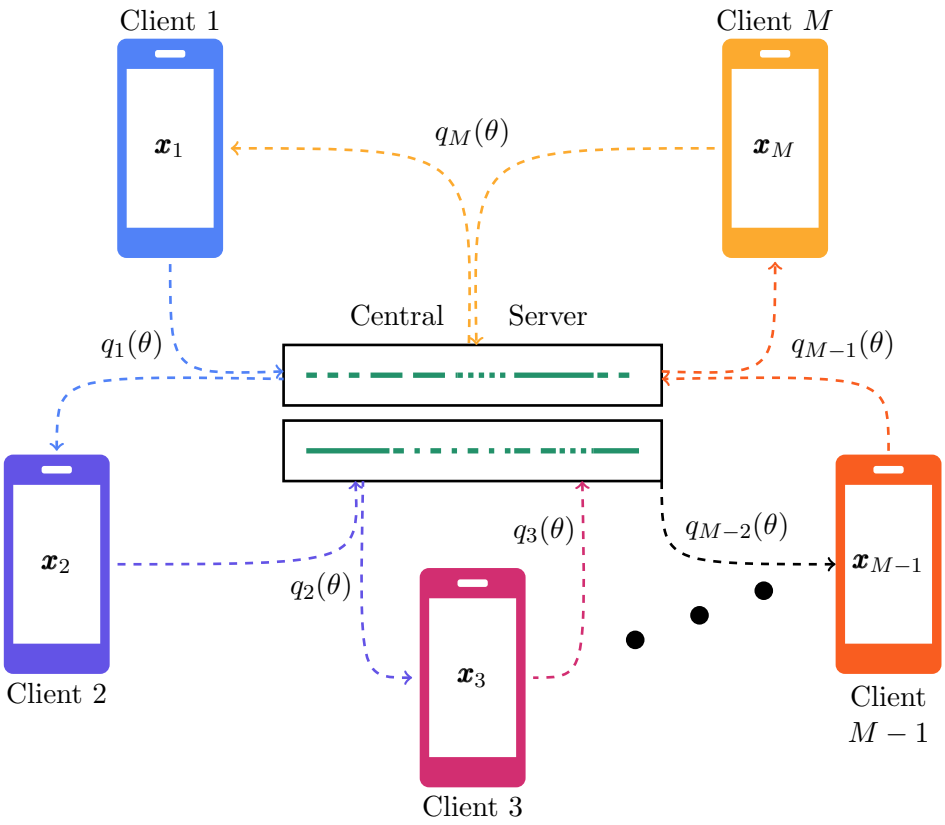


Figure 3.2: **Sequential PVI** illustration. Each client updates in sequence, based on the posterior computed by the immediately previous client.

This could be in part due to earlier clients having larger influence over their later counterparts. To formalise this notion, we have developed the following proposition based on the general concentration result of Miller (2021).

Lemma 14 *For some $\theta_0 \in \Theta$ that is closest to the local data generating process, then for the local posterior to concentrate around θ_0, we require*

$$Q^{\backslash m}(A_\epsilon) = \int_{A_\epsilon} q^{\backslash m}(\theta) d\mu(\theta) > 0 \text{ for all } \epsilon > 0, \text{ such that } A_\epsilon = \{\theta \in \Theta : f(\theta) < f(\theta_0) + \epsilon\}$$

where $f(\theta) := -\lim_{n\to\infty} \frac{1}{n_m} \mathbb{E}_{F_m^*}\left[\log p_m(x_{1:n_m}|\theta)\right]$ as in theorem 12.

Proof This follows directly from theorem 12, since this is a requirement for posterior concentration around some parameter $\theta_0 \in \Theta$. Concentration results of standard Bayesian inference require us to place sufficient weight on the prior distribution around the parameter $\theta_0 \in \Theta$ on which the posterior concentrates, i.e. $\Pi(A_\epsilon) > 0$ is required. We denote capital letters as the integral over the domain specified of their lower case counterpart distributions. In equation 3.4, we perform variational inference on the local data set with the cavity distribution as a local prior, therefore $Q^{\backslash m}(A_\epsilon) > 0$ is required, however if the cavity distribution is sufficiently misspecified with respect to the previously observed data, then this might not be the case. The set A_ϵ contains the parameters $\theta \in \Theta$ which are sufficiently close, in the sense of an ϵ–neighbourhood, to θ_0 through the likelihood function. However,

$Q^{\setminus m}(A_\epsilon)$ might not be greater than 0, since the cavity distribution is not defined as a prior distribution over the client data, but rather acts like one, without guaranteeing that it might not be skewed too much towards data from previous clients. ■

This result needs to hold in general and is not limited to sequential PVI and implies the Kullback–Leibler divergence between the local approximation and the cavity distribution regularises how close the client approximation is to the cavity distribution. However, sequential PVI can fail to capture the distribution of data that is too dissimilar across clients, not because of the previous lemma but rather because posteriors over concentrate towards the last seen data. This is prevalent especially in Bayesian Neural Networks (BNN).

Example 2 (BNNs with data heterogeneity) *Consider the training of a BNN with a 10–class classification data set, such as MNIST (LeCun et al., 1998) of handwritten digits, where 10 clients have only a single class each. In BNNs we typically place a mean–field unimodal Gaussian distribution over all parameters of the Neural Net, which means that posteriors are usually highly diffuse, since we have an over–parametrised model; in the case of a fully connected NN for the MNIST data set with a single hidden layer, we have 100,000s of parameters. Empirically, we can observe that the posterior accumulates around the class of the last seen client, since* $q^{(i)}(\theta) = q_m^{(i)}(\theta)$.

This causes under–regularisation through the use of the KL divergence in equation 3.4 which penalises deviation to the cavity distribution. The BNN observes the data and updates weights according to this one class shifting the output nodes of other classes away from the previously seen data of the other clients.

The regularisation of the KL divergence to misspecified prior distributions is explored in Knoblauch et al. (2022), where they develop a robust global variational inference framework to model misspecification, including prior misspecification as applicable here, and motivates us, in part, in the development of Partitioned Generalised Variational Inference, as explored in chapter 4.

Nevertheless, the sequential framework presented here, not limited to PVI, could be used if we do not trust a central server and instead perform the sequential nature by directly passing the new local posterior to the next client. However, under that assumption, we would not be secure against client adversaries that aim to poison the model through wrong data or adversaries that are passing a different distribution to the next client, that isn't based on the updated approximate posterior. This conundrum however raises the need for a central server that can be sufficiently trusted to not be an adversary to building an accurate global model, and using approaches such as local differential privacy to mitigate data leakage at the server, as developed in Heikkilä et al. (2023) for PVI.

3.2.2 Synchronous

The federated learning algorithms that we reviewed in detail in chapter 2.2 typically assume that we perform client updating in parallel for some subset of clients at a time, as in the gradient masked averaging approach to FedOpt and FedAvg of Tenison et al. (2023), as well as the Bayesian committee machine of Tresp (2000) or the federated posterior averaging

approach of Al-Shedivat et al. (2021). This parallel implementation is known as synchronous learning, where each client optimises based on a current global model and the server then aggregates the updates. We do this by sending the current approximate posterior to all clients and update the global posterior based on all the updates received.

$$q^{(i)}(\theta) \propto q^{(i-1)}(\theta) \prod_{m=1}^{M} \Delta_m^{(i)}(\theta) \tag{3.5}$$

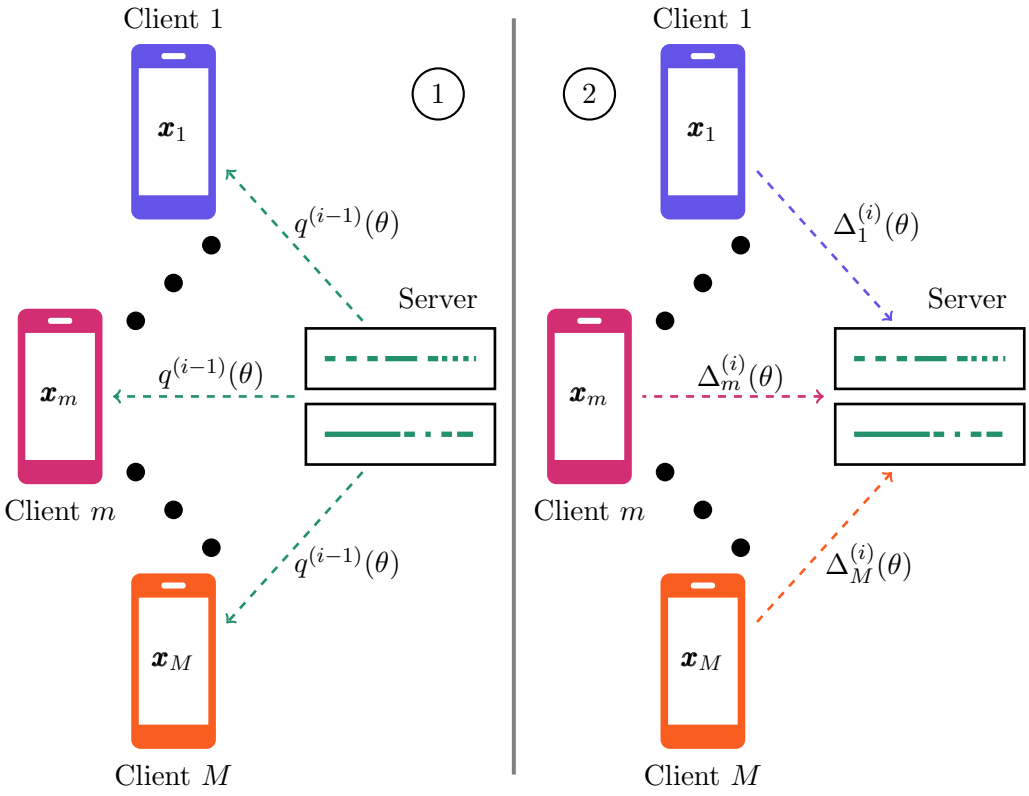


Figure 3.3: **Synchronous PVI** illustration. Each client, at some iteration i, updates based on a single global posterior in parallel and the server aggregates the results of all clients into a new posterior approximation according to equation 3.5.

This approach however requires us to compute a normalising constant at the server, since the product of the updates will not automatically normalise the new posterior distribution as in sequential PVI. This can be seen when decomposing equation 3.5 into the components of the $\Delta_m(\theta)$, Ashman et al. (2022).

$$\begin{aligned} q^{(i)}(\theta) &\propto q^{(i-1)}(\theta) \prod_{m=1}^{M} \Delta_m^{(i)}(\theta) \\ &= q^{(i-1)}(\theta) \prod_{m=1}^{M} \frac{t_m^{(i)}(\theta)}{t_m^{i-1}(\theta)} \\ &\propto q^{(i-1)}(\theta) \prod_{m=1}^{M} \frac{q_m^{(i)}(\theta)}{q^{\setminus m}(\theta)\, t_m^{(i-1)}(\theta)} \end{aligned}$$

$$\propto q^{(i-1)}(\theta) \prod_{m=1}^{M} \frac{q_m^{(i)}(\theta)\, t_m^{(i-1)}(\theta)}{q^{(i-1)}(\theta)\, t_m^{(i-1)}(\theta)}$$
$$= \frac{\prod_{m=1}^{M} q_m^{(i)}(\theta)}{[q^{(i-1)}(\theta)]^{M-1}}$$

This, depending on our choice for the approximating family $\mathcal{Q}$, can be tractable to compute such as for multivariate Gaussian distributions where we can multiply them together and obtain another multivariate Gaussian distribution, Buchholz et al. (2023). Note however, that we mean probability distributions and not random variables, since the product of Gaussian random variables is not Gaussian.

Furthermore, we can extend this synchronous framework to only consider a subset of clients at each iteration, such that we set the update schedule $\mathcal{B}_i$ in algorithm 6 to either be $\mathcal{B}_i = \{1, ..., M\}$ or we can randomly sample a subset of clients of some size B to optimise over, such that $\mathcal{B}_i$ is a random subset chosen uniformly without replacement. This is also the typical approach in FedAvg and FedOpt, as illustrated in algorithms 3 and 4 respectively, and aims to account for unavailable clients at each iteration.

Using this synchronous approach also enables the use of other privacy preserving approaches, such as global differential privacy against external malicious actors, or secure aggregation protocols, to protect from adversarial servers, Chen et al. (2022) and Kairouz et al. (2021).

3.2.3 Asynchronous

The drawbacks of sequential and synchronous approaches are that they aren't straggler resilient (Tziotis et al., 2023), meaning that we have to wait for all clients to finish optimisation, either in turn or in parallel. This problem is not limited to PVI but is especially detrimental here since in difference to, e.g. FedAvg, after we converge locally we cannot do further optimisation at a client level since the cavity distribution will not change. Hence, we do not get a better update with more local epochs. Therefore, we require asynchronous approaches to mitigate this, in which a client updates whenever it is available to do so, for however long it needs to converge, and then sends it's update to the server to aggregate into a new posterior. See figure 3.4 for a single client optimisation step. The current posterior at the server when an update arrives might not be the same posterior that was send to the client to do optimisation with. This is however allowed since each update at the server effectively replaces a clients old approximate likelihood with a new likelihood through $\Delta_m(\theta)$. This highlights the need for normalisation at the server, since the update cannot automatically normalise the new posterior since $\Delta_m(\theta)$ was computed using pervious likelihood terms that no longer are included in the current global posterior.

For an update computed based on a global approximate posterior $q^{(t)}(\theta)$ at time t, with the update taking time Δt, as in figure 3.4 we can compute the new global posterior at time $t + \Delta t$ based on a current approximate posterior at the server.

$$q^{(t+\Delta t)}(\theta) \propto q^{\text{current}}(\theta) \Delta_m^{(t+\Delta t)}$$

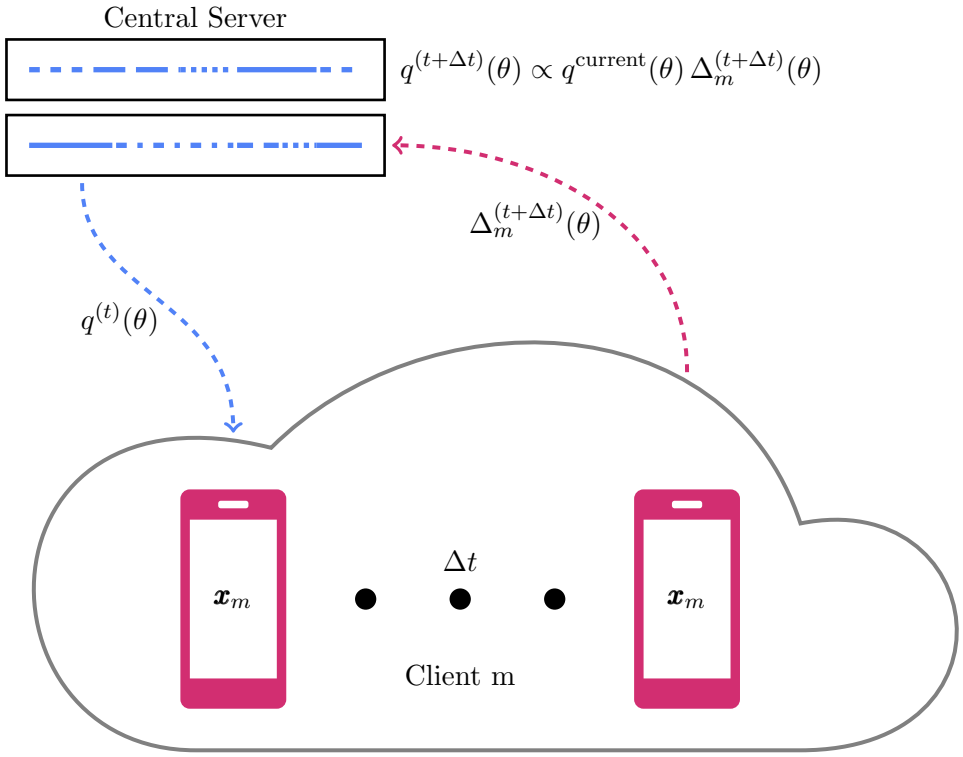


Figure 3.4: **Asynchronous PVI** visualisation. We illustrate this for a single client m that optimises based on some posterior and after it finishes optimisation sends the update to the server to aggregate. The global approximate posterior might have changed during the local optimisation.

$$= \pi(\theta) \frac{t_m^{(t+\Delta t)}(\theta)}{t_m^{(t)}(\theta)} \prod_{m=1}^{M} t_m^{\text{current}}(\theta)$$

Since the approximate likelihood of client m has not changes, $t_m^{\text{current}}(\theta) = t_m^{(t)}(\theta)$ we have:

$$q^{(t+\Delta t)}(\theta) \propto \pi(\theta) t_m^{(t+\Delta t)}(\theta) \prod_{k \neq m} t_k^{\text{current}}(\theta)$$

We therefore replace the old approximate likelihood with the new approximate likelihood of client m and renormalise the posterior.

This simultaneously highlights potential challenges of asynchronous approaches, since some updates might be based on old approximate likelihood terms of different clients, so that the updates become stale, Ashman et al. (2022).

3.2.4 Damping in Synchronous and Asynchronous approaches

Ashman et al. (2022) found that for the synchronous and asynchronous frameworks, we often require "damping" of the approximate likelihood terms in order to have more stable convergence and making normalisation at the server possible. Damping decreases the effect an update has on the new posterior approximation, which Ashman et al. (2022) show empirically to result in faster and more stable convergence. For a damping factor $\rho \in (0, 1]$

the new approximate likelihood can be found through

$$t_m^{(i)}(\theta) \propto \Big(\frac{q_m^{(i)}(\theta)}{q^{(i-1)}(\theta)}\Big)^{\rho} t_m^{(i-1)}(\theta) \tag{3.6}$$

Ashman et al. (2022) say that choosing $\rho \propto \frac{1}{M}$ results in stable convergence empirically, not giving any theory for this.

We noticed, however, that we can relate this to logarithmic opinion pools (Genest, 1984) when ρ is chosen as the fraction of all clients in the current approximation for synchronous PVI. We can rewrite equation 3.6 as follows and explore this at the server aggregation step.

$$\frac{t_m^{(i)}(\theta)}{t_m^{(i-1)}(\theta)} \propto \Big(\frac{q_m^{(i)}(\theta)}{q^{(i-1)}(\theta)}\Big)^{\rho}$$

$$q^{(i)}(\theta) \propto q^{(i-1)}(\theta) \prod_{m=1}^{M} \frac{t_m^{(i)}(\theta)}{t_m^{(i-1)}(\theta)} \propto q^{(i-1)}(\theta) \prod_{m=1}^{M} \Big(\frac{q_m^{(i)}(\theta)}{q^{(i-1)}(\theta)}\Big)^{\rho}$$

Then, if $\sum_{m=1}^{M} \rho = 1$ we have that $(q^{(i-1)}(\theta))^{M\rho} = q^{(i-1)}(\theta)$ such that:

$$q^{(i)}(\theta) \propto \prod_{m=1}^{M} \big(q_m^{(i)}(\theta)\big)^{\rho}$$

And by Genest (1984), with respect to some common dominating measure μ, say the Lebesgue measure on finite dimensional[1] Θ, then the externally Bayesian operator that we use to calculate the new approximate posterior is a logarithmic opinion pool of the form:

$$q^{(i)}(\theta) = \frac{\prod_{m=1}^{M} \big(q_m^{(i)}(\theta)\big)^{\rho}}{\int_{\Theta} \prod_{m=1}^{M} \big(q_m^{(i)}(\theta)\big)^{\rho} \, d\mu(\theta)} \qquad \mu - \text{a.s.}$$

This is significant, since it resembles not only the Bayesian Committee Machine (Tresp, 2000), but also a mixture of experts system, in which a global decision is made through the agreement of different experts weighted according to some parameter, in this case evenly with $\rho = \frac{1}{M}$.

Since the $q_m(i)(\theta)$ contain the prior distribution implicitly, this also closely resembles the consensus Monte Carlo algorithm presented in Scott et al. (2016), where $\pi(\theta)^{1/M}$ is included inside the product.

$$\prod_{m=1}^{M} \big(q_m^{(i)}(\theta)\big)^{\rho} = \prod_{m=1}^{M} \big(\pi(\theta) t_m^{(i)}(\theta) \prod_{k \neq m} t_k^{(i-1)}(\theta)\big)^{\rho} = \prod_{m=1}^{M} \big(\pi(\theta) t_m^{(i)}(\theta)\big)^{1/M} \big(t_m^{(i-1)}(\theta)\big)^{(1-1/M)}$$

Furthermore, we can change the weight that we place on a single local posterior by considering different factors ρ_m as long as $\sum_{m=1}^{M} \rho_m = 1$. This allows us to either weight the

1. Pinski et al. (2015) considers the set of Gaussian measures for infinite dimensional Θ, since the Lebesgue measure is only defined on finite dimensional spaces such as $\mathbb{R}^n$. However, this is not done in the context of logarithmic opinion pools, but rather for KL minimisation on infinite dimensional spaces, such as a polish space Θ.

opinions according to the amount of data, as in the aggregation step of FedAvg, or by considering some measure of accuracy of local posteriors as in classification where we can evaluate clients posteriors based on a global test set of unseen data points.

3.3 Comparing Tilted Variational Bayes and PVI

The idea of using the cavity distribution in a VI style algorithm is not novel to PVI, but was first (as far as we are aware) developed in Hensman et al. (2014), where this is used to develop a global approach to VI, Tilted Variational Bayes, that has faster computation time. The name comes from the use of the cavity distribution multiplied by the local data likelihood, $\frac{q^{\setminus m}(\theta)p_m(\boldsymbol{x}|\theta)}{\hat{Z}_m} = \hat{q}_m(\theta)$ which is also known as the tilted distribution in EP, Vehtari et al. (2020).

Hensman et al. (2014) consider a similar set–up as traditional EP, where the entire data set $x_{1:n}$ decomposes across data points. Furthermore, they consider only the case where the global parameters θ decompose into individual local latent variables, say θ_i for $i \in [n]$, such that the likelihood function is $p_i(x_i|\theta_i)$. This assumption turns out to be crucial in their derivation, since it automatically normalises their approximate posterior, as we explore below. We note that Hensman et al. (2014) do not explain the lack of a normalisation constant in their paper and furthermore, do not go into details about shared latent variables or global latent variables, in which cases we would require a normalisation constant for the posterior approximation. In difference to PVI the approximation $q(\theta)$ that they choose does include data, since it is defined as the product of all normalised tilted distributions, Hensman et al. (2014).

$$q^*_{TVB}(\theta) = \prod_{i=1}^{n} \hat{q}_i(\theta_i) \qquad \text{where} \qquad \hat{q}_i(\theta_i) = \frac{q^{\setminus i}(\theta_i)p_i(x_i|\theta_i)}{\int q^{\setminus i}(\theta_i)p_i(x_i|\theta_i)d\theta_i}$$

This product is normalised, since by Fubini's theorem we can exchange the order of integration, and each tilted distribution is normalised, $\int \hat{q}_i(\theta_i)d\theta_i = 1$.

$$\int \prod_{i=1}^{n} \hat{q}_i(\theta_i)d\mu(\theta) = \int \hat{q}_1(\theta_1)...\Big(\cancel{\int \hat{q}_n(\theta_n)d\mu(\theta_n)}^{\,1}\Big)...d\mu(\theta_1) = 1$$

Using TVB we can lower bound the log marginal likelihood with $q_{TVB}(\theta)$ as in VI.

$$\begin{aligned}
\text{ELBO}(q) &= \mathbb{E}_{q_{TVB}(\theta)}\Big[\log \frac{\pi(\theta)\prod_{i=1}^n p_i(x_i|\theta_i)}{\prod_{i=1}^n q^{\setminus i}(\theta_i)\, p(x_i|\theta_i)/\hat{Z}_i}\Big] \\
&= \mathbb{E}_{q_{TVB}(\theta)}\Big[\log \frac{\pi(\theta)}{\prod_{i=1}^n q^{\setminus i}(\theta_i)}\Big] + \sum_{i=1}^{n} \log \hat{Z}_i \\
&= \sum_{i=1}^{n} \log \hat{Z}_i + \sum_{i=1}^{n} \text{KL}(\hat{q}_i(\theta_i)||q^{\setminus i}(\theta_i)) - \text{KL}(q_{TVB}(\theta)||\pi(\theta))
\end{aligned}$$

This is however only a global approach to VI since the approximation $q(\theta)$ contains all data points, and furthermore requires the global parameters to be local latent variables, so

that the approximate posterior is automatically normalised. PVI approaches this idea of using the EP approximate likelihoods in VI through distributed computation that allows for federated learning without knowing the data inside the approximation.

3.4 Partitioned Variational Inference and Federated Learning

PVI as a framework for federated learning does not address most challenges by itself, but as we hinted at in the previous sections we can extend it to perform better under certain conditions in the system environment. As of now, PVI has little theoretical guarantees except for proposition 13, and the way in which the global VI objective relates to the local PVI objectives through equation 3.3. However, we are not guaranteed to converge in general, and even if we could guarantee convergence we have not established consistency, asymptotic normality nor coverage results for PVI posteriors, as in Miller (2021) for the generalised posteriors of Bissiri et al. (2016), or as in Knoblauch (2019) for Generalised Variational Inference.

PVI addresses heterogeneity partially through the updating schedule. For clients, choosing the asynchronous approach can mitigate resilience to stragglers and client unavailability. The quality of client data, however, is not addressed since we assume that the likelihood function $p_m(\boldsymbol{x}_m|\theta)$ is correctly specified, and hence if clients are only able to observe some noisy data without knowing the distribution of the noise, e.g. due to faulty equipment, then the likelihood will no longer be correctly specified, Bernardo and Smith (2000). Data heterogeneity, can partially be addressed though synchronous approaches and weighting the updates of clients according to the quantity or quality, although not as straight forward as going by quantity, of data.

The system of clients might be massively distributed, however PVI should be able to deal with this for i.i.d. data fairly easily, however normalising across a large amount of client updates might be computation heavy. This can be addressed by choosing simpler approximating families and likelihoods that are conjugate and hence allow for efficient normalisation. Furthermore, assuming that the prior and likelihood function of a client are in some conjugate exponential family, then Heikkilä et al. (2023) show that the number of partitions do not affect the PVI posterior.

Communication and computation might be challenging for devices and depending on the update scheme we choose, as well as the damping factor, we might take longer to converge, Ashman et al. (2022).

Privacy of local data is not guaranteed for PVI, and therefore Heikkilä et al. (2023) have developed a differentially private approach to PVI, where PVI is extended to both local and global differential privacy. The aim of which being to obscure the data, such that an adversary would not be sure what input the data was from some set of 'neighbouring' data points. However, they restrict themself to exponential family distributions, on which their their algorithms depend.

Having clients as an adversary to model development has not been explored in PVI. These adversarial clients might poison the data directly or the model updates they provide,

Kairouz et al. (2021). Gradient masked averaging (Tenison et al., 2023) might be a way of dealing with bad updates of individual clients, through ignoring components if they are too dissimilar across all updates. Such adversarial robustness provides an interesting area of research for the future.

Privacy of data also includes the guarantees of deciding at what point data should be forgotten and hence removed from a trained model as in Bayesian unlearning, see Nguyen et al. (2020). The motivation of privacy, and in conjunction with adversarial client robustness, where we learn at some point in time that a client intentionally or unintentionally provided false data for training, implies that we might want to remove some data from our current model. It is currently unclear how this can be applied to PVI, since we update locally using the cavity distribution which models the data of other clients, and hence cannot simply divide out the likelihood of the data as in VI, Nguyen et al. (2020). This provides further areas of research for the future that we want to continue exploring.

3.5 Partitioned Variational Inference and Continual Learning

PVI also extends to variational continual learning, as highlighted in Bui et al. (2018). In continual learning, we observe data in sequence and cannot necessarily revisit past data. Therefore, we can view continual learning as a single pass of sequential PVI, Bui et al. (2018). We require such approximation algorithms for continual learning, due to it being an NP–hard problem, Knoblauch et al. (2020). However, doing so with PVI does not add to the continual learning literature besides unifying it with traditional VI and distributed VI under the umbrella of PVI.

In continual learning, we consider the client data to be partitioned over time, i.e. earlier clients correspond to earlier observations. Then the use of PVI sequentially and with only a single iteration over clients implies that the local cavity distribution is the previous posterior, as highlighted previously. This recovers vanilla variational continual learning, since approximate likelihoods are initialised to be 1, and we do not necessarily place a limit on the number of clients, since we only perform a single pass through the clients. Hence, if new data arrives, we can treat it as the next client in the sequence and do optimisation using the new data as well as the previous posterior as a cavity distribution.

Therefore, if our current approximation is $q^{(i)}(\theta)$ then if we observe new data, $\boldsymbol{x}_{i+1}$, equation 3.4 becomes

$$q^{(i+1)}(\theta) = \underset{q(\theta)\in\mathcal{Q}}{\arg\min}\Big\{\mathbb{E}_{q(\theta)}\big[-\log p_{(i+1)}(\boldsymbol{x}_{i+1}|\theta)\big] + \mathrm{KL}\big(q(\theta)||q^{(i)}(\theta)\big)\Big\}$$

3.6 Implementation Details

To illustrate and understand the fundamentals of PVI, we implemented PVI from scratch[2].

2. All experiments were performed on a single laptop and do not require GPU clusters. All code for the experiments can be found at:
`https://github.com/Terje-M/Partitioned_Generalised_Variational_Inference`

During the experiments in chapter 3.7, and as motivated by Minka (2001b), we consider the following problem: Finding the posterior distribution of a (multivariate) Gaussian distribution that is embedded in some noisy data, where the noise if generated by some different, heteroscedastic (multivariate) Gaussian. The data generating process thus forms a heteroscedastic mixture of (multivariate) Gaussian distributions.

This problem was for instance considered in Minka (2001b), where it was known as the "Clutter problem", see also Bishop (2006) for further details on EP under this problem. We consider this problem due to several reasons; (1) being that it can be varied enough to produce interesting challenges, McLachlan and Peel (2000), (2) outliers are prevalent in most practical settings and seeing how PVI is able to deal with these, even in toy data sets, underscores it's use, and (3) it can be sufficiently controlled where all variables are known, so that changing specific instances allows investigation of PVI. Furthermore, the use of a mixture of Gaussians as a likelihood function has interesting challenges since the likelihood and prior are not conjugate and can therefore not be inferred by directly minimising a local KL divergence. Examples of such practical uses are Bayesian on–line changepoint detection, as investigated in Knoblauch et al. (2018), or in medical research for genetics or diabetes, McLachlan and Peel (2000).

We define the multivariate Gaussian distribution below.

Definition 15 (Multivariate Gaussian) *A multivariate Gaussian distribution about a parameter θ and hyper–parameters μ (mean vector) and Σ (covariance matrix), is defined as $\pi(\theta) \sim \mathcal{N}(\mu, \Sigma)$*

$$\pi(\theta|\mu, \Sigma) := \frac{1}{\det(2\pi\Sigma)^{\frac{1}{2}}} \exp\Big\{-\frac{1}{2}(\theta-\mu)^\top \Sigma^{-1}(\theta-\mu)\Big\}$$

Therefore, our contaminated model of a multivariate Gaussian distribution is:

$$p(\boldsymbol{x}|\mu, \Sigma, \nu, \Lambda) = (1-w)\,\mathcal{N}(\boldsymbol{x}|\mu, \Sigma) + w \cdot \epsilon(\boldsymbol{x}|\nu, \Lambda) \tag{3.7}$$

where $\epsilon(\boldsymbol{x}|\nu, \Lambda) \sim \mathcal{N}(\nu, \Lambda)$ and represents the noise in our model. The term $w \in [0, 1]$ is a parameter that specifies the likelihood of a data point coming from either of these distributions. In our experiments, we model this data generating process by using the PyTorch distributions library and sampling a specified number of data points from equation 3.7, where each data point's distribution is the noise term with probability w and the true distribution with probability $1-w$. We call such a distribution unimodal if the distribution has a single peak, and therefore the two distributions are sufficiently close or have large covariance matrices. We call a distribution bimodal, if it has two distinct peaks, see figure 3.5 for an illustration.

We can then place a multivariate Gaussian prior on this distribution:

$$\pi(\theta) \sim \mathcal{N}(\mathbf{0}, a\mathbf{I}_D)\,, \qquad a \in \mathbb{R}$$

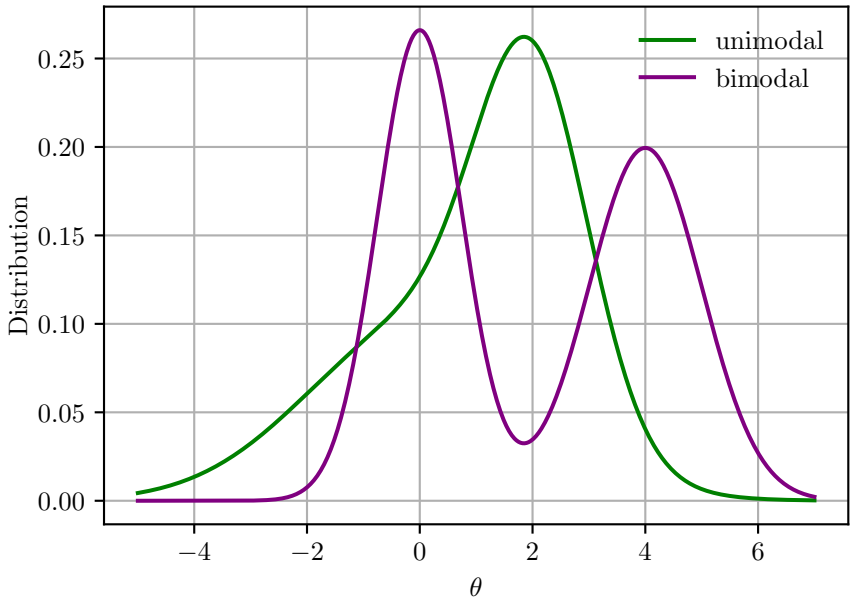


Figure 3.5: A unimodal and a bimodal mixture of two Gaussian distributions.

As we stated before, our aim is to model out the noise, hence finding a multivariate Gaussian approximation to the non–noisy posterior. We therefore choose $q(\theta)$ to lie within the multivariate Gaussian family fo distribution, hence having a mean parameter and a covariance matrix parameter.

$$q(\theta) \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{C}) \,, \qquad \boldsymbol{m} \in \mathbb{R}^D, \, \boldsymbol{C} \in \mathbb{R}^{D \times D}$$

In order to do this, the approximating likelihood terms will be unnormalised multivariate Gaussian distributions, of the form

$$t_m(\theta | \mu_m, \Sigma_m) = \exp\Big\{ -\frac{1}{2} (\theta - \mu_m)^\top \Sigma_m^{-1} (\theta - \mu_m) \Big\}$$

The normalising constant, in this case, is irrelevant, since the multivariate Gaussian is an exponential family distribution, and the formulation above contains the sufficient statistics. Therefore, we can multiply these terms together and normalise according to the formulation in Buchholz et al. (2023), where we derive a new covariance term and mean parameter of a multivariate Gaussian distribution that is the automatically normalised. For M such factors, the normalised multivariate Gaussian distribution is $\mathcal{N}(\mu_{\text{new}}, \Sigma_{\text{new}})$:

$$\Sigma_{\text{new}}^{-1} = \sum_{m=1}^{M} \Sigma_m^{-1}$$

$$\mu_{\text{new}} = \Sigma_{\text{new}} \sum_{m=1}^{M} \Sigma_m^{-1} \mu_m$$

The proof of which follows by taking the square inside the exponential, considering the terms containing θ, and rearranging the exponential into a the form of a multivariate Gaussian

distribution. Similarly, when diving out a term and normalising, we simply subtract the corresponding terms instead of adding them inside the summations above.

By initialising the mean to be the 0 vector and the covariance matrix to be infinity times the identity matrix, the approximate likelihood is 1. And hence the initial approximate posterior is the prior distribution.

A local update step for a client m with data set $\boldsymbol{x}_m$ which is independent and identically distributed locally, means equation 3.4 becomes:

$$q_m(\theta) = \underset{q(\theta \in \mathcal{Q})}{\arg\min} \Big\{ \mathbb{E}_{q(\theta)} \big[-\sum_{i=1}^{n_m} \log p(x_{m_i} | \theta, \Sigma, \nu, \Lambda) \big] + \mathrm{KL}\big(q(\theta) \big|\big| q^{\setminus m}(\theta) \big) \Big\}$$

The Kullback–Leibler divergence between two multivariate Gaussians is well defined, and can be phrased independent of θ and is therefore tractable to compute.

Lemma 16 *The Kullback–Leibler divergence between two multivariate Gaussian distributions, $q(\theta) \sim \mathcal{N}(\mu, \Sigma)$ and $\pi(\theta) \sim \mathcal{N}(\nu, \Lambda)$, such that $\mu, \nu \in \mathbb{R}^D$ and $\Sigma, \Lambda \in \mathbb{R}^{D \times D}$, is*

$$\mathrm{KL}(q||\pi) = \frac{1}{2} \Big((\mu - \nu)^\top \Lambda^{-1} (\mu - \nu) + \mathrm{trace}(\Lambda^{-1}\Sigma) + \log \frac{\det(\Lambda)}{\det(\Sigma)} - D \Big)$$

Proof For a proof, see Pardo Llorente (2006), or appendix C. ■

However, the expectation term is not easily computable, since it involves taking the logarithm of a sum of different exponentials. Therefore, we employ the use of Monte Carlo approximations, Murphy (2023). MC approximations approximate intractable integrals by repeatedly sampling θ from the distribution, $q(\theta)$, K times, then

$$\mathbb{E}_{q(\theta)} \big[-\sum_{i=1}^{n_m} \log p(x_{m_i} | \theta, \Sigma, \nu, \Lambda) \big] \approx \frac{1}{K} \sum_{k=1}^{K} \big[-\sum_{i=1}^{n_m} \log p(x_{m_i} | \theta_k, \Sigma, \nu, \Lambda) \big] \ , \quad \{\theta_k\}_{k=1}^K \sim q(\theta)$$

Using this, we can evaluate the inside of the local minimisation step efficiently. To find the minimum, we can differentiate the inside with respect to the variational parameters of $q(\theta | \boldsymbol{m}, \boldsymbol{C})$, this being $\boldsymbol{m}$ and $\boldsymbol{C}$, and evaluate minima, i.e. where these equal 0. However, since this loss landscape is typically not convex nor smooth, gradient descent would only find local minima, and not a global minimum. Hence, we use stochastic optimisers such as SGD, Adagrad or as in our case Adam, Kingma and Ba (2015). The Adam optimiser depends on a learning rate or step size η, which we vary according to the problem we consider, and the decay rates $\beta_1, \beta_2 \in [0, 1)$, which we leave unchanged throughout the experiments as $\beta_1 = 0.9$ and $\beta_2 = 0.999$, as well as a small, close to 0, value $\epsilon = 10^{-8}$ which we leave unchanged to avoid division by 0. We do not cover the details of Adam here—we have shown the aggregation steps of Adam as part of algorithm 4 and details can be found in Kingma and Ba (2015)—since PyTorch allows us to use optimisers.

Another reason for using PyTorch with the inbuilt Adam optimiser is the use of the Monte Carlo approximation above and how to differentiate this with respect to the variational parameters, since this is evaluated a constant. Nevertheless, we can differentiate through the expectation via the "reparametrisation trick" (Murphy, 2023). We do not go

into the details why or how this works here, see Murphy (2023, Chapter 10.2.1) for details, since PyTorch allows us to sample from the approximate posterior and differentiate through this by simply using the *rsample()* function, and hence, we do not have to implement the reparametrisation.

EP does not need an optimisation method, since the equations are known in closed form. For the details on the EP implementation for this specific problem we refer the reader to Minka (2001b) and Bishop (2006, Chapter 10.7.1) who cover this extensively.

3.7 Experiments

We examine a number of toy data set experiments based on the problem discussed in the previous section. Unless otherwise stated, we use sequential PVI for the experiments below. We begin by considering the problem presented in Minka (2001b) for unimodal and bimodal distributions, comparing EP, VI and PVI, for one dimensional data. Our main motivation in these experiments is seeing how the PVI posterior compares to traditional Variational Inference, since we motivate PVI by rearranging the global VI objective. For VI we have all the data points at a single client, and otherwise we use the same specifics in VI as in PVI.

3.7.1 Unimodal Gaussian Mixture Model

Let the data generating process be $p(x|\theta) := \frac{1}{2}\mathcal{N}(2,1) + \frac{1}{2}\mathcal{N}(0,2)$, where the second term is noise. We use the learning rate of $\eta = 0.01$, 50 epochs across all clients and 30 optimisation steps for each client iteration. We sample 40 data points from the DGP and distribute them across 40 clients, where we set the prior $\pi(\theta) = \mathcal{N}(0, 10)$.

For the unimodal mixture model, as shown in figure 3.6, we can see that all three methods model the data and the data generating distribution reasonably well. PVI performs similarly to VI as is desired, while being less overconfident as EP is in it's prediction, and hence better at uncertainty quantification. This illustrates, that under a correctly specified likelihood function, we can expect PVI to somewhat model out the noise and not be too sensitive to tail data, as it is mode seeking, and we know the noise distribution.

3.7.2 Bimodal Gaussian Mixture Model

We consider the likelihood function $p(x|\theta) = \frac{1}{2}\mathcal{N}(4,1) + \frac{1}{2}\mathcal{N}(0,0.75)$ which generated 40 data points. Using learning rate $\eta = 0.01$, 50 iterations over all 40 clients, each containing a single data point, where each iteration has 30 Adam iterations, then we have used the prior $\pi(\theta) = \mathcal{N}(0, 10)$ to get figure 3.7 below. VI uses the same set–up, where we only have a single client containing the entire data set.

The bimodal data distribution appears to be more challenging for EP to properly handle, since q_{EP} is skewed towards the noisy data, while being overconfident in it's prediction. This is due to the nature of the reverse KL divergence used in EP where it is mode–avoiding (Minka, 2005), and tries to cover both modes, while being required to be a normal distribution and knowing the noise distribution. For PVI, this does not seem to be a challenge, since the KL is mode seeking, and furthermore it behaves similarly to VI on the

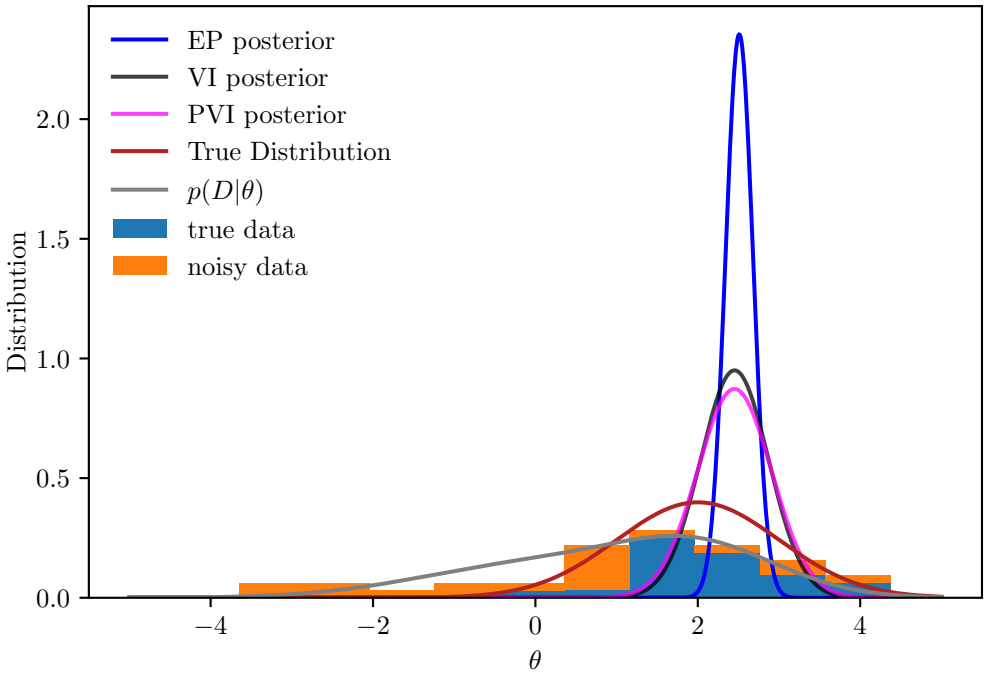


Figure 3.6: Best viewed in colour. EP vs VI vs PVI on a unimodal, one–dimensional model.

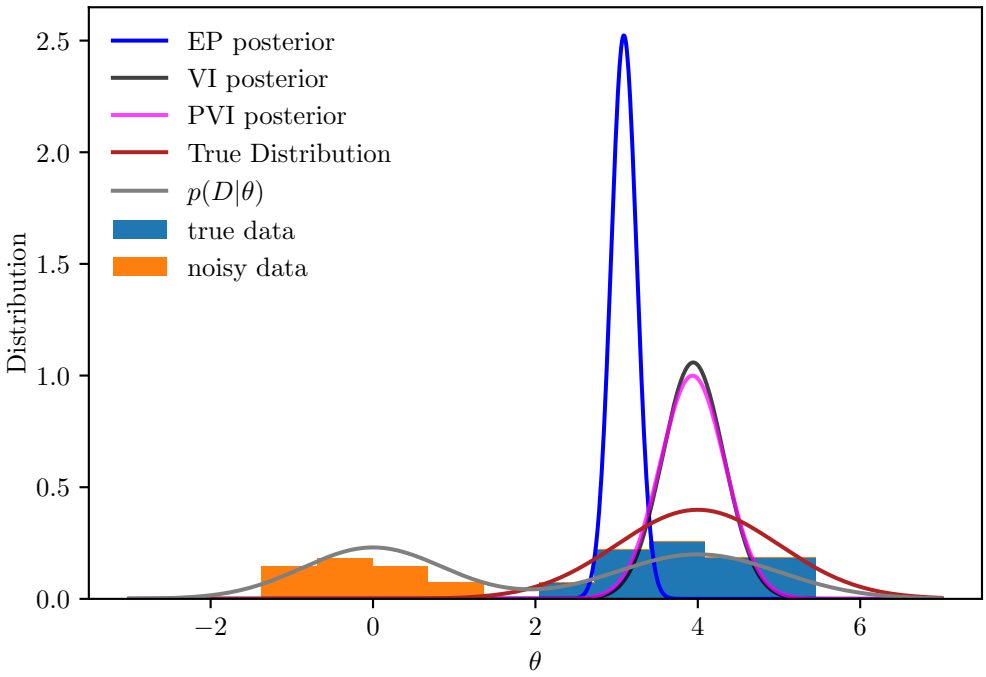


Figure 3.7: Best viewed in colour. EP vs VI vs PVI on a bimodal, one–dimensional model.

above data set, as well as in repetitions of the experiments in figure 3.7. This highlights desirable properties of PVI, mainly being that we are able to recover the VI posterior, even if each client only has a single data point.

3.7.3 Effect of different amounts of Clients

Since we conjecture about the PVI posterior being equal to the VI posterior even if each data point has it's own client, we consider now the effect of splitting data across clients in comparison to standard VI. Define the data generated by the following likelihood function:

$$p(x|\theta) = \frac{1}{2}\mathcal{N}((1.5, 2.5)^\top, 0.8 \times I_2) + \frac{1}{2}\mathcal{N}((1,1)^\top, 1.5 \times I_2)$$

where I_2 is the two dimensional identity matrix. We generate 50 data points from this and split the data homogeneously (evenly) across the amount of clients as specified in table 3.1. We consider the prior $\pi(\theta) = \mathcal{N}((0,0)^\top, 10I_2)$, and keep the learning rate $\eta = 0.002$ fixed across all client splits. We perform VI using the same learning rate on the entire data set so that we can compare the mean and covariance matrices between these.

To expand on this notion of client heterogeneity and distributed systems in FL, we investigate how different amounts of data set splits of one single data set, affects the PVI posterior distribution, and how far it deviates from the VI posterior as a reference. We have found that in most cases, PVI is quite similar to VI, the results are summarised in table 3.1 and illustrated in figure 3.8. We employ the euclidean norm as a measure of deviation from the VI posterior mean, and the Frobenius norm difference of the covariance matrices of VI and PVI, as a measure of how similar the matrices are, as well as the notion of the (log) Standard Generalised Variance, Knoblauch and Damoulas (2018), which is defined as the (log) determinant of a covariance matrix, and determines how different the generalised variances of the VI and PVI posteriors are. When comparing these results we can determine, that the number of clients does not have a large impact on the approximate posterior distribution of PVI, and while we do not exactly recover the VI posterior, our results come very close to it in terms of mean and covariance parameters. Figure 3.8 highlights this similarity. Therefore, we can expect, irregardless of the number of clients we employ, PVI to behave similar to VI.

3.7.4 Update Schedule comparison

For the update schedule comparison we use a more complex distribution, where we let the covariance matrix be any positive definite matrix, since we define the likelihood function for the data as:

$$p(x|\theta) = 0.65\mathcal{N}\Big(\begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}3 & 2.5\\2.5 & 3\end{bmatrix}\Big) + 0.35\mathcal{N}\Big(\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}2.5 & -1.8\\-1.8 & 2\end{bmatrix}\Big)$$

We generate 50 data points through this and split the data across 10 clients, each having 5 unique data points. For PVI, we consider the learning rate to be $\eta = 0.0005$ and for VI we use $\eta = 0.002$. The amount of Adam optimisation epochs per client is fixed to be 30

Table 3.1: Quantitative results for different amount of clients in PVI posteriors in comparison with the standard VI posterior distribution, all clients have the same 50 point data set and identical prior conditions.

Number of Clients	Mean difference $\|\|\mu_{VI} - \mu_{PVI}\|\|_2$	Covariance matrix difference $\|\|\Sigma_{VI} - \Sigma_{PVI}\|\|_F$	Log SGV difference $\|\log\det(\Sigma_{VI}) - \log\det(\Sigma_{PVI})\|$
2 Clients	0.0422	0.0021	0.0636
3 Clients	0.0536	0.0053	0.1774
4 Clients	0.0320	0.0017	0.0548
5 Clients	0.0223	**0.0001**	**0.0045**
6 Clients	0.0314	0.0076	0.2285
8 Clients	**0.0209**	0.0116	0.3412
10 Clients	0.0292	0.0083	0.2505
12 Clients	0.0405	0.0018	0.0590
15 Clients	0.0328	0.0055	0.1845
17 Clients	0.0320	0.0096	0.2851
20 Clients	0.0341	0.0220	0.6035
25 Clients	0.0210	0.0013	0.0409
50 Clients	0.0310	0.0017	0.0546

per iteration, however we vary the amount of total iterations such that all update schedules converge to some distribution.

Table 3.2: Quantitative results for different update schedules in PVI posteriors in comparison with the standard VI posterior distribution, shown in figure 3.9.

Update Schedule	Mean difference $\|\|\mu_{VI} - \mu_{PVI}\|\|_2$	Covariance matrix difference $\|\|\Sigma_{VI} - \Sigma_{PVI}\|\|_F$	Log SGV difference $\|\log\det(\Sigma_{VI}) - \log\det(\Sigma_{PVI})\|$
Sequential	0.0363	4.3671	0.5720
Synchronous	**0.0279**	4.1251	0.4171
Asynchronous	0.5546	**0.1938**	**0.3547**

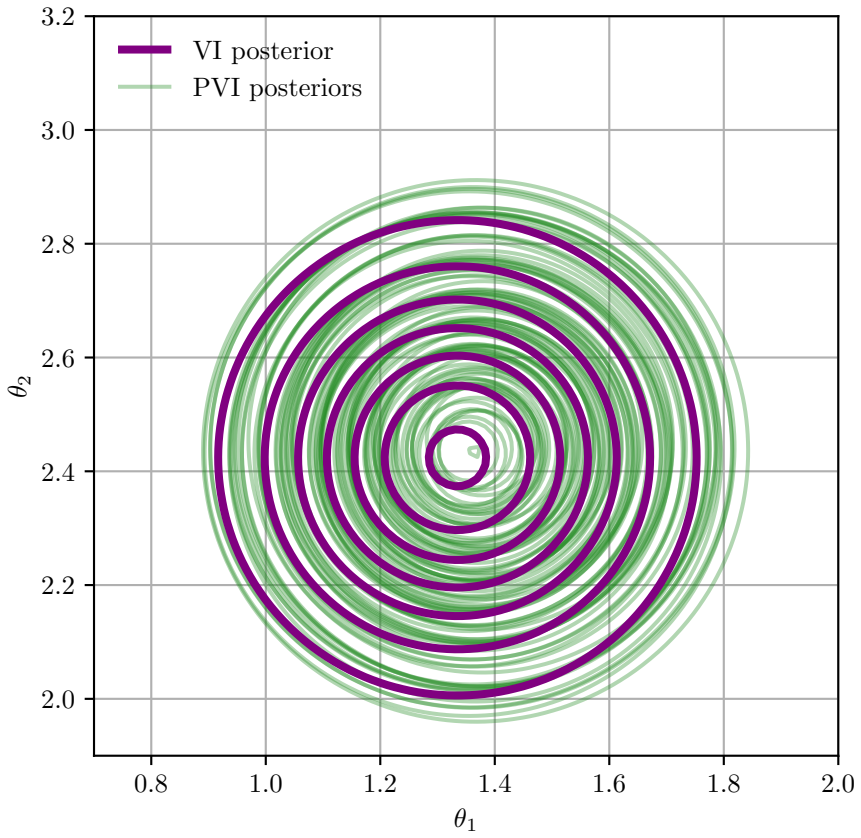
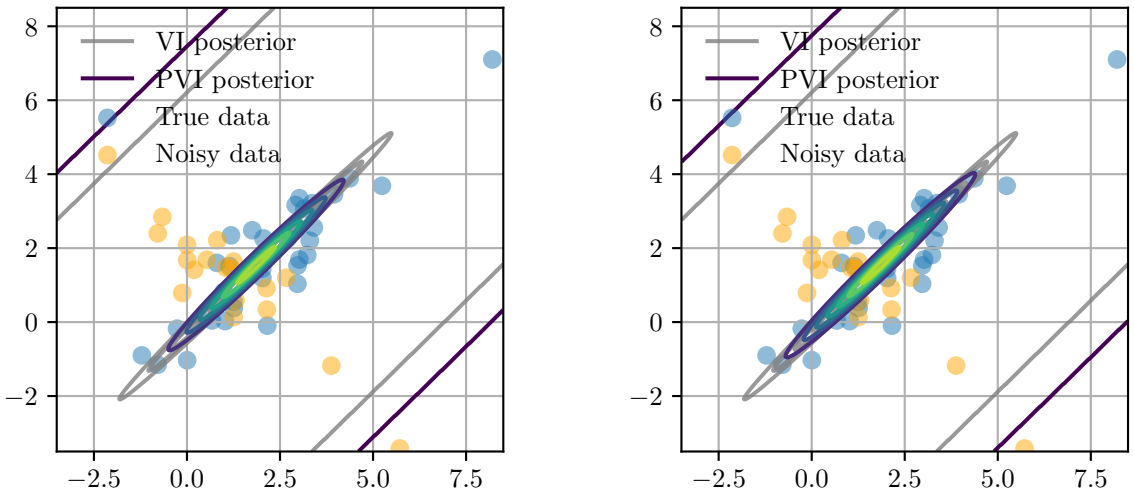


Figure 3.8: Best viewed in colour. PVI posteriors for different amounts of clients resemble the VI posterior.

The implementations of sequential and synchronous PVI are straightforward and follow from the equations. For asynchronous updating, we simulate this on a single machine by randomly assigning each client a specific wait time in terms of iterations by uniformly choosing a wait time from the set $\{0, 1, ..., 7\}$, before it's posterior can be aggregated into the model. This means some clients have more influence over the posterior than those with larger wait times.

As figure 3.9 and table 3.2 indicate that for all update schedules, we can expect the PVI posterior to behave similar to the VI posterior, even with more challenging covariance matrices as in the previous experiments. However, we can observe that for sequential and synchronous updating, where all data points are considered at an overall iteration, we have slightly better performance from synchronous PVI than the sequential counterparts in all metrics considered. This is interesting mainly because sequential PVI should be more able to recover the VI posterior than the synchronous counterparts, due to having more theoretical guarantees. For the asynchronous approach, it can even be observed that it is better than the other approaches at uncertainty quantification, however this is most likely due to being less confident in it's predictions, since in total, less data was observed and the effect incorporated into the approximate likelihood terms.



(a) Sequential

(b) Synchronous

(c) Asynchronous

Figure 3.9: Update Schedules of PVI vs VI posterior.

Chapter 4

Partitioned Generalised Variational Inference

Since PVI is a Bayesian approach to federated learning and tacitly assumes that the likelihood and the prior are well specified, implies that it is not robust to model misspecification. We therefore develop a novel federated learning algorithm we term Partitioned Generalised Variational Inference that is robust to model misspecification and addresses drawbacks of PVI. We build our algorithm on the Partitioned Variational Inference framework of Bui et al. (2018) and the robust Generalised Variational Inference approach of Knoblauch et al. (2022).

Chapter 4.1 We discuss Generalised Variational Inference as developed in Knoblauch et al. (2022) and how this global approach can address model misspecification in traditional VI through considering arbitrary loss functions and divergence measures.

Chapter 4.2 proposes Partitioned Generalised Variational Inference, a novel approach for robust, probabilistic federated learning, which we discuss in detail, showing modularity of PGVI, and examines the relationship between frequentist FL and PGVI.

Chapter 4.3 We explore how we address model misspecification through PGVI, by considering robust divergence measures and robust loss functions. In particular we consider the Alpha–Rényi divergence and the Beta-loss function.

Chapter 4.4 We implement PGVI on the Gaussian Mixture Model we defined in the chapter 3.6. We show how this can be done without largely changing the existing PVI code base through deriving closed form solutions for divergence measures and loss functions.

Chapter 4.5 We demonstrate the empirical effectiveness of PGVI on similar toy data sets as considered in the PVI counterpart. We show how PGVI compares against traditional PVI when the model is correctly specified and when it is misspecified, showing robustness to outliers, more desirable uncertainty quantification, and verify modularity of PGVI.

4.1 Generalised Variational Inference

Introduced in Knoblauch et al. (2022), Generalised Variational Inference (GVI) is a framework that improves upon challenges in traditional VI by extending on the optimisation centric view of Bayes' Theorem, Csiszar (1975) and Zellner (1988). We consider generalised Bayesian posterior distributions as considered in Bissiri et al. (2016) and theoretically developed by Miller (2021) who develops excellent theory about concentration, asymptotic normality, and coverage for posteriors of the form

$$q_B^*(\theta) = \exp(-\ell_n(\theta; x_{1:n}))\pi(\theta)/z_n \tag{4.1}$$

where $z_n = \int_\Theta \exp(-\ell_n(\theta))d\pi(\theta)$. We denote $\ell_n : \Theta \times \mathcal{X} \to \mathbb{R}$ to be an arbitrary loss function for our data, and we recover the standard Bayesian posterior for $\ell_n(\theta; x_{1:n}) = -\log p_n(x_{1:n}|\theta)$. Bissiri et al. (2016) further develop the work of Zellner (1988) to the optimisation centric view of generalised posteriors.

$$q_B^*(\theta) = \underset{q(\theta)\in\mathcal{P}(\Theta)}{\arg\min} \Big\{ \mathbb{E}_{q(\theta)}\big[\ell_n(\theta; x_{1:n})\big] + \mathrm{KL}(q(\theta)||\pi(\theta)) \Big\} \tag{4.2}$$

However, other Discrepancy Variational Inference methods still outperform these general posteriors empirically when we restrict ourselves to optimise over $\mathcal{Q} \subset \mathcal{P}(\Theta)$, even though vanilla VI is optimal with respect to equation 4.2, as shown in Knoblauch et al. (2022). This is due to attractive properties of such DVI methods (Minka, 2001b; Li and Turner, 2016; Hasenclever et al., 2017; Futami et al., 2018; Jewson et al., 2018; Medina et al., 2022) with respect to the choice of $\mathcal{Q}$, the robustness to misspecification and outliers, and information geometric properties of different divergence measures.

4.1.1 Model Misspecification

The $\mathcal{M}$–closed assumption, that models are correctly specified and that there exists some parameter $\theta^* \in \Theta$ that generated the data, and which we can find; however, the $\mathcal{M}$–open assumption acknowledges that in most applications, this isn't the case and this θ^* might not exist at all, Bernardo and Smith (2000). Jewson et al. (2018) defines the data generating process as some true belief a decision maker can find about the data distribution given infinite time, observations, and computational power, and since this isn't possible we are required to be in the $\mathcal{M}$–open framework.

Since there exists no parameter, $\theta^* \notin \Theta$, that explains the DGP rigorously, we aim to find some parameter $\theta_0 \in \Theta$—not necessarily unique in general, as pointed out in Kleijn and van der Vaart (2006)—that is closest to this true distribution in the sense of some divergence, Walker (2013). Jewson et al. (2018) further points out that even though we now know that any model we could come up with is likely incorrect, it can constitute a best approximation about what a decision maker (DM) wants to learn about. Hence, adopting such a parametric approach where we learn about some parameter θ_0 is still a reasonable thing to do, especially since Bayes' theorem can find a parameter which is closest to the true DGP, Berk (1966), Kleijn and van der Vaart (2006) and Shalizi (2009).

For Generalised Variational Inference, model misspecification is primarily concerned with: (1) prior misspecification, where our prior beliefs about the model parameter are incorrect, (2) likelihood misspecification, where the likelihood function represents a subjective belief about the distribution that could have generated the data, and (3) that we have computational power to find a posterior belief given some data, Knoblauch et al. (2022). Clearly, this is the case in most models, since we rarely are able to know the explicit likelihood function, priors are often uninformative (Murphy, 2023) and do not reflect any belief, and we have limited computational power and time.

4.1.2 The Rule of Three and Generalised Variational Inference

Under model misspecification, the need to move away from the KL divergence becomes explicit when examining the empirical performance of DVI posteriors, and when considering the properties of the KL divergence. Especially the non–robustness to tail distributions, for instance in epsilon contaminated models, as explored in Jewson et al. (2018), encodes explicit performance beliefs a DM would want to model. In Generalised Variational Inference (GVI) we combine the ideas of Discrepancy VI with those of generalised posteriors to become robust to model misspecification and develop, as a result, favourable behaviour of posterior distributions, such as better uncertainty quantification.

Knoblauch et al. (2022) propose an axiomatic view of Bayesian inference, in which the KL divergence in equation 4.2 is replaced by arbitrary, robust divergence measures. The Rule of Three (RoT) quantifies some belief about the distribution after observing some data which satisfies a representation theorem and recovers the generalised Bayesian posterior.

Definition 17 (Rule of Three) *For a space $H \subseteq \mathcal{P}(\Theta)$, data $x_{1:n}$, prior distribution $\pi(\theta)$, loss function $\ell_n(\theta; x_{1:n})$, and divergence measure $D(\cdot||\cdot)$, the rule of three posterior is*

$$P(\ell, D, H) := \underset{q(\theta)\in H}{\arg\min} \Big\{ \mathbb{E}_{q(\theta)}\big[\ell_n(\theta; x_{1:n})\big] + D(q(\theta)||\pi(\theta)) \Big\} = q^*(\theta)$$

Through the RoT, we can derive GVI by restricting ourselves to a strict subspace of $\mathcal{P}(\Theta)$ which is represented as some variational family that parametrises distributions based on some variational parameters, i.e. $\mathcal{Q} = \{q(\theta|\kappa) : \kappa \in K\} \subset \mathcal{P}(\Theta)$, where K is a space of variational parameters, see Knoblauch et al. (2022) and Knoblauch (2019). Then, if the loss function is furthermore assumed to be additive—which is ideal for computation—we can write the GVI posterior as

$$q^*_{GVI}(\theta) = P(\ell, D, \mathcal{Q}) = \underset{q(\theta)\in\mathcal{Q}}{\arg\min} \Big\{ \mathbb{E}_{q(\theta)}\big[\sum_{i=1}^{n} \ell_i(\theta; x_i)\big] + D(q(\theta)||\pi(\theta)) \Big\} \tag{4.3}$$

Similarly as in theorem 12, the expected loss under our approximation defines the parameter of interest which we would like to find, while the divergence quantifies how much uncertainty we want to place in this estimate. In fact, without a divergence, $D = 0$, the GVI posterior is the maximum likelihood estimator, Knoblauch et al. (2022). The space $\mathcal{Q}$ we are optimising over determines the computational complexity of our posterior.

4.1.3 Addressing Model Misspecification through GVI

Generalised Variational Inference is able to address model misspecification through the arguments of the RoT. In particular Knoblauch et al. (2022) shows that changing the loss function directly affects model misspecification through alleviating likelihood misspecification. This is in particular interesting since this is the main challenge of $\mathcal{M}$–open inference. Furthermore, by changing the divergence we can mitigate prior misspecification, as well as undesirable uncertainty quantification. The divergence chosen then affects how much we are able to deviate from the prior distribution during inference, and how certain we can be in the distribution of the parameter of interest. We do this by exploiting the desirable properties of divergence measures through their information geometric properties.

However this also means that we are not inherently interested in finding an approximation to the Bayesian posterior, but rather a distribution that is of interest to a DM. We could therefore be interested in modelling out noise from a distribution, or our aim can be to capture this noise distribution explicitly. In general, a DM needs to understand what their aim is when choosing loss functions and divergence measures.

4.1.4 Convergence of Generalised Posteriors under Misspecification

We have already discussed the convergence of traditional Bayesian posteriors under model misspecification based on the work of Berk (1966) which is the first result of posterior convergence, Kleijn and van der Vaart (2012) who give asymptotic normality results for misspecified models, and most notably Shalizi (2009) who extends posterior convergence to rates of convergence and non–i.i.d. data, where likelihoods depend on all previously seen data points.

Miller (2021) extends the notion of generalised posteriors, as introduced earlier, to not require there to be any data, using the measure theoretic style of integration as defined by capital letters over the specified domain, as follows

Definition 18 *Let $(\Theta, \mathcal{A}, \Pi)$ be an arbitrary probability measure space and let $f_n : \Theta \to \mathbb{R}$ be a sequence of functions, such that $z_n = \int_\Theta \exp(-n f_n(\theta))\Pi(d\theta) < \infty$ for all $n \in \mathbb{N}$, then the generalised posterior measure is given by*

$$\Pi_n(d\theta) = \exp(-n f_n(\theta))\Pi(d\theta)/z_n$$

we take $d\theta$ to mean an infinitesimally small neighbourhood of θ. Using this as a starting assumption, Miller (2021) then shows that, under some further assumptions discussed explicitly in Miller (2021), the generalised posterior concentrates around some minimising parameter $\theta_0 \in \Theta$, not necessarily the data generating parameter. Furthermore, he shows that the sequence of posteriors $\Pi_n(d\theta)$ is asymptotically normal, extending the work of Kleijn and van der Vaart (2012), and that they provide sufficient coverage which allows for uncertainty quantification. However, although the concentration and asymptotic normality results apply to misspecified models as well, in the misspecified setting Miller (2021) points out that coverage results for generalised posteriors need not hold in general.

Knoblauch (2019) further develops frequentist consistency results for Generalised Variational Inference posteriors based on the notion of Gamma–convergence, see Braides (2002), and gives an outline strategy of proving frequentist consistency for specific GVI posteriors. However, the assumptions are somewhat more restrictive than those considered in Miller (2021). Consistency of GVI posteriors stipulates that under the assumptions of Knoblauch (2019) as $n \to \infty$ the GVI posterior converges to the Dirac–delta distribution, i.e. it converges to the optimal parameter θ_0 in probability, with stronger claims being made for independent data or restricted variational families. However, Bernstein–von Mises results are not established for GVI posteriors, which would ensure that these are asymptotically normal.

4.2 Partitioned Generalised Variational Inference

We extend the Generalised Variational Inference framework to distributed settings through the Partitioned Variational Inference approach. We assume that the approximate posterior distribution decomposes as in PVI to include approximate likelihood terms, which appear necessary for federated learning, as used in other probabilistic models such as in Mekkaoui et al. (2021) for federated SGLD. The generalised posterior distribution that we target should therefore be of the form:

$$q_B^*(\theta) \propto \prod_{m=1}^{M} \exp(-\ell_m(\theta; \boldsymbol{x}_m))\pi(\theta) \tag{4.4}$$

This implicitly assumes that the loss function is additive across clients, and that the data is independent across clients. For PGVI we want to approximate this posterior similarly to PVI by using approximate loss terms $t_m(\theta) \approx \exp(-\ell_m(\theta; \boldsymbol{x}_m))$ such that these form an approximation $q(\theta) \propto \pi(\theta) \prod_{m=1}^{M} t_m(\theta)$, which is a generalised posterior. Then, we can replace the standard log likelihood in PVI with the exponential loss term and derive an equivalent local update with generalised posteriors as

$$q_m(\theta) = \underset{q(\theta)\in\mathcal{Q}}{\arg\min} \Big\{ \mathbb{E}_{q(\theta)}\big[\ell_m(\theta; \boldsymbol{x}_m)\big] + \mathrm{KL}(q(\theta)||q^{\setminus m}(\theta)) \Big\}$$

Furthermore, to derive a local Generalised Variational Inference update, we move away from the KL divergence, due to the reasons elaborated on previously, and towards general divergence measures. This then yields the local update equation of Partitioned GVI.

$$q_m(\theta) = \underset{q(\theta)\in\mathcal{Q}}{\arg\min} \Big\{ \mathbb{E}_{q(\theta)}\big[\ell_m(\theta; \boldsymbol{x}_m)\big] + D(q(\theta)||q^{\setminus m}(\theta)) \Big\} \tag{4.5}$$

Each local update is therefore a GVI posterior that uses the cavity distribution as a local prior and is required to have the form of $q(\theta)$, where the approximate loss terms are multiplicative. For the algorithm of PGVI we can replace equation 3.4 with equation 4.5 in algorithm 7 to derive our novel algorithm 9. Note that the server steps for algorithm 8 and 6 are equivalent, and PGVI changes the way clients compute updates (algorithm 9).

Algorithm 8 Partitioned Generalised Variational Inference, Server (Our approach)

Server:

Inputs: M clients, prior distribution $\pi(\theta)$, divergence D, loss function $\ell(\cdot;\boldsymbol{x})$, and variational family $\mathcal{Q}$

Initialise: $t_m^{(0)}(\theta) = 1, \forall m \in [M] \quad \implies \quad q^{(0)}(\theta) = \pi(\theta)$

until convergence for iteration $i = 1, 2, ...$ **do**
  For client m in update schedule $\mathcal{B}_i$ **do in parallel**
    $\Delta_m^{(i)}(\theta) \leftarrow \text{ClientUpdate}(q^{(i-1)}(\theta), m, \mathcal{Q})$
  end for
  # Aggregate Client updates:

$$q^{(i)}(\theta) \propto q^{(i-1)}(\theta) \prod_{m \in \mathcal{B}_i} \Delta_m^{(i)}(\theta)$$

Return: $q(\theta)$

Algorithm 9 Partitioned Generalised Variational Inference, Client Update (Our approach)

ClientUpdate$(q^{(i-1)}(\theta), m, \mathcal{Q})$**:**

Inputs: Current approximation $q^{(i-1)}(\theta)$, divergence measure D, loss function $\ell_m(\cdot;\boldsymbol{x})$ and variational family $\mathcal{Q}$

Compute Cavity Distribution:

$$q^{\setminus m}(\theta) \propto \frac{q^{(i-1)}(\theta)}{t_m^{(i-1)}(\theta)}$$

Compute local approximation:

$$q_m^{(i)}(\theta) \leftarrow \underset{q(\theta)\in\mathcal{Q}}{\arg\min} \Big\{ \mathbb{E}_{q(\theta)}\big[\ell_m(\theta;\boldsymbol{x}_m)\big] + D\big(q(\theta)\big|\big|q^{\setminus m}(\theta)\big)\Big\} \tag{4.6}$$

Update local approximate likelihood:

$$t_m^{(i)}(\theta) \propto \frac{q_m^{(i)}(\theta)}{q^{\setminus m}(\theta)}$$

Return: $\Delta_m^{(i)}(\theta) := \frac{t_m^{(i)}(\theta)}{t_m^{(i-1)}(\theta)}$ to server

RoT modularity of GVI (Knoblauch et al., 2022) directly applies to PGVI and hence we can derive an equivalent result for PGVI.

Theorem 19 (PGVI modularity) *Fix $\boldsymbol{x}_{1:M}$, M, n_m, $\pi(\theta)$, $\mathcal{Q}$, and the update schedule and aggregation at the server. Further, let $P(\ell, D, \mathcal{Q}) = q_1^*(\theta) \in \mathcal{Q}$. Then, if a DM wants to produce a different posterior $q_2^*(\theta) \in \mathcal{Q}$ through the RoT, then*

1. *Robustness to model misspecification is achieved by changing the loss function ℓ.*

2. *Robustness to prior and cavity misspecification, without changing the target parameter $\theta_0 \in \Theta$, is achieved through changing the divergence D.*

3. *Desirable uncertainty quantification is achieved by changing the divergence D, without changing what we are targeting.*

This result is a direct consequence of applying the GVI updates locally and keeping the server aggregation and updating schedule fixed. Note that changing the likelihood function changes our target parameter $\theta_0 \in \Theta$, since modelling out outliers directly influences this.

PGVI also relates to FedAvg if we discount the divergence term, i.e. $D = 0$. Then doing Monte Carlo integration of the expectation term is exactly the Stochastic Gradient Descent objective. And if we then take a regulated amount of steps for when we compute the local approximation, the output will be a Dirac–delta distribution, which places a point mass at a single point (coinciding with the frequentist MLE at the limit) and is 0 everywhere else. Communicating the change in weights instead of the change in approximate likelihoods to the server would result in identical behaviour to FedAvg, provided the server aggregates these updates as in FedAvg. Regardless, the main objective of PGVI can be adapted to get an analogous frequentist federated learning algorithm. This relationship also shows that we should not necessarily train a local posterior to convergence, since this might cause over–concentration on the local data set as in FedAvg (McMahan et al., 2017), but rather for only a number of iterations for the optimiser we choose.

4.2.1 Cavity Distribution as Local Prior

In PGVI we use the cavity distribution as a local prior distribution, where the cavity distribution of some client m acts as an approximation to

$$q^{\setminus m}(\theta) \approx \pi(\theta) \prod_{k \neq m} \exp(-\ell_k(\theta; \boldsymbol{x}_k))/\hat{z}_m$$

where $\hat{z}_m$ is a local normalising constant.

In EP and PVI we have implicitly assumed that the approximating likelihoods are some, not necessarily normalised, probability distributions. However, since we model exponentials of loss functions, this isn't necessarily required, as long as the approximate loss term of PGVI can be written as some exponential, which encompasses the approximate likelihood terms of PVI. Therefore, the global PGVI posterior is some generalised posterior where the approximate loss terms are

$$t_m(\theta) = \exp(-\tilde{\ell}_m(\theta))$$

with $\tilde{\ell}_m(\theta)$ modelling the loss of a client.

Then the cavity distribution can be found through

$$q^{\setminus m}(\theta) \propto \pi(\theta) \exp\Big\{\tilde{\ell}_m(\theta) - \sum_{k \neq m} \tilde{\ell}_k(\theta)\Big\}$$

This is an application of the log–sum–exp property (Murphy, 2023), and is equivalent to the way we have been updating it in PVI.

Furthermore, since the cavity distribution will likely be misspecified due to approximating the other local loss terms at a local update, PGVI can use robust divergences to counteract local prior misspecification through the cavity distribution. This allows us to use potentially bad initial estimates for the data likelihoods of other clients without seeing them directly in federated settings. Furthermore, this can be able to counteract heterogeneous client data, since we can determine how much we want to disagree with the prior distribution, or the cavity distribution, for local updates and how much weight we want to place onto these updates at some server.

4.3 Addressing Model Misspecification in Federated Learning

In order to address misspecification of prior, or cavity distributions, as well as likelihood functions, we can implement different robust divergence measures and loss functions, depending on what a DM wants to model. Changing the loss function addresses model misspecification, and changing the divergence addresses prior distribution misspecification as well as uncertainty quantification.

4.3.1 Robust Divergence Measures

Robust divergence measures allow us to address wrongly specified prior distributions and cavity distributions that are skewed towards different data under heterogeneity. In our experiments of PGVI we consider the Alpha–Rényi divergence, using the parametrisation of Cichocki and Amari (2010) and Knoblauch et al. (2022).

Definition 20 (Alpha–Rényi divergence) *The α–Rényi divergence between two probability distributions $q(\theta)$ and $\pi(\theta)$ for $\alpha \in \mathbb{R}\backslash\{0,1\}$ is given by*

$$D_{AR}^{(\alpha)}(q(\theta)||\pi(\theta)) = \frac{1}{\alpha(\alpha-1)} \log \int_{\Theta} q(\theta)^{\alpha}\, \pi(\theta)^{1-\alpha} d\mu(\theta)$$

This also shows why this distribution is more robust to cavity distribution misspecification, since the hyper–parameter α directly influences how much weight is placed on the cavity distribution in comparison to the distribution we are optimising over. The $D_{AR}^{(\alpha)}$ recovers the Kullback–Leibler divergence as $\alpha \to 1$ and the reverse KL divergence as $\alpha \to 0$, Amari (2016). Therefore, Knoblauch et al. (2022) suggests picking $\alpha \in [0,1]$ is a sensible choice in order to have a trade–off between underestimating the variance, as in VI, and over–fitting the variance as in the zero–avoiding reverse KL approach.

$D_{AR}^{(\alpha)}$ has also been considered as a DVI method in Li and Turner (2016), as well as in power EP of Minka (2004), where the KL in vanilla EP is replaced by this divergence. However, we remark that this is distinctly different to PGVI—analogous to the relationship between DVI and GVI—since power EP assumes that the α–Rényi divergence minimises a distribution to both the cavity distribution and local likelihood function, and does not decompose into an expectation term and an D_{AR}^{α} term solely to the cavity distribution.

Prior misspecification is especially prevalent in Bayesian Neural Networks (BNNs), where a single multivariate Gaussian distribution is placed over all parameter weights, Knoblauch et al. (2022). And we have established that the KL divergence sometimes does not regularise enough under heterogeneous data distributions, as can be seen when performing classification with BNNs where each client contains data of only a single label (Ashman et al., 2022) and in sequential updating the PVI posterior is highly skewed towards the last seen client. Therefore, choosing an α such that we cover more than a single mode of the underlying distribution with an approximation, can be beneficial.

There are many other divergences that we could use instead, such as the family of $\alpha\beta\gamma$–divergences of Cichocki and Amari (2010), which includes the α–divergence, of which the Rényi divergence is a parametrisation, as well as the robust (to outliers) beta and gamma divergences (Futami et al., 2018; Jewson et al., 2018). We give here a definition of the beta divergence, which is used to derive robust losses in the next section (Futami et al., 2018; Knoblauch et al., 2022).

Theorem 21 (Beta Divergence) *The β–divergence between two probability distributions $q(\theta)$ and $\pi(\theta)$ is given by*

$$D_B^{(\beta)}(q(\theta)||\pi(\theta)) = \frac{1}{\beta(\beta-1)}\int q(\theta)^\beta d\mu(\theta) + \frac{1}{\beta}\int \pi(\theta)^\beta d\mu(\theta) - \frac{1}{\beta-1}\int q(\theta)\pi(\theta)^{\beta-1}d\mu(\theta)$$

This divergence is very robust to outliers and prior misspecification, Futami et al. (2018), and if modelling out noise is the aim of the DM, then choosing this divergence is advisable.

Felekis et al. (2022) considered a range of other divergences with GVI in Gaussian Process Regression for uncertainty quantification of both aleatoric uncertainty (for data uncertainty) and epistemic uncertainty (in the underlying model). Such divergences include the Jenson–Shannon divergence, a symmetric version of the KL divergence, Fisher divergence (or half the Hyvärinen divergence), or the total variation distance. Many more divergences could be used for GVI, which are commonly used in information geometry, Nielsen (2021, 2023); Nielsen and Okamura (2024); Pardo Llorente (2006); Kassab and Simeone (2022).

4.3.2 Robust Loss Functions

Robust loss functions allow us to overcome model misspecified by the modularity of PGVI. This formulation of PGVI is highly similar to finding the optimal parameter θ_0 as in theorem 12, and is in fact similar to what changing the likelihood entails. And while the Kullback–Leibler divergence was chosen in theorem 12, Walker (2013) states that there is an arbitrariness in choosing divergences, and it is not necessary to choose the KL divergence. This means that for robustness, we would like to choose loss functions that are based on divergences of the true data generating process to the misspecified likelihood function, by $D(F^*(\boldsymbol{x})||p_m(\cdot|\theta))$. This targets the optimal value $\theta_0 \in \Theta$. And depending on the aim of a DM, we can be robust to model misspecification, by changing what divergence we use for this loss. Nevertheless, in PGVI we are limited to the loss functions, that are additive. And although there are quite a few losses that are based on divergences, see Knoblauch et al. (2022, Table 2), only a few are additive.

These additive losses include the Beta and Gamma loss functions, based on the robust divergences of the same name.

Definition 22 (Beta Loss function) *The β–loss function of some (misspecified) likelihood function $p_m(\boldsymbol{x}_m|\theta)$ is given by*

$$\mathcal{L}_p^{(\beta)}(\theta, \boldsymbol{x}_m) = -\frac{1}{\beta - 1}(p_m(\boldsymbol{x}_m|\theta))^{\beta-1} + \frac{\int_{\mathcal{X}} p(\boldsymbol{y}|\theta)^\beta d\boldsymbol{y}}{\beta}$$

We can also use the gamma loss, is based on the gamma divergence, which is a transformation of the beta divergence. Specifically by the component–wise transformation (Knoblauch et al., 2022)

$$c_0 \int q(\theta)^{c_1} \pi(\theta)^{c_2} d\mu(\theta) \to c_0 \log \int q(\theta)^{c_1} \pi(\theta)^{c_2} d\mu(\theta)$$

The gamma loss, derived from the transformed beta divergence, is defined as follows.

Definition 23 (Gamma Loss function) *The γ–loss function of some (misspecified) likelihood function $p_m(\boldsymbol{x}_m|\theta)$ is given by*

$$\mathcal{L}_p^{(\gamma)}(\theta, \boldsymbol{x}_m) = -\frac{1}{\gamma - 1}(p_m(\boldsymbol{x}_m|\theta))^{\gamma-1} \cdot \frac{\gamma}{\left(\int_{\mathcal{X}} p(\boldsymbol{y}|\theta)^\gamma d\boldsymbol{y}\right)^{\frac{\gamma-1}{\gamma}}}$$

These losses also recover the log likelihood as $\beta = \gamma \to 1$, and choosing $\beta, \gamma > 1$ results in robustness to outliers. Knoblauch et al. (2022) suggest choosing $\beta = 1 + \epsilon$ for $\epsilon > 0$ results in a nice trade–off between alleviating model misspecification and being close to the log likelihood, since these losses work less well if the likelihood is correct. The robustness of these losses comes from the weight we place upon the prior, or in this case the misspecified likelihood, through the hyper–parameter β ($/\gamma$). In the experiments we will consider $\mathcal{L}^{(\beta)}$.

4.4 Implementing Partitioned Generalised Variational Inference

Considering the same experimental set–up as in chapter 3.7 for PVI, we only need to change the way in which we compute the divergence and loss functions. In the experiments we consider mostly the Alpha–Rényi divergence as a robust divergence to prior misspecification, and which improves uncertainty quantification in comparison to PVI. We can find this in closed form for our model, assuming that the approximate posterior is, the same as with PVI, a (multivariate) Gaussian distribution.

Lemma 24 (Closed form of $D_{AR}^{(\alpha)}$) *The Alpha–Rényi divergence between two multivariate Gaussian distributions, $q(\theta) \sim \mathcal{N}(\mu, \Sigma)$ and $q^{\setminus m}(\theta) \sim \mathcal{N}(\nu, \Lambda)$ is given by*

$$\begin{aligned}\mathrm{D}_{AR}^{(\alpha)}(q(\theta)||q^{\setminus m}(\theta)) = &\frac{(\mu - \nu)^\top (\alpha\Lambda + (1-\alpha)\Sigma)^{-1}(\mu - \nu)}{2} \\ &- \frac{1}{2\alpha(\alpha - 1)} \log \frac{|\alpha\Lambda + (1-\alpha)\Sigma|}{|\Sigma|^{1-\alpha}\,|\Lambda|^\alpha}\end{aligned}$$

Proof See appendix C for a proof of this. ■

We further consider the Beta Loss function, where we assume that we do not know the noise distribution, and instead only assume that the data is generated from a multivariate Gaussian distribution, hence our likelihood is misspecified. Then $\mathcal{L}_p^{(\beta)}$ can also be computed as a closed form.

Lemma 25 (Closed form of $\mathcal{L}_p^{(\beta)}$) *The beta loss function of a misspecified multivariate Gaussian likelihood function, $p_m(\boldsymbol{x}_m|\theta) \sim \mathcal{N}(\boldsymbol{x}|\theta, \Sigma)$, for Monte Carlo sample θ_s is given by*

$$\mathcal{L}_{p_m}^{(\beta)}(\theta_s; \boldsymbol{x}_m) = -\frac{1}{\beta - 1} p_m(\boldsymbol{x}_m|\theta_s)^{\beta-1} + \frac{1}{(2\pi)^{(D/2)(\beta-1)} \det(\Sigma)^{(\beta-1)/2} \beta^{(D+2)/2}}$$

Proof See appendix C for a proof of this. ■

The latter part of this does not depend on θ and hence we do not require it if we do not also want to estimate the covariance matrix of the likelihood function, however we assume throughout the experiments in the next section that the true covariance matrix is known.

Since the beta loss function is additive we can approximate the expectation term using Monte Carlo integration as

$$\mathbb{E}_{q(\theta)}\big[\mathcal{L}(\theta; \boldsymbol{x}_m)\big] \approx \frac{1}{S} \sum_{s=1}^{S} \mathcal{L}_{p_m}^{(\beta)}(\theta_s; \boldsymbol{x}_m) \,, \quad \{\theta_s\}_{s=1}^{S} \sim q(\theta)$$

4.4.1 Influence Functions

To compare the robustness of Divergences or Loss functions to outliers, we can compute Influence functions, Jewson et al. (2018) and Knoblauch et al. (2022). To do this, we compute a base PGVI posterior of $x_{1:n}$ observations, and compare this with the PGVI posteriors after $x_{1:n+1}$ observations, where the $n{+}1^{\text{st}}$ observation is some distance to the true mean. We can then compare the influence of these observations by computing the Fisher–Rao distance between these two distributions, and plot these influences based different x_{n+1} observations that are more outliers.

In the experiments, we consider a similar set–up to Jewson et al. (2018) in which we sample 100 data points from a one–dimensional Student-t distribution with 4 degrees of freedom. We further assume that the DGP has 0 mean and variance 1, and that this is misspecified where we assume it is a normal distribution, $\mathcal{N}(0, 1^2)$. The 101^{st} observation is then artificially set to be the set $x_{n+1} \in \{2, 4, 6, 8, 10, 12, 14\}$. We approximate the posterior using a univariate normal distribution. The Influence of an observation on posterior $q_2(\theta) \sim \mathcal{N}(\mu_2, \sigma_2^2)$ in comparison to the base posterior without the outlier, $q_1(\theta) \sim \mathcal{N}(\mu_1, \sigma_1^2)$, can then be computed through the Fisher–Rao divergence, which is symmetric, for univariate normal distributions, Nielsen (2023).

$$D_{FR}(q_1(\theta)||q_2(\theta)) = \sqrt{2} \log\Big(\frac{1 + \Delta(\mu_1, \sigma_1, \mu_2, \sigma_2)}{1 - \Delta(\mu_1, \sigma_1, \mu_2, \sigma_2)}\Big)$$

where $\Delta(\mu_1, \sigma_1, \mu_2, \sigma_2)$ is given by:

$$\Delta(\mu_1, \sigma_1, \mu_2, \sigma_2) = \sqrt{\frac{(\mu_2 - \mu_1)^2 + 2(\sigma_2 - \sigma_1)^2}{(\mu_2 - \mu_1)^2 + 2(\sigma_2 + \sigma_1)^2}}$$

Equivalently, it can be expressed as follows, Nielsen (2023).

$$D_{FR}(q_1(\theta)||q_2(\theta)) = 2\sqrt{2}\,\text{arctanh}\big(\Delta(\mu_1, \sigma_1, \mu_2, \sigma_2)\big)$$

4.5 Experiments

We consider several experiments of Gaussian Mixture Models, as in the PVI experiments. For the experiments in chapter 4.5.2, we compare PGVI with the PVI results of chapters 3.7.1 and 3.7.2. All experiments use the Adam optimiser with fixed decay parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as suggested in Kingma and Ba (2015), where we vary the learning rate η for the experiments. Unless otherwise stated we use sequential PGVI (and PVI).

4.5.1 Influence of outliers in PGVI

When investigating the influence of outliers on PGVI posteriors, we can directly verify the modularity of PGVI as stated in theorem 19. When considering outliers with a misspecified likelihood, which we assume to be a single one dimensional Normal distribution, we want to show that model misspecification, i.e. incorrect likelihoods and outliers, can only be overcome through changing the loss function but not the divergence. The results are shown in figure 4.1 for the KL divergence with the log likelihood, the α–Rényi divergence with the log likelihood, and the the α–Rényi divergence with the β–loss function.

In this experiment we generate 100 data points from a student–t distribution with four degrees of freedom, zero mean and variance 1, as suggested in Jewson et al. (2018). We define $\pi(\theta) = \mathcal{N}(1, 2.5)$, and assume that the misspecified likelihood function is the unit normal distribution. We split these data points into two clients each containing 50 data points, which we consider to be the base case to which we calculate the influence of additional observations through the Fisher–Rao divergence. For the outlier as the 101^{st} observation, we add a data point from $\{2, 4, 6, 8, 10, 12, 14\}$ to the data set of the second client. Using a learning rate of $\eta = 0.0025$ we compute the posteriors for each additional data point for each PGVI implementation. We can plot the influence of this additional observation with increasing distance from the true mean, and is shown in figure 4.1, where we would like the PGVI posterior to place less weight on the additional observation when it is more likely to be an outlier.

The figure (4.1) shows that using the negative log likelihood as a loss function results in the influence of an additional observation increasing with it's distance to the true mean, where the distance is measured in standard deviations. Since the likelihood is misspecified, regardless of the outliers, using the β–loss function results in desirable behaviour by placing less weight on observations that are more likely to be outliers. The change in divergence measure by itself does not change the influence to outliers as is expected by theorem 19.

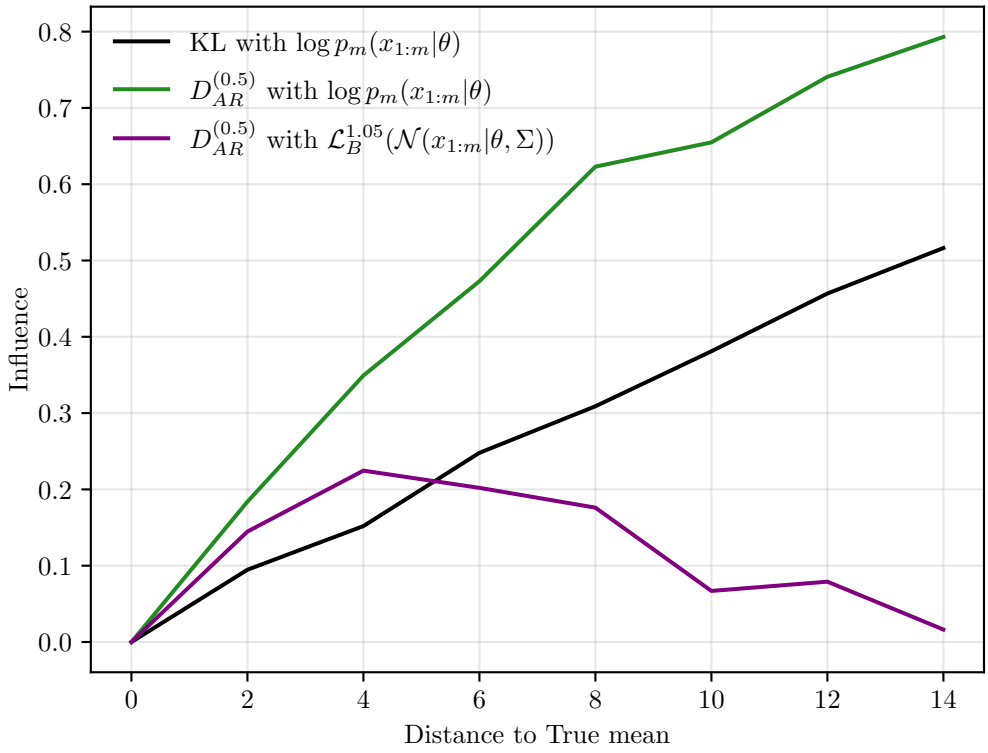


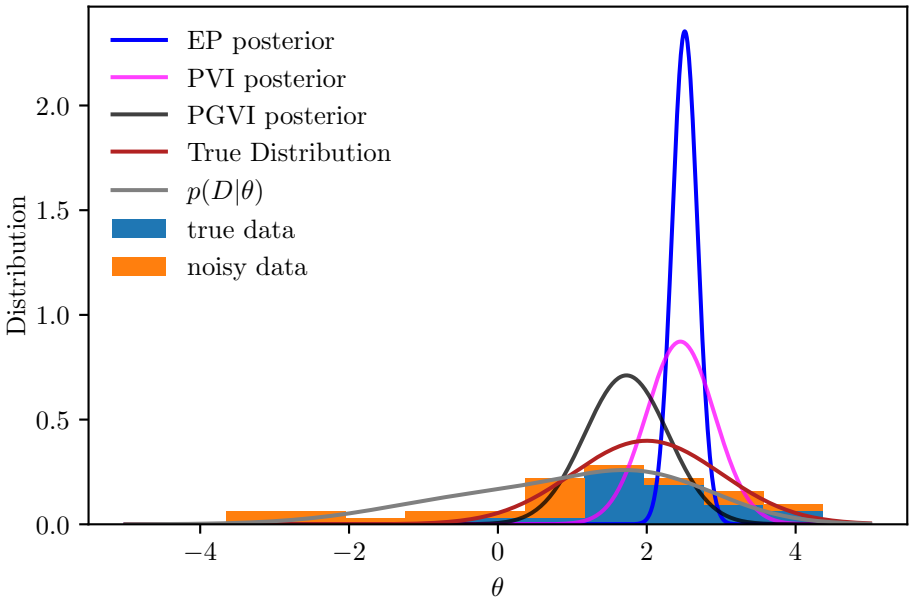
Figure 4.1: Influence function on choosing different divergence measures and loss functions to outliers.

$\mathcal{L}_B^{(\beta)}$ initially places more weight in the additional observation, but after it is determined to be an outlier this effect is reversed. By changing the loss to a robust loss function makes PGVI robust to model misspecification.

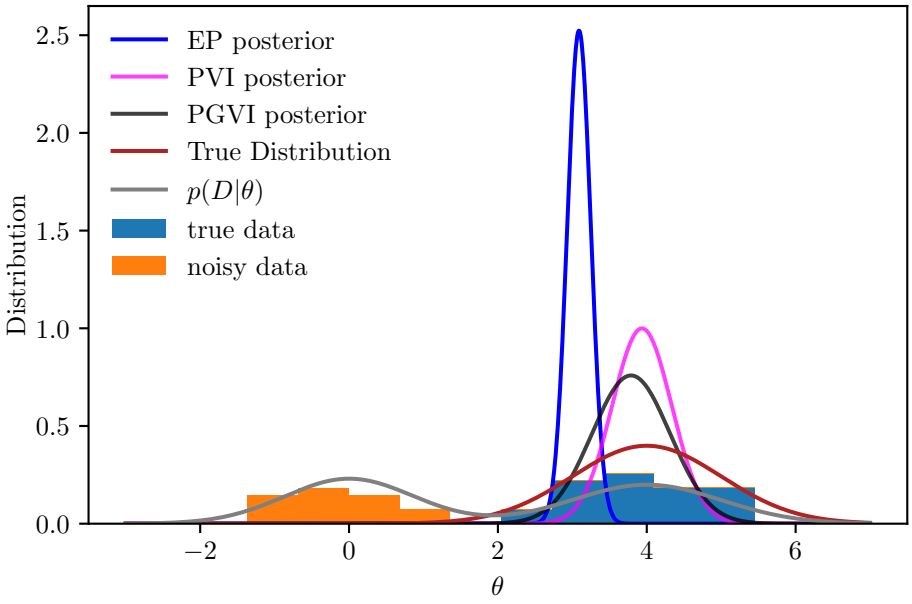
4.5.2 PGVI on Unimodal and Bimodal DGPs

To compare PGVI with PVI and EP we consider the experiments of chapters 3.7.1 and 3.7.2. For PGVI we use the α–Rényi divergence with $\alpha = 0.75$ and the beta loss function with $\beta = 1.5$. Everything else remains unchanged from the respective PVI chapters. For PGVI we would like to observe less weight placed on the tails of the distribution, since we no longer use the KL divergence and less influence placed on outliers, since we use the β–loss function. Furthermore, the α–Rényi divergence should add more uncertainty quantification to the PGVI posterior, which follow by the modularity of PGVI, theorem 19.

Figure 4.2 shows a comparison of these posteriors. We can see that in the bimodal case, the PGVI posterior concentrates around the mean of the true data distribution, even though the prior distribution is centred at 0, and we do not know the noise data mean with the misspecified likelihood for PGVI. Furthermore, the use of the α–Rényi divergence allows for better uncertainty quantification, so that we model the non–noisy DGP better than with PVI.



(a) Unimodal DGP



(b) Bimodal DGP

Figure 4.2: Best viewed in colour. Comparing **PGVI** with PVI and EP on the experiments for unimodal and bimodal DGPs in chapter 3.7. We use the Alpha–Rényi divergence with $\alpha = 0.75$ and the Beta–loss function with $\beta = 1.5$ for a misspecified Normal likelihood.

In the unimodal case, PGVI is unable to correctly differentiate between the noise and true data, since we only know that the distribution is a Gaussian with some variance, which we assume to know. This leads to the PGVI posterior placing less weight on the tails of the unimodal distribution due to the use of the beta loss function and $D_{AR}^{(\alpha)}$. But given that we

do not know the noise distribution in the PGVI scenario, the uncertainty induced by the Rényi divergence makes the PGVI posterior a good approximation.

4.5.3 Number of Clients with PGVI

When comparing the number of clients in PGVI where we keep the loss function, the correctly specified log likelihood, and the divergence, α–Rényi divergence with $\alpha = 0.5$, fixed, then the number of clients does not seem to affect the PGVI posterior distribution. For each client partition, we randomly assign the data points to the clients, where the data is generated by the following likelihood function:

$$p(x|\theta) = \frac{1}{2}\mathcal{N}\Big(\begin{bmatrix}1.5\\2.5\end{bmatrix}, 0.8I_2\Big) + \frac{1}{2}\mathcal{N}\Big(\begin{bmatrix}1\\1\end{bmatrix}, 1.5I_2\Big)$$

We sample 50 data points from this distribution, which has the same likelihood function as the version for PVI, however the data is not the same. The prior distribution is $\pi(\theta) = \mathcal{N}((0,0)^\top, 10I_2)$ and we use the learning rate $\eta = 0.0005$.

We are only fitting an isotropic (spherical) multivariate Gaussian to the model—the covariance matrix is some constant times the identity matrix—since the DGP is a mixture of isotropic Gaussians. This implies that the Frobenius norm of the differences in covariance matrices will be generally very low. However, since the log SGV difference is also low across all client partitions as shown in table 4.1, we can conclude that all covariance matrices are very similar. Furthermore, the table shows that the mean parameters are all very similar, where $\mu_{PGVI} \in \mathbb{B}(\mu_{GVI}, 0.0215)$, i.e. the PGVI mean is in an open ball centred at the GVI mean with radius 0.0215. This can be observed in figure 4.3 where it is hard to differentiate between any two posteriors. This indicates that the modularity result of PGVI could add invariance to client partitions. However, the data is homogeneously partitioned across clients in this example.

4.5.4 Update Schedules with PGVI

When comparing different update schedules on PGVI, where we keep everything fixed but the way we update each client, then we can observe that these lead to relatively similar results to each other and to the GVI posterior. To show this, we define the likelihood function

$$p(x|\theta) = 0.55\mathcal{N}\Big(\begin{bmatrix}2\\3\end{bmatrix}, \begin{bmatrix}2.5 & 2\\2 & 2.7\end{bmatrix}\Big) + 0.45\mathcal{N}\Big(\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}2.2 & -1.5\\-1.5 & 2\end{bmatrix}\Big)$$

from which we sample 50 data points, and split these across 10 clients evenly. We use the α–Rényi divergence with $\alpha = 0.6$, the well specified negative log likelihood as a loss function, and the multivariate Gaussian prior with zero mean and $10I_2$ covariance matrix. The learning rate $\eta = 0.00025$ is fixed across update schedules. Furthermore, we compute the GVI posterior, i.e. PGVI with a single client, for the entire data set to compare the posteriors generated by the update schedules with the GVI posterior. The quantitative results for this are shown in table 4.2.

Table 4.1: Quantitative results for different amount of clients in PGVI posteriors in comparison with the standard GVI posterior distribution, all clients have the same 50 point data set and identical prior conditions.

Number of Clients	Mean difference $\|\|\mu_{GVI} - \mu_{PGVI}\|\|_2$	Covariance matrix difference $\|\|\Sigma_{GVI} - \Sigma_{PGVI}\|\|_F$	Log SGV difference $\|\log\det(\Sigma_{GVI}) - \log\det(\Sigma_{PGVI})\|$
2 Clients	0.0216	**0.0019**	**0.0291**
3 Clients	0.0135	0.0382	0.6687
4 Clients	0.0068	0.0111	0.1720
5 Clients	0.0211	0.0177	0.2819
6 Clients	0.0213	0.0279	0.4648
8 Clients	0.0181	0.0056	0.0856
10 Clients	0.0161	0.0187	0.2987
12 Clients	0.0073	0.0114	0.1773
15 Clients	0.0055	0.0122	0.1903
17 Clients	0.0078	0.0062	0.0946
20 Clients	0.0049	0.0181	0.2892
25 Clients	**0.0047**	0.0042	0.0639
50 Clients	0.0068	0.0127	0.1990

The experiment indicates that PGVI posteriors are similar even under different update schedules. We do not use any damping in these experiments, but for more complex data generating processes or variational families, this might be necessary for PGVI.

Figure 4.4 illustrates how these posteriors compare to the GVI posterior and to the data distribution. Figure 4.4d especially highlights that these posteriors are quite similar in direct comparison, where all three update schedules are overlaid, showing little deviation. This is a desirable property that we would like PGVI to have, i.e. we would like for PGVI posteriors to be invariant to the update schedule chosen, provided all clients contribute. However, we cannot prove this currently.

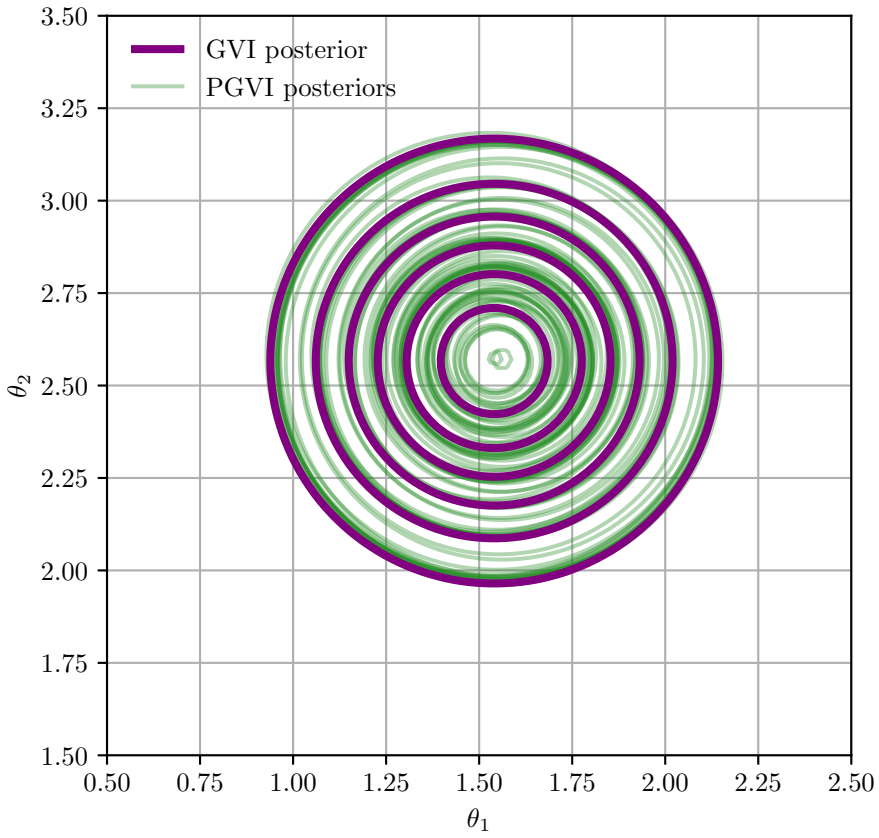
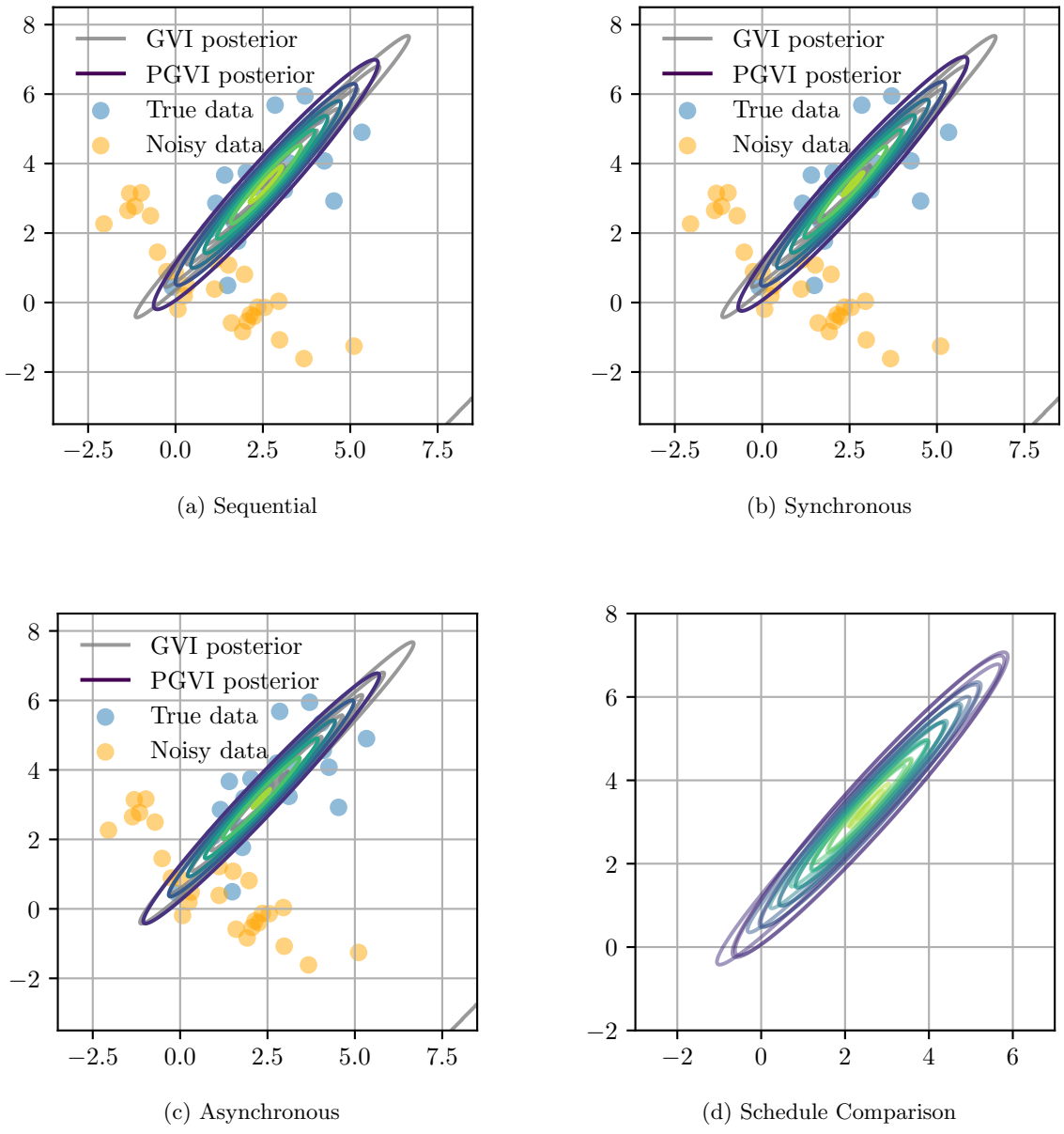


Figure 4.3: Best viewed in colour. PGVI posteriors for different amounts of clients resembles the GVI posterior. We use $D_{AR}^{(0.5)}$ and the correctly specified log likelihood.

Table 4.2: Quantitative results for different update schedules in PGVI posteriors in comparison with the standard GVI posterior distribution, shown in figure 4.4. We use $D_{AR}^{(0.6)}$ and the correctly specified log likelihood.

Update Schedule	Mean difference $\|\|\mu_{GVI} - \mu_{PGVI}\|\|_2$	Covariance matrix difference $\|\|\Sigma_{GVI} - \Sigma_{PGVI}\|\|_F$	Log SGV difference $\|\log\det(\Sigma_{GVI}) - \log\det(\Sigma_{PGVI})\|$
Sequential	0.8275	3.3246	**0.0211**
Synchronous	0.8457	3.0150	0.0727
Asynchronous	**0.4920**	**2.3456**	0.1810



(a) Sequential

(b) Synchronous

(c) Asynchronous

(d) Schedule Comparison

Figure 4.4: Update Schedules of PGVI vs GVI posterior for the Alpha–Rényi Divergence. We set $\alpha = 0.6$ and use the correctly specified log likelihood.

Chapter 5

Federated Learning with Partitioned Generalised Variational Inference

We have already carried out experiments with simulated data sets on Mixtures of Gaussian distributions and have observed desirable performance of PGVI in comparison to PVI. PGVI can be implemented in existing PVI code bases by changing the way we compute the divergence and expectation terms as illustrated in algorithm 8.

Chapter 5.1 We introduce Bayesian Neural Networks and how we can implement PGVI on the PVI code base of Ashman et al. (2022).

Chapter 5.2 shows the results of running PGVI on a number of classification data sets for different federated learning settings.

5.1 Federated Learning of Bayesian Neural Networks

We demonstrate PGVI in deep learning by considering Bayesian Neural Networks. To do this we use the code base of Ashman et al. (2022) where they provide an implementation of PVI on a range of examples, such as PVI as continual (on–line) learning of Sparse Gaussian Processes for binary classification. They also consider training of Bayesian Neural Networks for multi–label classification, in particular Ashman et al. (2022) considers the MNIST data set of 60,000 handwritten and labelled digits (0-9), LeCun et al. (1998). They place a Mean–Field Multivariate Gaussian distribution over the parameters, which is an exponential family distribution with diagonal covariance matrix, such that $\Sigma_{i,j} = 0$ for all $i \neq j$.

We can therefore replace the Kullback–Leibler divergence by other divergence measures, and since the approximation and cavity distributions will therefore be exponential family distributions, we can make use of the closed forms of $\alpha\beta\gamma$–divergences for exponential family distributions, in order to replace the KL divergence, Knoblauch et al. (2022). This includes the Alpha–Rényi divergence, which has a closed form for exponential family distributions of the form

$$q(\theta|\kappa) = h(\theta) \exp\{\eta(\kappa)^\top \phi(\theta) - A(\eta(\kappa))\} \tag{5.1}$$

with $A(\eta(\kappa)) = \log(\int \exp\{\eta(\kappa)^\top \phi(\theta)\} dh(\theta))$. Then, from Knoblauch et al. (2022, Proposition 33 and Corollary 35) it easily follows that the Alpha–Rényi divergence has closed

form.

$$D_{AR}^{(\alpha)}(q(\theta|\kappa_q)||\pi(\theta|\kappa_\pi)) = \frac{1}{\alpha(\alpha-1)}A(\alpha\eta(\kappa_q)+(1-\alpha)\eta(\kappa_\pi)) - \frac{1}{\alpha-1}A(\eta(\kappa_q)) + \frac{1}{\alpha}A(\eta(\kappa_\pi))$$

We can implement this in the code base of Ashman et al. (2022) since they work with exponential family distributions and define the function $A(\cdot)$. See listing C.1 in appendix C.4 for the explicit details of how we implemented this in the existing code base.

In particular we consider the task of classifying images according to some label, such that the input data is the twople $(\boldsymbol{x}_i, y_i)_{i=1}^n$ where the vector (tensor) $\boldsymbol{x}_i$ is some pixel representation of the input image and y_i is it's associated label [1]; we consider this to be the training set of size n. Furthermore, these pixel values take discrete values in some set range, $[0, ..., 255]$, and can be in black and white where these correspond to greyscale, or be in colour and represent tensors with red, green, and blue values at each position. We assume all these values in $\boldsymbol{x}_i$ to be in $[0, 1]$, which we can achieve by dividing all components by the largest possible value, 255. This shows potential differences in scale for the input depending on the colour scheme, i.e. whether the image is monochromatic, or in colour, how many different labels exist, and how high the resolution of the image is. This directly influences the size of the BNN since we require the input dimensions to be the different colour channels for each component of the vector (tensor) of $\boldsymbol{x}_i$, if it is monochromatic then we have an input node for each j corresponding to a row of $\boldsymbol{x}_i$, and if it is in colour then we require this three times for each colour channel, since each row of the tensor contains three values corresponding to red, green, and blue. For the hidden layer—we consider only the case of a single hidden layer—we have a fixed amount of latent dimensions. The output layer of the BNN corresponds to nodes for each possible label.

We consider Bayesian Neural Networks that are fully connected that have an input layer which directly corresponds to the dimension of the inputs; we have a node for each colour of each pixel, where the image size is fixed. Each node is connected to each node in the next layer layer, which is in our case the hidden layer, which we consider to be of size 200 throughout all experiments. Lastly, the output layer is related to the amount of different labels, where we have a node for each possible label. The input layer has ReLU activation function (Murphy, 2023) and the output layer forms a categorical distribution over the output nodes, depending on the distribution we place over the network.

5.2 PGVI on Bayesian Neural Networks

We consider several classification data sets that have different complexities, we summarise the data sets that we use in table 5.1. The training and test sets are independent from each other and are as suggested by the respective data set creators.

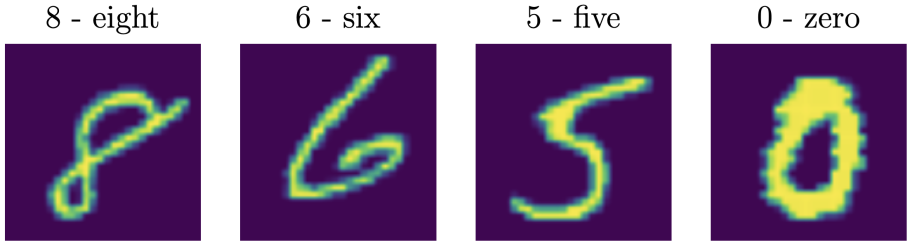
See figure 5.1 for examples from the data sets considered in table 5.1.

Throughout all experiments, the prior distribution is a Multivariate Gaussian distribution, where each component of the mean is chosen uniformly from the interval $[-0.1, 0.1]$.

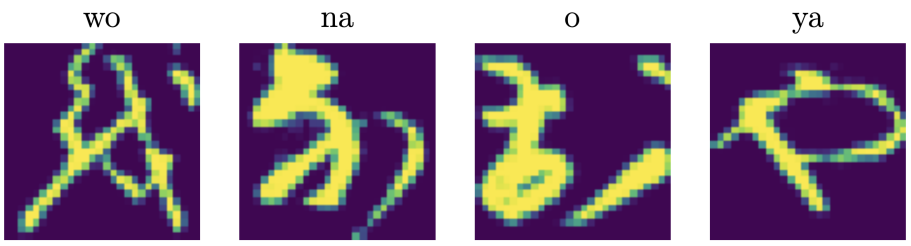
1. We consider only data sets that have correctly labelled images without class intersection, i.e. if an image has some class then it cannot also be in another class; they also do not have missing data.

Table 5.1: Classification data sets considered in the experiments.

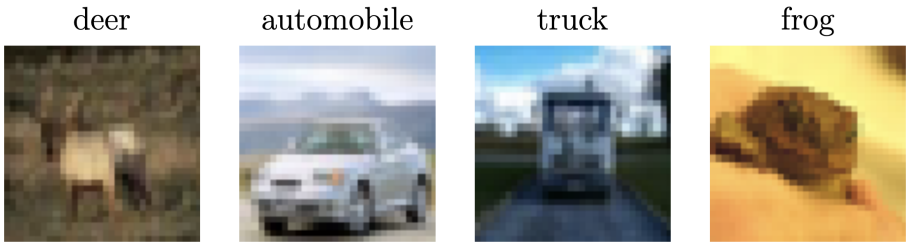
Data set	Task	Input Dimension	Labels	Training set	Test set
MNIST (LeCun et al., 1998)	Handwritten digits	28×28	10	60,000	10,000
Kuzushiji–49 (Clanuwat et al., 2018)	Japanese Kuzushiji	28×28	49	232,365	38,547
CIFAR–10 (Krizhevsky, 2009)	Tiny RGB Images	$32 \times 32 \times 3$	10	50,000	10,000



(a) MNIST (LeCun et al., 1998)



(b) Kuzushiji–49 (Clanuwat et al., 2018)



(c) CIFAR–10 (Krizhevsky, 2009)

Figure 5.1: Best viewed in colour. Example data points taken from the respective data sets.

The covariance matrix is fixed to be diagonal which is initially the identity matrix. The approximation is a Mean Field Multivariate Gaussian distribution and is defined as $q_{MF}(\theta|\mu, \sigma\mathcal{I}_D)$, where μ and σ are some D dimensional vectors, depending on the size of the network.

$$q_{MF}(\theta|\mu, \sigma\mathcal{I}_D) := \prod_{i=1}^{D} q_\mu(\mu_i) \prod_{j=1}^{D} q_\sigma(\sigma_j)$$

This implies that the computation is scalable, since we can compute gradients with respect to each component vector, and the covariance matrix is necessarily positive definite, so we have little numerical instability. The approximating factors are assumed to be normalised Mean–Field Multivariate Gaussians.

We use the Adam optimiser (Kingma and Ba, 2015) where we vary the learning rate η across experiments, but keep the default decay parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, as initialised by PyTorch. Throughout the following experiments, the global VI posterior is the posterior computed, where a single client has all the training data, as in traditional Variational Inference, using the Kullback–Leibler divergence and the negative log likelihood as a loss function. The PVI posterior is the posterior computed over $M > 1$ clients, where we consider the KL divergence and negative log likelihood as well, see equation 3.4. For the PGVI posterior, we consider only client splits with $M > 1$ and the Alpha–Rényi divergence with varying α values as a divergence measure of equation 4.5, and the negative log likelihood as a loss function. The loss function is calculated through Monte Carlo approximation, see chapter 3.6, where we use 100 samples from the current posterior, and mini–batches of data of fixed size chosen randomly from the client data set. We consider mini–batch sizes of 512 for MNIST and Kuzushiji–49, and 256 for CIFAR–10 experiments. We can therefore formulate the local update equation (equation 4.5) for mini–batch size $\mathcal{B}$ for a client with n_m data points as:

$$\begin{aligned}
&\mathbb{E}_{q(\theta)}\Big[-\sum_{i=1}^{n_m}\log p_m(x_i|\theta)\Big] + D_{AR}^{(\alpha)}\big(q(\theta)||q^{\setminus m}(\theta)\big) \\
&\approx \frac{n_m}{\mathcal{B}}\frac{1}{100}\sum_{b=1}^{\mathcal{B}}\sum_{s=1}^{100}\big(-\log p_m(x_b|\theta_s)\big) + D_{AR}^{(\alpha)}\big(q(\theta)||q^{\setminus m}(\theta)\big), \quad \{\theta_s\}_{s=1}^{100} \sim q(\theta).
\end{aligned}$$

The Alpha–Rényi divergence can be calculated explicitly as shown in the previous section. The amount of optimisation epochs for each client varies across experiments, however to prevent over–pruning of weights (Ashman et al., 2022), we stop earlier if the expected log likelihood term does not change in several successive iterations, and therefore the maximum number optimisation epochs that we set was never reached in all experiments. Furthermore, we consider sequential updating, see chapter 3.2.1 for details, throughout the experiments in this chapter.

5.2.1 MNIST handwritten digit classification

The MNIST data set, LeCun et al. (1998), is a commonly used benchmark of classification algorithms, with some achieving 99.98% accuracy. These methods, however rely heavily on the underlying structure and require complex Convolutional Neural Networks (CNN) or Residual Neural Networks (ResNet) to work efficiently. In general, for our experiments we consider fully connected Neural Networks that do not have a complex structure, and hence do not achieve such accuracy, however Bayesian methods are able to quantify uncertainty in a clearly defined sense and hence are well worth the slight accuracy trade–off. The MNIST data set, see figure 5.1a for example data points, is relatively balanced across classes and

we consider homogeneous, random splits of data across 10 clients for the federated settings.

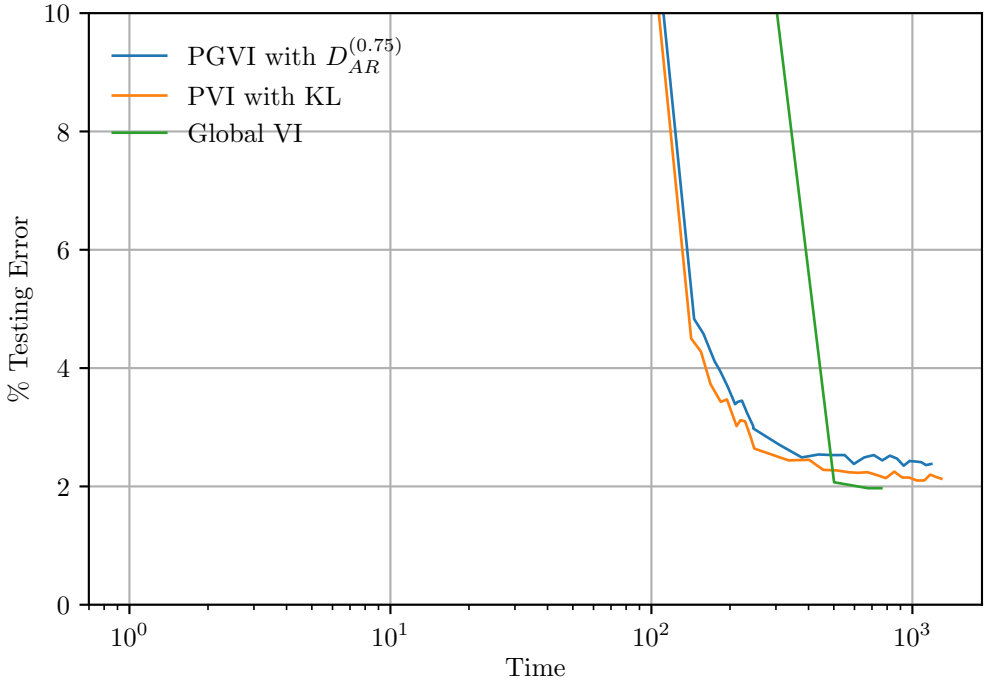


Figure 5.2: Best viewed in colour. PGVI vs PVI on the MNIST classification data set. We use $D_{AR}^{(0.75)}$ as a divergence. The x–axis is in log scale. We consider time in seconds for the total amount of iterations combined.

In Ashman et al. (2022), they consider several implementations of sequential PVI with different learning rates and find that $\eta = 0.002$ works well for PVI, so we can initially consider this learning rate for PGVI as well. As shown in figure 5.2, using an arbitrarily chosen alpha value of 0.75 for PGVI, we can compare, PGVI with PVI and see that it performs similarly to PVI, with both performing just slightly worse than global VI. PVI achieves an accuracy of 97.9% and global, i.e. single client, VI achieves 98.0% which validates proposition 13, that PVI is able to converge to the VI posterior. PGVI achieved only an accuracy of 97.6%, however this close performance to PVI without fine tuning the learning rate for PGVI or considering the effect of α in detail, is already a promising sign that empirically PGVI performs well.

Since PGVI uses a different divergence, we should consider the effect of learning rates on PGVI, as in figure 5.3, since we use a learning rate for PVI that was chosen carefully. Furthermore, we can suppose that we would maybe not want to place as much weight on the prior as before, so we choose $\alpha = 0.5$, keeping the same set–up as before otherwise.

In this case, we vary the learning rate of Adam and we can observe from figure 5.3 that the optimal learning of PVI does not correspond to the optimal learning rate for PGVI, and

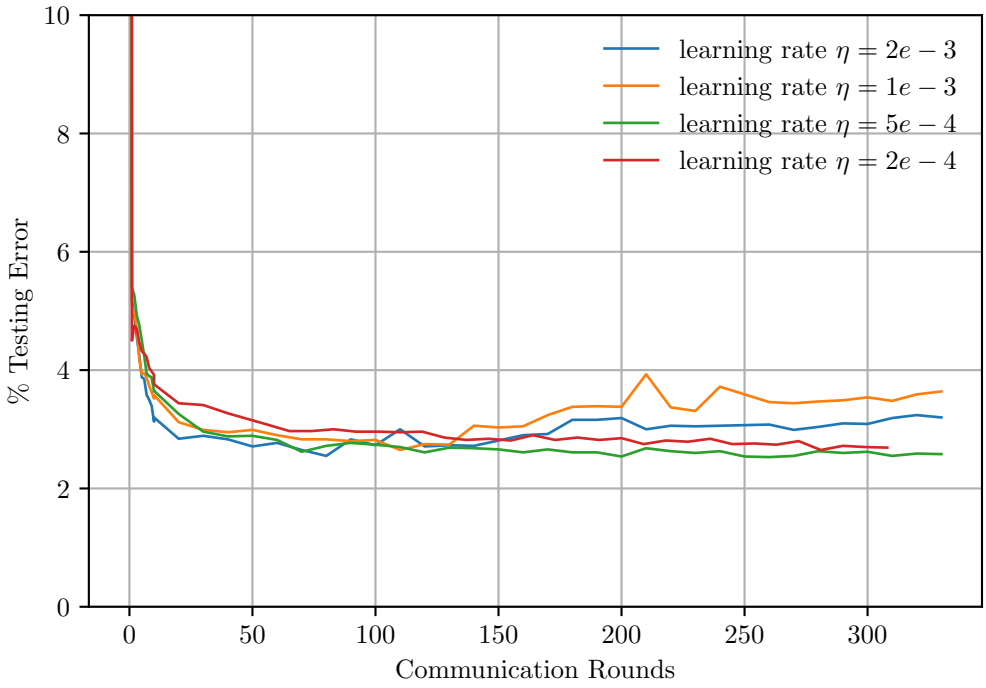


Figure 5.3: Best viewed in colour. PGVI the MNIST classification data set for different learning rates of Adam. We use $D_{AR}^{(0.5)}$ as a divergence. Communication rounds refer to the amounts of updates communicated to the server.

a much lower learning rate of $\eta = 0.0005$ works better. Furthermore, we can observe that placing less weight on the cavity distribution causes a loss in accuracy, as even with better learning rates the best achieves only 97.5% accuracy.

Therefore, we should investigate how the value of α impacts the classification accuracy for PGVI. Fixing the learning rate, as suggested by figure 5.3 as $\eta = 0.0005$, and instead varying the value of α in $D_{AR}^{(\alpha)}(q(\theta)||q^{\setminus m}(\theta))$, while keeping the other conditions the same yields figure 5.4. This figure suggests two things, the first being that, somewhat surprisingly, we achieve better performance with higher values of alpha, than with lower values. However, this optimal α does not tend to infinity, since for $\alpha = 15.0$, this performance increase drastically reverses again. The best performance among the values considered is for $\alpha = 5.0$, and with this, we are able to achieve an accuracy of 98.1% even outperforming traditional Variational Inference where the entire data set is known at a time. This highlights the possibility of PGVI overcoming a major drawback of PVI, namely that PVI can only be as good as Variational Inference.

5.2.2 Kuzushiji–49 Japanese Kuzushiji classification

The Japanese written language is more complex than those using the Latin alphabet, since it includes far more basic characters (hiragana) and complex characters (kanji) than the

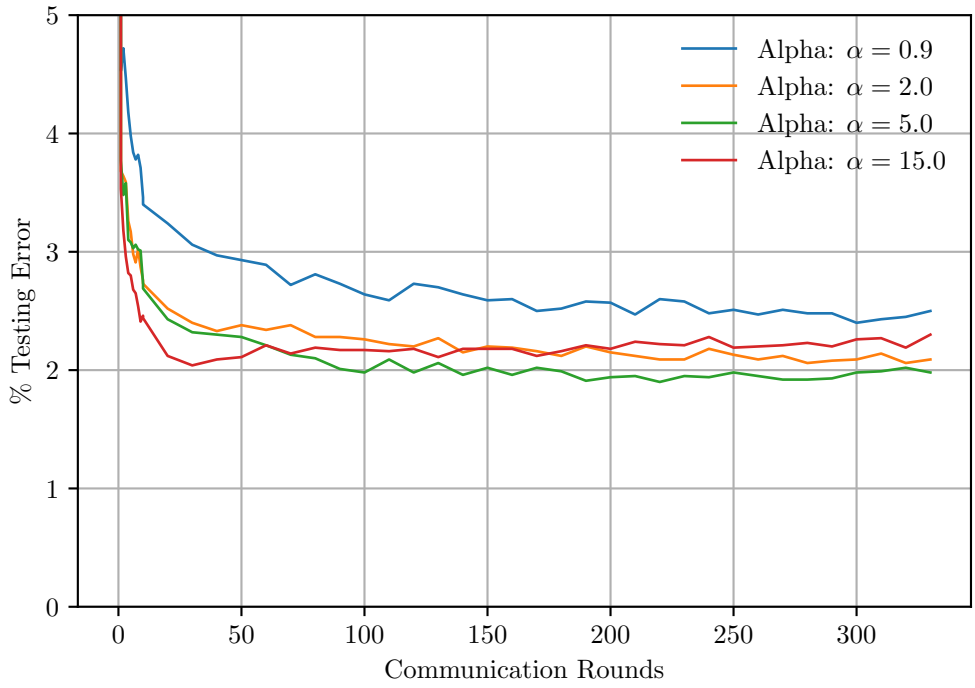


Figure 5.4: Best viewed in colour. MNIST classification for different values of Alpha in $D_{AR}^{(\alpha)}$. Communication rounds refer to the amounts of updates communicated to the server.

26 letters of the English alphabet. Furthermore, ancient Japanese writing uses a cursive and outdated style that most Japanese people are not able to read, Clanuwat et al. (2018). Therefore, efforts have been made to digitise these works and translate them, however due to the sheer amount of documents and lack of experts in the field, most texts remain inaccessible (Clanuwat et al., 2018). Machine learning through pattern recognition can therefore offer invaluable help in "translating" these characters into digital and standard Japanese, which could then further be translated into different languages. We focus on the Kuzushiji–49 data set of "deformed" hiragana symbols, Clanuwat et al. (2018), an example can be seen in figure 5.1b. This data set is heavily imbalanced since different characters have different occurrences in writing. Heterogeneity is also further introduced when noticing that different characters can have different ways of writing them (Clanuwat et al., 2018). See figure 5.5 for an example of this.

For this experiment, due to the large size of the training set, see table 5.1, we consider 25 clients with homogenous splits of data. Note that this data is still heterogeneous across clients due to inherent label heterogeneity. We consider a learning rate of $\eta = 0.0002$ across both PVI and PGVI. Note that we have 49 output nodes in this case, and not simply 10 as for the MNIST data set.

For an alpha value of $\alpha = 2.5$ we can see that PGVI significantly outperforms PVI as shown in figure 5.6. PGVI achieves an accuracy of 81.0% while PVI only achieves 77.4% during the same amount of iterations. Therefore, we can see that especially for increasingly

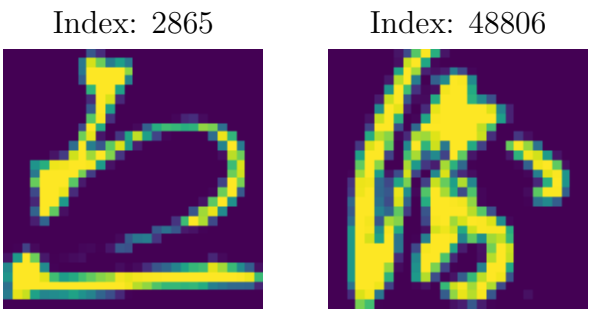


Figure 5.5: Two different versions of the same character in the Kuzushiji–49 data set. Both Kuzushiji represent the character 'e', label 45 in the data set, but have different ways of writing it. The character is the one of fewest occurrences in the entire data set.

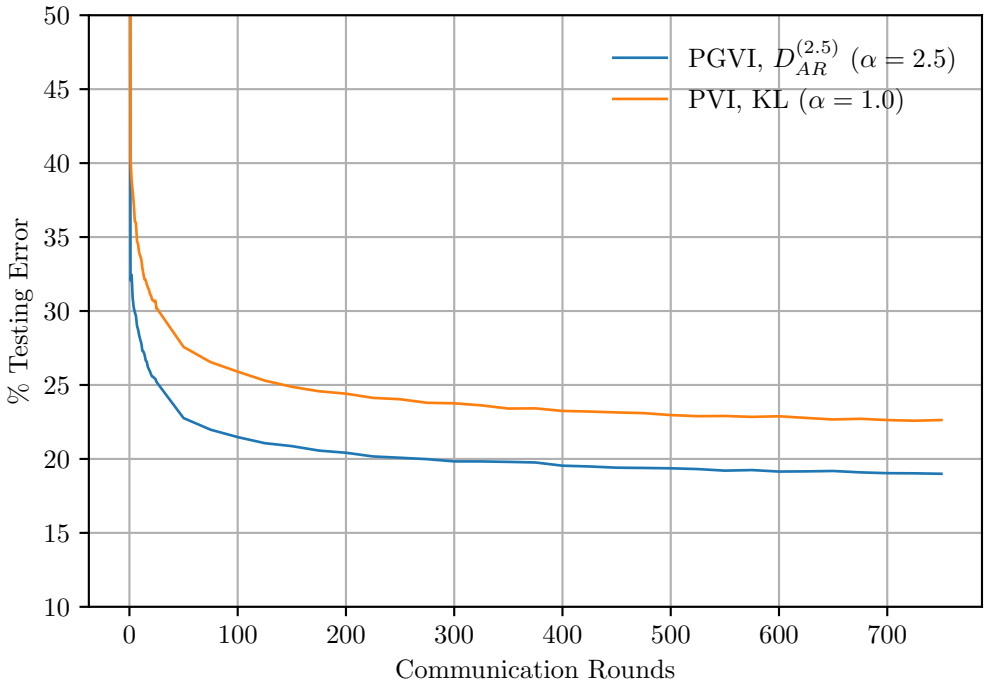


Figure 5.6: Best viewed in colour. PGVI vs VI on the Kuzushiji–49 classification data set. We use $D_{AR}^{(2.5)}$ as a divergence. Communication rounds refer to the amounts of updates communicated to the server.

complex, heterogeneous data sets PGVI is a more desirable target for a decision maker, since for classification tasks, she would be interested in higher accuracy.

5.2.3 CIFAR–10 tiny images classification

Image recognition is an instrumental tool in many AI applications, and hence working with colour images, such as those in figure 5.1c, for classification demonstrates the effectiveness of such methods. Notably, recognition of cancer cells in healthcare falls under applications of such classification methods, and demonstrates the potential for federated learning methods.

In particular we consider the CIFAR–10 data set of Krizhevsky (2009) where we have balanced data across the 10 classes we consider. For this data set, the amount of input nodes is significantly larger than for the previous experiments, increasing the number of parameters in the model. We consider a learning rate $\eta = 0.0002$, with 10 homogeneously split clients and mini–batch size of 256.

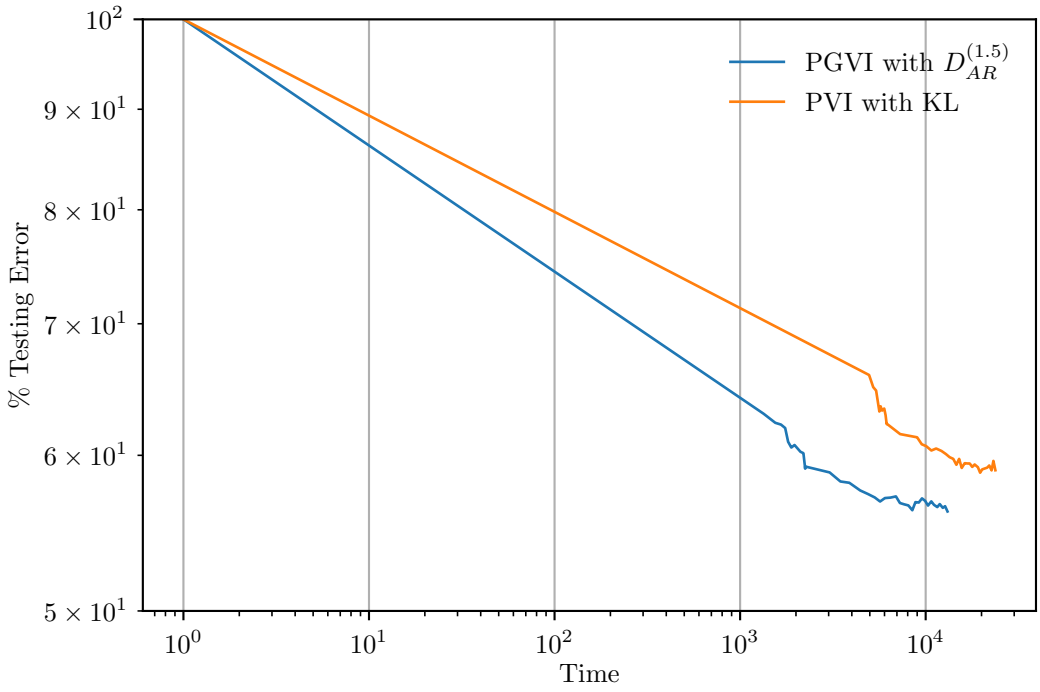


Figure 5.7: Best viewed in colour. PGVI vs PVI on the CIFAR–10 classification data set. We use $D_{AR}^{(1.5)}$ as a divergence. The x and y axis are in log scale.

Figure 5.7 shows that when considering PVI and PGVI on the exact same starting conditions with the only difference being the divergence measure chose, PGVI can outperform PVI both in terms of speed, time in seconds to train, and achieve higher accuracy. However, even though PGVI outperforms PVI here, it still lacks significantly in accuracy when comparing it with state of the art implementations of federated learning, consider for instance Gradient Masked Averaging FedAdam of Tenison et al. (2023), where they achieve 87% accuracy using ResNet18.

Chapter 6

Conclusion and Future Work

This thesis has introduced Partitioned Generalised Variational Inference, which we have developed as a robust alternative to PVI. PGVI fits under the probabilistic approaches to federated learning and improves upon the existing literature through the ease of implementation, and the empirical performance in comparison to standard Partitioned Variational Inference. Furthermore, this thesis unifies the frequentist and Bayesian federated learning literature, discussing how these approaches deal with the unique challenges imposed by federated and distributed settings, such as heterogeneous data and client distributions. We have laid out a necessary foundation and introduction to federated learning, concerning both frequentist and probabilistic approaches.

6.1 Contributions

Federated learning—the collaborative training of a machine learning model through disjoint data sets that cannot be shared—is a non–trivial, and challenging problem that has inspired growing research interest in the past few years. We have explored the federated learning literature in chapter 2, elaborating on the conception of FL and where this approach was inspired from. Further research on frequentist federated learning has vastly improved on these first FedSGD and FedAvg algorithms, with a myriad of different choices available nowadays to address specific problems that can occur in federated learning, a selection of which we have discussed. Nevertheless, Bayesian approaches to federated learning have been neglected, with only a handful of contributions having been made in the literature. Therefore, we have reviewed existing literature on Bayesian FL and the mathematical tools to understand these; reframing Bayes' theorem as an optimisation problem, and approaching this in a tractable manner through Variational Inference and Expectation Propagation. These serve as the foundation of Partitioned Variational Inference, which we discuss in detail in chapter 3, since it provides a principled way of performing approximate Bayesian inference in federated settings. In particular, our contributions to PVI include discussions on damping in synchronous and asynchronous PVI, as well as implementing PVI from scratch to assess the performance of PVI empirically. We have found this to perform well against a global VI approach, where the entire data set is known.

The use of Bayes' theorem, however presents unique challenges under model misspecification where we do not know the true data generating process, and hence have a misspecified likelihood function. Chapter 4 therefore discusses the Generalised Variational Inference framework, which we utilise in order to develop a novel federated learning algorithm, which we term Partitioned Generalised Variational Inference. We demonstrate the performance of this approach and the robustness to model misspecification empirically, showing how we can be robust to outliers and misspecified likelihood functions through different loss functions. This approach does not model the Bayesian posterior distribution, but is rather concerned with targeting a distribution that is of interest to a decision maker. We have further demonstrated how PGVI can lead to more desirable uncertainty quantification. Additionally, we have discussed how the PGVI objective bridges the gap between Bayesian and frequentist federated learning through the arbitrariness of loss functions and divergence measures.

Furthermore, chapter 5 demonstrates the empirical performance of PGVI on real world data sets in deep learning classification tasks. In practice, federated learning has many applications ranging from simple handwritten text identification, to cancer cell recognition in healthcare, which are natural problem domains to consider under federated settings in regards to the increasing desire to keep local data private. Furthermore, we demonstrate that PGVI can be implemented in existing PVI code bases. The performance of PGVI is typically at least as good as PVI and can even outperform it, as well as global VI, when we consider classification through Bayesian Neural Networks, as demonstrated in chapter 5.

6.2 Open Problems

This thesis focuses heavily on the algorithmic frameworks for federated learning, however neglecting specific challenges present in federated learning. In particular, we would like to continue working on local data privacy, and how we can be robust to adversarial attacks from third party malicious actors, and adversarial servers, as well as robustness to adversarial clients, which in our opinion presents unique and fascinating challenges to federated learning, which have yet to be explored thoroughly. Additionally, the Gradient Masked Averaging approach presents interesting ideas to address data heterogeneity, especially when different clients can have different data generating processes. The implementation of GMA in conjunction with PGVI offers an interesting branch of research, potentially improving on challenges with data heterogeneity across clients in PGVI.

Most notably, the theoretical guarantees underlying both PGVI and PVI, as well as many other FL algorithms, are sparse. Distributed Bayesian computation is not well understood, and therefore we would like to further explore even simple concentration results for PGVI. We have already started exploring concentration of generalised Bayesian and GVI posteriors, which we believe could aid in further understanding necessary conditions for PGVI posteriors to concentrate around some optimal parameter.

So far we have only considered the Alpha–Rényi divergence in the experiments of chapter 5, with a simplistic Bayesian Neural Network architecture. However, having introduced different loss functions and divergence measures as well, we would like to explore the effect of

further changing the divergence has on the predictive distribution in the classification tasks considered. We also do not know the exact data generating process under which the data was created, hence we do have model misspecification in the experiments, so changing the log likelihood to the beta loss function might improve the performance of PGVI significantly. And since the BNN experiments have demonstrated some drawbacks of PGVI in terms of accuracy in comparison to frequentist methods, to draw a better comparison we should either implement frequentist methods on fully connected neural networks with a single hidden layer, or implement PGVI on ResNet architecture.

Furthermore, since we are in part motivated by better uncertainty quantification, we could demonstrate PGVI by considering model (epistemic) and data (aleatoric) uncertainty through Gaussian Process regression, as done in Felekis et al. (2022) for global GVI. These require more time than was available for this thesis, but presents key ideas that we would like to continue exploring beyond this thesis.

Chapter 7

Project Management

This thesis initially was concerned with understanding only Partitioned Variational Inference and hence, our initial focus was on developing the knowledge necessary to understand probabilistic machine learning and Partitioned Variational Inference in particular. In order to do this, and go beyond the objectives outlined in the specification required excellent project management. Our initial objectives where, 1. understanding EP and VI in order to explore PVI, 2. fix the code base of Ashman et al. (2022) to validate the experiments for PVI, and 3. develop PVI from scratch to consider new problem domains. As can be seen in figure 7.1, we achieved these objectives by the beginning of term 2. To do this, we first considered an introduction probabilistic machine learning through Rogers and Girolami (2016), on which we built by exploring first Variational Inference, especially through Blei et al. (2016) and then Expectation Propagation through Minka (2001b). We consolidated this knowledge by many fruitful discussions that allowed me personally to understand what I initially did not understand as well as I thought I did. Having build the necessary foundations we explored the PVI paper of Bui et al. (2018) extensively, which in combination with attempting to fix the code base of Ashman et al. (2022), developed our understanding of PVI. And due to efficient time management, despite multiple illnesses throughout the project, meant that I was able to review both Generalised Variational Inference and frequentist federated learning algorithms over the course of the Christmas holidays and the initial few weeks of term 2. This efficient time management allowed time for developing PGVI and contributing to PVI.

7.1 Legal, Social, and Ethical Considerations

Legal, social, and ethical considerations are of no concern to this thesis. The data and the code base that we use is freely available for non–commercial—as well as commercial in some cases—purposes and does therefore not violate any privacy nor legal concerns. For ethical concerns, we like to highlight that this thesis is not primarily concerned with client data privacy against adversaries and hence does not employ state of the art methods like differential privacy or secure aggregation to protect against data leakage during communication. In fact, the development of federated learning aims to improve privacy of local data, therefore working towards more consumer protections against privacy violations by centralised data

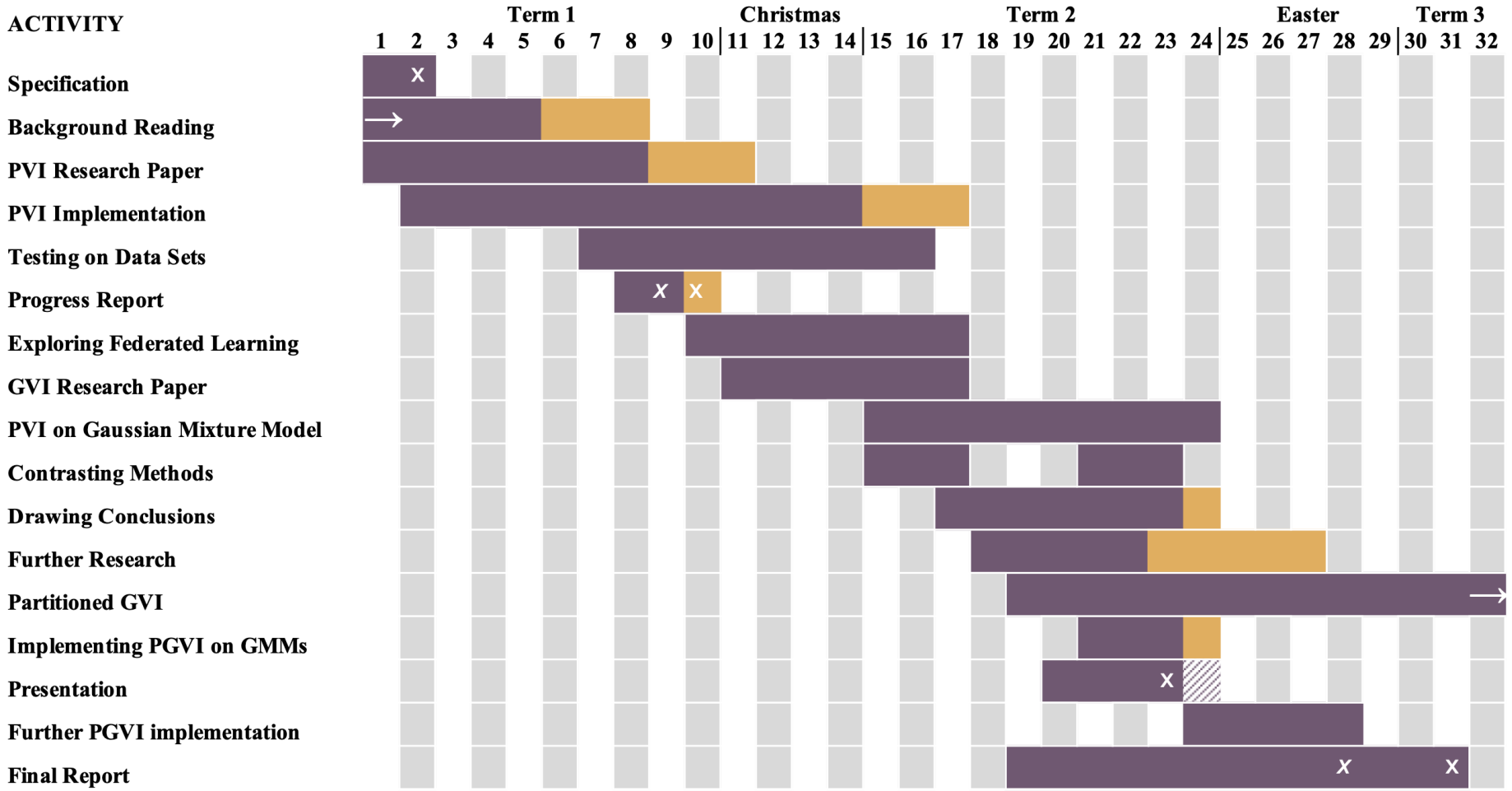


Figure 7.1: Best viewed in colour. Gantt chart showing progress made throughout the academic year. Purple shadowing illustrates planned duration, while yellow shading represents additional time taken, for a certain task. The one rectangle with shading through vertical lines represents initially allocated time, that was not required. Note that the bars are often interrelated and doing one simultaneously with another task implies that these are either related or the time was split between them.

collection. The employment of federated learning also means that servers do not store client data, hence making it harder for adversaries to obtain large amounts of confidential data.

To this end, we will make all code used throughout the experiments publicly available, and free to use, and describe the set–ups considered in the experiments in detail to make this work reproducible. We hope that we can improve upon privacy concerns in the future and that others might use this to research to further develop federated learning.

Appendix A

Relevant Measure Theoretic Results and Definitions

We define our measure space on the measurable space $(\Theta, \mathcal{E})$, where Θ is a polish space and $\mathcal{E}$ the corresponding borel sigma algebra, and while a borel sigma algebra is standard, polish spaces are not and hence we define them here as given in (Cohn, 2013).

Definition 26 (Polish Space) *A Polish Space Θ is a separable topological space that can be metrized using a complete metric.*

Further, we define our notion of probability measures.

Definition 27 (Probability Measure) *A set function μ over a measurable space $(\Theta, \mathcal{E})$, defined as a space and corresponding sigma algebra, is a probability measure if*

1. *μ satisfies the definition of a measure,* **and**
2. *$\mu(\Theta) = 1$.*

The following is an equivalent definition for the Kullback–Leibler divergence between two probability measures, given in Pinski et al. (2015).

Definition 28 (KL Divergence between Measures) *For a target measure $\mu \in \mathcal{P}(\Theta)$ that we want to approximate by a simpler measure $\nu \in \mathcal{Q} \subset \mathcal{P}(\Theta)$ such that $\nu \ll \mu$, then the KLD is*

$$\mathrm{KLD}(\nu||\mu) = \int_{\Theta} \left(\log \frac{d\nu}{d\mu}(\theta)\right) \frac{d\nu}{d\mu}(\theta)\, d\mu(\theta)$$

We also give a definition for convex functions on potentially non–convex domains, due to Miller (2021).

Definition 29 (Convex Function) *A function $\psi : \Theta \to \mathbb{R}$ is convex if $\forall \theta, \theta' \in \Theta$ and $\forall \lambda \in [0,1]$ such that $\lambda\theta + (1-\lambda)\theta' \in \Theta$ then*

$$\psi(\lambda\theta + (1-\lambda)\theta') \leq \lambda\psi(\theta) + (1-\lambda)\psi(\theta')$$

If Θ is convex then $\forall \theta, \theta' \in \Theta$ we have $\lambda\theta + (1-\lambda)\theta' \in \Theta$, $\forall \lambda \in [0,1]$.

Appendix B

Proofs

Here is a collection of proofs of theorems, lemmas or propositions used within this thesis that would impede the clarity and structure of the text.

Appendix B.1. Proof of Theorem 9

We proof the theorem, Minka (2004), that under exponential family distributions with equivalent sufficient statistics, the KL minimisation reduces to moment matching. The proof follows the arguments of Herbich (2005) and is adapted from there, while making parts of it more explicit, especially in the derivation of the Hessian being equivalent to the covariance matrix.

Proof To prove this we will exploit the fact that the gradient of the log partition function is equal to the expectation of the sufficient statistics with respect to the approximating exponential distribution $q_\eta(\theta)$, Bishop (2006, Equation 2.226):

$$\nabla_\eta \log Z(\eta) = \frac{\int \nabla_\eta \exp\{\eta^\top \phi(\theta)\} dh(\theta)}{Z(\eta)} = \mathbb{E}_{q_\eta(\theta)}[\phi(\theta)] \tag{B.1}$$

Then let $f(\eta)$ be a function of the natural parameters η that represents the KL divergence, and let $\hat{p}(\theta) := p_i(x_i|\theta)\, q^{\setminus i}(\theta)$.

$$\begin{aligned} f(\eta) :=\ & \mathrm{KL}(\hat{p}(\theta)||q_\eta(\theta)) = \mathbb{E}_{\hat{p}(\theta)}\Big[\log \frac{\hat{p}(\theta)}{q_\eta(\theta)}\Big] \\ &= \mathbb{E}_{\hat{p}(\theta)}[\log \hat{p}(\theta)] - \mathbb{E}_{\hat{p}(\theta)}[\log q_\eta(\theta)] \\ &= \mathbb{E}_{\hat{p}(\theta)}[\log \hat{p}(\theta)] + \mathbb{E}_{\hat{p}(\theta)}[\log Z(\eta)] - \mathbb{E}_{\hat{p}(\theta)}[\eta^\top \phi(\theta)] \\ &= \mathbb{E}_{\hat{p}(\theta)}[\log \hat{p}(\theta)] + \log Z(\eta) - \eta^\top \mathbb{E}_{\hat{p}(\theta)}[\phi(\theta)] \end{aligned}$$

This is minimized where $\nabla_\eta f(\eta) = 0$, and using equation B.1 we get:

$$\begin{aligned} \nabla_\eta f(\eta) &= \nabla_\eta \cancelto{0}{\mathbb{E}_{\hat{p}(\theta)}[\log \hat{p}(\theta)]} + \nabla_\eta \log Z(\eta) - \nabla_\eta \eta^\top \mathbb{E}_{\hat{p}(\theta)}[\phi(\theta)] \\ &= \mathbb{E}_{q_\eta(\theta)}[\phi(\theta)] - \mathbb{E}_{\hat{p}(\theta)}[\phi(\theta)] = 0 \end{aligned}$$

Rearranging gives the desired result. It remains to be shown that this is indeed a minimum, and we will consider the matrix of second derivatives (Hessian) of $f(\eta)$ to do this.

$$
\begin{aligned}
[\nabla\nabla_\eta f(\eta)]_{\mathrm{i,j}} &= [\nabla_{\eta_j}(\nabla_{\eta_i}\log Z(\eta) - \cancel{\mathbb{E}_{p(\theta)}[\phi(\theta)]}^{0})]_{i,j} = \Big[\frac{\partial^2 \log Z(\eta)}{\partial\eta_i\,\partial\eta_j}\Big]_{i,j} \\
&= \frac{\partial}{\partial\eta_j}\frac{\int \phi_i(\theta)\exp\{\eta^\top\phi(\theta)\}dh(\theta)}{Z(\eta)} \\
&= \frac{\partial}{\partial\eta_j}(g(\eta)t(\eta)), \\
&\text{where } g(\eta) := \int \phi_i(\theta)\exp\{\eta^\top\phi(\theta)\}\,dh(\theta), \text{ and } t(\eta) := \frac{1}{Z(\eta)} \\
&= t(\eta)\frac{\partial}{\partial\eta_j}(g(\eta)) + g(\eta)\frac{\partial}{\partial\eta_j}(t(\eta)), \text{ by the chain rule. Then} \\
t(\eta)\frac{\partial}{\partial\eta_j}(g(\eta)) &= \frac{1}{Z(\eta)}\int \phi_i(\theta)\frac{\partial}{\partial\eta_j}\exp\{\eta^\top\phi(\theta)\}dh(\theta) \\
&= \frac{1}{Z(\eta)}\int \phi_i(\theta)\,\exp\{\eta^\top\phi(\theta)\}\phi_j(\theta)dh(\theta) \\
&= \int \phi_i(\theta)\phi_j(\theta)\frac{1}{Z(\eta)}\exp\{\eta^\top\phi(\theta)\}dh(\theta) = \mathbb{E}_{q_\eta(\theta)}[\phi_i(\theta)\phi_j(\theta)] \\
g(\eta)\frac{\partial}{\partial\eta_j}(t(\eta)) &= g(\eta)\Big(-\frac{1}{(Z(\eta))^2}\Big)\frac{\partial}{\partial\eta_j}Z(\eta) = -\frac{g(\eta)}{(Z(\eta))^2}\int\frac{\partial}{\partial\eta_j}\exp\{\eta^\top\phi(\theta)\}dh(\theta) \\
&= -\frac{\int \phi_i(\theta)\exp\{\eta^\top\phi(\theta)\}dh(\theta)\int\exp\{\eta^\top\phi(\theta)\}\phi_j(\theta)dh(\theta)}{Z(\eta)Z(\eta)} \\
&= -\int \phi_i(\theta)\frac{1}{Z(\eta)}\exp\{\eta^\top\phi(\theta)\}dh(\theta)\int\phi_j(\theta)\frac{1}{Z(\eta)}\exp\{\eta^\top\phi(\theta)\}dh(\theta) \\
&= -\mathbb{E}_{q_\eta(\theta)}[\phi_i(\theta)]\,\mathbb{E}_{q_\eta(\theta)}[\phi_j(\theta)] \\
\implies [\nabla\nabla_\eta f(\eta)]_{\mathrm{i,j}} &= \mathbb{E}_{q_\eta(\theta)}[\phi_i(\theta)\phi_j(\theta)] - \mathbb{E}_{q_\eta(\theta)}[\phi_i(\theta)]\,\mathbb{E}_{q_\eta(\theta)}[\phi_j(\theta)]
\end{aligned}
$$

This is the covariance matrix at the solution over $q_\eta(\theta)$, which is by definition positive-definite, hence a minimum, Amari (2016). ■

The first and last line of the proof showing that the Hessian is equivalent to the covariance matrix are given in Herbich (2005) without derivation; this is provided here.

Appendix B.2. Proof of Theorem 12

We give here the proof of theorem 12, since it adds to the understanding of the theorem and aids in developing ideas of how to approach potential proofs of distributed Bayesian inference in the future. We highlight that this proof is taken directly from Miller (2021) and is not original to this thesis.

Proof [Due to Miller (2021)] We note that$f_n(\theta) \to f(\theta)$ a.s. and that $\theta_0 \in A_\epsilon$ and therefore $A_\epsilon \neq \emptyset$. Let $\epsilon > 0$ and let $\mu_n(E) = \int_E \exp(-nf_n(\theta))d\Pi(\theta)$ where $E \subseteq \Theta$. Since $\infty > \int_\Theta p_n(x_{1:n}|\theta)d\Pi(\theta) = \mu_n(\Theta)$, $\forall n$ per assumption. Then let $\beta > 0$

$$
1 - \Pi_n(A_\epsilon) = \Pi_n(A_\epsilon^c) = \frac{\mu_n(A_\epsilon^c)}{\mu_n(\Theta)}\frac{\exp(n(f(\theta_0)+\beta))}{\exp(n(f(\theta_0)+\beta))}
$$

We aim to show that this probability goes to 0 as $n \to \infty$, by seeing that the numerator is finite and bounded, and that the denominator goes to infinity, hence the fraction goes to 0.

Considering the numerator:
We know that $\exists N$ large enough such that $\forall n > N$, $\inf_{\theta \in A_\epsilon^c} f_n(\theta) \geq f(\theta_0) + \beta$ by the assumption that $\liminf_n \inf_{\theta \in A_\epsilon^c} f_n(\theta) > f(\theta_0) + \epsilon$, $\forall \epsilon > 0$. Therefore, $\forall n > N$ and $\forall \theta \in A_\epsilon^c$ we have that $\exp\big(-n(f_n(\theta) - f(\theta_0) - \beta)\big) \leq 1$, and hence

$$\exp(n(f(\theta_0) + \beta))\mu_n(A_\epsilon^c) = \int_{A_\epsilon^c} \exp(-n(f_n(\theta) - f(\theta_0) - \beta))d\Pi(\theta) \leq \int_{A_\epsilon^c} d\Pi(\theta) \leq 1$$

Considering the denominator:
Let $\theta \in A_{\beta/2}$ such that $f_n(\theta) - f(\theta_0) - \beta \to f(\theta) - f(\theta_0) - \beta < -\beta/2 < 0$ which implies that $\exp(-n(f_n(\theta) - f(\theta_0) - \beta)) \to \infty$ as $n \to \infty$. And since $\exp(\,\cdot\,) \to [0, \infty]$ regardless of it's argument $\exp(-n(f_n(\theta) - f(\theta_0) - \beta))$ is a non negative sequence of functions for all n, we can apply Fatou's lemma (Rudin, 1987, Lemma 1.28)

$$\begin{aligned}\liminf_{n\to\infty} \exp(n(f(\theta_0) + \beta))\mu_n(\Theta) &= \liminf_{n\to\infty} \int_{A_{\beta/2}} \exp(-n(f_n(\theta) - f(\theta_0) - \beta))d\Pi(\theta) \\ &\geq \int_{A_{\beta/2}} \big(\liminf_{n\to\infty} \exp(-n(f_n(\theta) - f(\theta_0) - \beta))\big)d\Pi(\theta) = \infty\end{aligned}$$

and since $\Pi(A_{\beta/2}) > 0$ by definition. Finally, since $\mu_n(\Theta) > \mu_n(A_{\beta/2})$ we have that $\exp(n(f(\theta_0) + \beta))\mu_n(\Theta) \to \infty$. This gives the desired result. ■

This result, while nice and easy to follow, does not extend to rates of convergence as the results of Kleijn and van der Vaart (2006), or Shalizi (2009) do. However, since theorem 12 is concerned with showing that the Bayesian posterior concentrates around some parameter $\theta_0 \in \Theta$ we do not need to consider more difficult versions here. Although, the proof of Shalizi (2009) is quite long, it is also elegant and an important result.

Appendix C

Derivations and Proofs for Experiment Equations

Appendix C.1. Kullback Leibler Divergence

We give a proof of the closed form of the Kullback–Leibler divergence between two multivariate Gaussian distributions.

Proof We define two Multivariate Gaussians, $q(\theta) \sim \mathcal{N}(\mu, \Sigma)$ and $\pi(\theta) \sim \mathcal{N}(\nu, \Lambda)$, then

$$\begin{aligned}
\mathrm{KL}(q||\pi) &= \int q(\theta|\mu,\Sigma) \log \frac{\frac{\exp\left\{-\frac{1}{2}(\theta-\mu)^\top \Sigma^{-1}(\theta-\mu)\right\}}{(2\pi \det(\Sigma))^{1/2}}}{\frac{\exp\left\{-\frac{1}{2}(\theta-\nu)^\top \Lambda^{-1}(\theta-\nu)\right\}}{(2\pi \det(\Lambda))^{1/2}}} d\theta \\
&= \int q(\theta|\mu,\Sigma) \log \frac{\det(\Lambda)^{1/2}}{\det(\Sigma)^{1/2}} \frac{\exp\{-\frac{1}{2}(\theta-\mu)^\top \Sigma^{-1}(\theta-\mu)\}}{\exp\{-\frac{1}{2}(\theta-\nu)^\top \Lambda^{-1}(\theta-\nu)\}} d\theta
\end{aligned}$$

Rearranging this as an expectation and using the rules of logarithm gives:

$$\begin{aligned}
&= \mathbb{E}_{q(\theta)}\Big[\log\Big(\frac{\det(\Lambda)}{\det(\Sigma)}\Big)^{1/2} - \frac{1}{2}(\theta-\mu)^\top \Sigma^{-1}(\theta-\mu) + \frac{1}{2}(\theta-\nu)^\top \Lambda^{-1}(\theta-\nu)\Big] \\
&= \frac{1}{2}\mathbb{E}_{q(\theta)}\Big[\log\Big(\frac{\det(\Lambda)}{\det(\Sigma)}\Big) - (\theta-\mu)^\top \Sigma^{-1}(\theta-\mu) + (\theta-\nu)^\top \Lambda^{-1}(\theta-\nu)\Big]
\end{aligned}$$

Since $(\theta-\mu)^\top \Sigma^{-1}(\theta-\mu) \in \mathbb{R}$ and $tr(a) = a$ for $a \in \mathbb{R}$, we can rewrite this.

$$= \frac{1}{2}\mathbb{E}_{q(\theta)}\Big[\log\Big(\frac{\det(\Lambda)}{\det(\Sigma)}\Big) - \mathrm{tr}\big((\theta-\mu)^\top \Sigma^{-1}(\theta-\mu)\big) + \big((\theta-\nu)^\top \Lambda^{-1}(\theta-\nu)\big)\Big]$$

Using the standard result from linear algebra that $\mathrm{tr}(ABC) = \mathrm{tr}(BCA)$, and through linearity of expectation, we get

$$\begin{aligned}
&= \frac{1}{2}\mathbb{E}_{q(\theta)}\Big[\log\Big(\frac{\det(\Lambda)}{\det(\Sigma)}\Big) - \mathrm{tr}\big(\Sigma^{-1}(\theta-\mu)(\theta-\mu)^\top\big) + \big((\theta-\nu)^\top \Lambda^{-1}(\theta-\nu)\big)\Big] \\
&= \frac{1}{2}\mathbb{E}_{q(\theta)}\Big[\log\Big(\frac{\det(\Lambda)}{\det(\Sigma)}\Big) - \mathrm{tr}\big(\Sigma^{-1}(\theta-\mu)(\theta-\mu)^\top\big) + \big((\theta-\nu)^\top \Lambda^{-1}(\theta-\nu)\big)\Big] \\
&= \frac{1}{2}\Big[\log\Big(\frac{\det(\Lambda)}{\det(\Sigma)}\Big) - \mathbb{E}_{q(\theta)}\big[\mathrm{tr}\big(\Sigma^{-1}(\theta-\mu)(\theta-\mu)^\top\big)\big] + \mathbb{E}_{q(\theta)}\big[\big((\theta-\nu)^\top \Lambda^{-1}(\theta-\nu)\big)\big]\Big] \\
&= \frac{1}{2}\Big[\log\Big(\frac{\det(\Lambda)}{\det(\Sigma)}\Big) - \mathrm{tr}\big(\mathbb{E}_{q(\theta)}[\Sigma^{-1}(\theta-\mu)(\theta-\mu)^\top]\big) + \mathbb{E}_{q(\theta)}\big[(\theta-\nu)^\top \Lambda^{-1}(\theta-\nu)\big]\Big]
\end{aligned}$$

where we have used the fact that $\mathbb{E}[\text{tr}(\cdot)] = \text{tr}(\mathbb{E}[\cdot])$ and linearity of expectations. Then using the fact that for $\theta \sim q(\theta)$ we have

$$\mathbb{E}_{\theta\sim q(\theta)}\Big[(\theta-\nu)^\top\Lambda(\theta-\nu)\Big] = (\mu-\nu)^\top\Lambda(\mu-\nu) + \text{tr}(\Lambda\Sigma)$$

then we have that

$$\begin{aligned}
&= \frac{1}{2}\Big[\log\Big(\frac{\det(\Lambda)}{\det(\Sigma)}\Big) - \text{tr}\big(\Sigma^{-1}\mathbb{E}_{q(\theta)}[(\theta-\mu)(\theta-\mu)^\top]\big) + (\mu-\nu)^\top\Lambda^{-1}(\mu-\nu) + \text{tr}(\Lambda^{-1}\Sigma)\Big] \\
&= \frac{1}{2}\Big[\log\Big(\frac{\det(\Lambda)}{\det(\Sigma)}\Big) - \text{tr}\big(\Sigma^{-1}\Sigma\big) + (\mu-\nu)^\top\Lambda^{-1}(\mu-\nu) + \text{tr}(\Lambda^{-1}\Sigma)\Big] \\
&= \frac{1}{2}\Big[\log\Big(\frac{\det(\Lambda)}{\det(\Sigma)}\Big) - \text{tr}\big(\text{I}_D\big) + (\mu-\nu)^\top\Lambda^{-1}(\mu-\nu) + \text{tr}(\Lambda^{-1}\Sigma)\Big]
\end{aligned}$$

which gives the desired result. ■

Pardo Llorente (2006) gives a different way of deriving this result. He first considers the Alpha–Rényi divergence between two multivariate Gaussians—as we do in the next section—and then takes the limits as $\alpha \to 1$, which recovers the KL divergence.

Appendix C.2. Alpha–Rényi Divergence

Definition 30 (Rényi's Alpha Divergence) *The Alpha–Rényi Divergence between two probability distributions $q(\theta)$ and $\pi(\theta)$ with respect to a dominating measure $\mu(\theta)$, is defined as the following equation, for all $\alpha \neq 0, 1$:*

$$\text{D}_{AR}^{(\alpha)}(q(\theta)||\pi(\theta)) = \frac{1}{\alpha(\alpha-1)}\log\int q(\theta)^\alpha\pi(\theta)^{1-\alpha}d\mu(\theta) \tag{C.1}$$

C.2.1 Alpha–Rényi Divergence between two Multivariate Gaussians

Before we start the proof of the Alpha–Rényi Divergence between two Multivariate Gaussians, we state a well known lemma, due to Henderson and Searle (1981).

Lemma 31 (Woodbury Matrix Identity) *For matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{k\times k}$, $U \in \mathbb{R}^{n\times k}$ and $V \in \mathbb{R}^{k\times n}$, with A and B invertible, then the following identity holds:*

$$(A+UBV)^{-1} = A^{-1} - A^{-1}U(B^{-1}+VA^{-1}U)^{-1}VA^{-1} \tag{C.2}$$

We define two Multivariate Gaussians as $q(\theta) \sim \mathcal{N}(\mu, \Sigma)$ and $\pi(\theta) \sim \mathcal{N}(\eta, \Lambda)$, then the Alpha–Rényi Divergence ($D_{AR}^{(\alpha)}$) between $q(\theta)$ and $\pi(\theta)$ can be expressed in closed form.

Lemma 32 (Alpha-Rényi Divergence between two Multivariate Gaussians) *The $D_{AR}^{(\alpha)}$ between the multivariate Gaussians $q(\theta)$ and $\pi(\theta)$ takes the following form:*

$$\begin{aligned}
\text{D}_{AR}^{(\alpha)}(q(\theta)||\pi(\theta)) = &\frac{(\mu-\eta)^\top(\alpha\Lambda+(1-\alpha)\Sigma)^{-1}(\mu-\eta)}{2} \\
&-\frac{1}{2\alpha(\alpha-1)}\log\frac{|\alpha\Lambda+(1-\alpha)\Sigma|}{|\Sigma|^{1-\alpha}\,|\Lambda|^\alpha}
\end{aligned} \tag{C.3}$$

Proof We consider the $D_{AR}^{(\alpha)}$ between $p(\theta)$ and $\pi(\theta)$, as defined in equation (1).

$$\mathrm{D}_{AR}^{(\alpha)}(q(\theta)||\pi(\theta)) = \frac{1}{\alpha(\alpha-1)} \log \int \Big(\frac{\exp\{-\frac{1}{2}(\theta-\mu)^\top \Sigma^{-1}(\theta-\mu)\}}{(2\pi)^{D/2}|\Sigma|^{1/2}}\Big)^\alpha \Big(\frac{\exp\{-\frac{1}{2}(\theta-\eta)^\top \Lambda^{-1}(\theta-\eta)\}}{(2\pi)^{D/2}|\Lambda|^{1/2}}\Big)^{1-\alpha} d\mu(\theta)$$

Then let $f(\theta)$ be the integral in the equation above, then we can write:

$$f(\theta) := \int \mathcal{N}(\mu,\Sigma)^\alpha \mathcal{N}(\eta,\Lambda)^{1-\alpha} d\mu(\theta) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\Sigma|^{\frac{\alpha}{2}}|\Lambda|^{\frac{1-\alpha}{2}}} \int \exp\Big\{\frac{-\alpha}{2}(\theta-\mu)^\top\Sigma^{-1}(\theta-\mu) - \frac{(1-\alpha)}{2}(\theta-\eta)^\top\Lambda^{-1}(\theta-\eta)\Big\} d\mu(\theta)$$

Then we let $g(\theta) := (\theta-\mu)^\top\alpha\Sigma^{-1}(\theta-\mu) - (\theta-\eta)^\top(1-\alpha)\Lambda^{-1}(\theta-\eta)$, then we can write:

$$\begin{aligned} g(\theta) =& \theta^\top(\alpha\Sigma^{-1} + (1-\alpha)\Lambda^{-1})\theta - 2(\mu^\top\alpha\Sigma^{-1} + \eta^\top(1-\alpha)\Lambda^{-1})\theta \\ &+ \mu^\top\alpha\Sigma^{-1}\mu + \eta^\top(1-\alpha)\Lambda^{-1}\eta \end{aligned}$$

We note here that covariance matrices are symmetric, hence $\Sigma^\top = \Sigma$. And we can simply complete the square for this equation, using the formula for $\theta^\top M\theta - 2b^\top\theta + \text{const.}$, which gives $= (\theta - M^{-1}b)^\top M(\theta - M^{-1}b) - b^\top M^{-1}b + \text{const.}$. Then we can define b and M as follows:

$$\begin{aligned} M &= \alpha\Sigma^{-1} + (1-\alpha)\Lambda^{-1} \\ b &= (\mu^\top\alpha\Sigma^{-1} + \eta^\top(1-\alpha)\Lambda^{-1})^\top \\ &= \alpha\Sigma^{-1}\mu + (1-\alpha)\Lambda^{-1}\eta \\ M^{-1}b &= (\alpha\Sigma^{-1} + (1-\alpha)\Lambda^{-1})^{-1}(\alpha\Sigma^{-1}\mu + (1-\alpha)\Lambda^{-1}\eta) \end{aligned}$$

Then we can substitute this into $g(\theta)$:

$$g(\theta) = (\theta - M^{-1}b)^\top M(\theta - M^{-1}b) - b^\top M^{-1}b + \mu^\top\alpha\Sigma^{-1}\mu + \eta^\top(1-\alpha)\Lambda^{-1}\eta$$

Substituting g back into $f(\theta)$ results in the following equation, taking out the terms independent of θ from the integral:

$$\begin{aligned} f(\theta) =& \frac{1}{|\Sigma|^{\frac{\alpha}{2}}|\Lambda|^{\frac{1-\alpha}{2}}}\Big[\int \frac{1}{(2\pi)^{\frac{D}{2}}} \exp\Big\{-\frac{1}{2}(\theta-M^{-1}b)^\top M(\theta-M^{-1}b)\Big\} d\mu(\theta)\Big] \\ &\times \exp\Big\{-\frac{1}{2}(\mu^\top\alpha\Sigma^{-1}\mu + \eta^\top(1-\alpha)\Lambda^{-1}\eta - b^\top M^{-1}b)\Big\} \\ =& \frac{|M^{-1}|^{\frac{1}{2}}}{|\Sigma|^{\frac{\alpha}{2}}|\Lambda|^{\frac{1-\alpha}{2}}} \times \exp\Big\{-\frac{1}{2}(\mu^\top\alpha\Sigma^{-1}\mu + \eta^\top(1-\alpha)\Lambda^{-1}\eta - b^\top M^{-1}b)\Big\} \end{aligned}$$

Where we have used the fact that the integral was an unnormalised multivariate Gaussian distribution, which then returns it's normalising constant. Now we evaluate the terms

individually, taking the ARD terms before the integral $\frac{1}{\alpha(\alpha-1)}\log(\cdot)$, which decompose into a sum through the multiplication symbol.

(1)

$$\begin{aligned}
&\frac{1}{\alpha(\alpha-1)}\log\Big(\frac{|M^{-1}|^{\frac{1}{2}}}{|\Sigma|^{\frac{\alpha}{2}}|\Lambda|^{\frac{1-\alpha}{2}}}\Big)\\
=&\frac{1}{\alpha(\alpha-1)}\log\Big(\frac{|(\alpha\Sigma^{-1}+(1-\alpha)\Lambda^{-1})^{-1}|}{|\Sigma|^{\alpha}|\Lambda|^{1-\alpha}}\Big)^{1/2}\\
=&\frac{1}{2\alpha(\alpha-1)}\log\frac{|(\alpha\Sigma^{-1}+(1-\alpha)\Lambda^{-1})^{-1}|}{|\Sigma|^{\alpha}|\Lambda|^{1-\alpha}}
\end{aligned}$$

Since we do not want to deal with matrix inverses and since $|M^{-1}| = 1/|M|$, we can do the following to M:

$$\begin{aligned}
M = \alpha\Sigma^{-1}+(1-\alpha)\Lambda^{-1} &= \Sigma^{-1}(\alpha I_D + \Sigma(1-\alpha)\Lambda^{-1})\\
&= \Sigma^{-1}(\alpha\Lambda+(1-\alpha)\Sigma)\Lambda^{-1}\\
&= \Lambda^{-1}(\alpha\Lambda+(1-\alpha)\Sigma)\Sigma^{-1}
\end{aligned}$$

Taking the determinant of this inverse:

$$\begin{aligned}
|M^{-1}| &= |(\Sigma^{-1}(\alpha\Lambda+(1-\alpha)\Sigma)\Lambda^{-1})^{-1}|\\
&= 1/|\Sigma^{-1}(\alpha\Lambda+(1-\alpha)\Sigma)\Lambda^{-1}|\\
&= 1/(|\Sigma^{-1}||(\alpha\Lambda+(1-\alpha)\Sigma)||\Lambda^{-1}|)\\
&= \frac{|\Sigma||\Lambda|}{|(\alpha\Lambda+(1-\alpha)\Sigma)|}
\end{aligned}$$

Substituting this back into the equation, gives:

$$\begin{aligned}
&= \frac{1}{2\alpha(\alpha-1)}\log\frac{|\Sigma||\Lambda|}{|\Sigma|^{\alpha}|\Lambda|^{1-\alpha}|(\alpha\Lambda+(1-\alpha)\Sigma)|}\\
&= \frac{1}{2\alpha(\alpha-1)}\log\frac{|\Sigma|^{1-\alpha}|\Lambda|^{-\alpha}}{|(\alpha\Lambda+(1-\alpha)\Sigma)|}\\
&= -\frac{1}{2\alpha(\alpha-1)}\log\frac{|(\alpha\Lambda+(1-\alpha)\Sigma)|}{|\Sigma|^{1-\alpha}|\Lambda|^{\alpha}}
\end{aligned}$$

(2)

$$\begin{aligned}
&\frac{1}{\alpha(\alpha-1)}\log\exp\Big\{-\frac{1}{2}(\mu^{\top}\alpha\Sigma^{-1}\mu+\eta^{\top}(1-\alpha)\Lambda^{-1}\eta-b^{\top}M^{-1}b)\Big\}\\
=&\frac{1}{\alpha(\alpha-1)}\Big(-\frac{1}{2}(\mu^{\top}\alpha\Sigma^{-1}\mu+\eta^{\top}(1-\alpha)\Lambda^{-1}\eta-b^{\top}M^{-1}b)\Big)\\
=&-\frac{1}{2\alpha(\alpha-1)}(\mu^{\top}\alpha\Sigma^{-1}\mu+\eta^{\top}(1-\alpha)\Lambda^{-1}\eta-b^{\top}M^{-1}b)
\end{aligned}$$

We initially only are concerned with the part inside the exponential. Let's begin by analysing the term $b^{\top}M^{-1}b$, where we employ the trick that $MM^{-1} = I_D$. Further noting, since M

is symmetric, so is M^{-1} and hence we have $b^\top M^{-1} = M^{-1}b$.

$$\begin{aligned} b^\top M^{-1} b &= (\alpha\Sigma^{-1}\mu + (1-\alpha)\Lambda^{-1}\eta)M^{-1}(\mu^\top\alpha\Sigma^{-1} + \eta^\top(1-\alpha)\Lambda^{-1}) \\ &= \alpha^2\mu^\top\Sigma^{-1}M^{-1}\Sigma^{-1}\mu + (1-\alpha)^2\eta^\top\Lambda^{-1}M^{-1}\Lambda^{-1}\eta \\ &\quad - 2\alpha(\alpha-1)\mu^\top\Sigma^{-1}M^{-1}\Lambda^{-1}\eta \end{aligned}$$

Combining this with the equation before for part (2), we get:

$$\begin{aligned} =&\mu^\top\alpha\Sigma^{-1}\mu - \alpha^2\mu^\top\Sigma^{-1}M^{-1}\Sigma^{-1}\mu \\ &+ \eta^\top(1-\alpha)\Lambda^{-1}\eta - (1-\alpha)^2\eta^\top\Lambda^{-1}M^{-1}\Lambda^{-1}\eta \\ &+ 2\alpha(\alpha-1)\mu^\top\Sigma^{-1}M^{-1}\Lambda^{-1}\eta \\ \\ =&\alpha\mu^\top(\Sigma^{-1} - \alpha\Sigma^{-1}M^{-1}\Sigma^{-1})\mu \\ &+ (1-\alpha)\eta^\top(\Lambda^{-1} - (1-\alpha)\Lambda^{-1}M^{-1}\Lambda^{-1})\eta \\ &+ 2\alpha(\alpha-1)\mu^\top\Sigma^{-1}M^{-1}\Lambda^{-1}\eta \end{aligned} \tag{C.4}$$

Focusing on the last term, we can apply the inverse identity $A^{-1}B^{-1}C^{-1} = (CBA)^{-1}$, and recalling the equivalent formulation for M from step (1) earlier:

$$\begin{aligned} 2\alpha(\alpha-1)\mu^\top\Sigma^{-1}M^{-1}\Lambda^{-1}\eta &= 2\alpha(\alpha-1)\mu^\top(\Lambda M\Sigma)^{-1}\eta \\ &= 2\alpha(\alpha-1)\mu^\top(\Lambda\Lambda^{-1}(\alpha\Lambda + (1-\alpha)\Sigma)\Sigma^{-1}\Sigma)^{-1}\eta \\ &= 2\alpha(\alpha-1)\mu^\top(\alpha\Lambda + (1-\alpha)\Sigma)^{-1}\eta \end{aligned}$$

For the rest we will make use of the Woodbury Matrix Identity, due to Henderson and Searle (1981), to help solve the other parts.

Taking the expression inside the μs of equation (3), we can rearrange it into a form on which we can apply the Woodbury Matrix Identity, Henderson and Searle (1981), using $U = V = I_D$ and $n = k = D$:

$$\begin{aligned} \Sigma^{-1} - \alpha\Sigma^{-1}M^{-1}\Sigma^{-1} &= \Sigma^{-1} - \alpha\Sigma^{-1}(\alpha\Sigma^{-1} + (1-\alpha)\Lambda^{-1})^{-1}\Sigma^{-1} \\ &= \Sigma^{-1} - \frac{\alpha(1-\alpha)}{\alpha(1-\alpha)}\alpha\Sigma^{-1}(\alpha\Sigma^{-1} + (1-\alpha)\Lambda^{-1})^{-1}\Sigma^{-1} \\ &= \Sigma^{-1} - \frac{1}{1-\alpha}\Sigma^{-1}(\alpha(1-\alpha))(\alpha\Sigma^{-1} + (1-\alpha)\Lambda^{-1})^{-1}\Sigma^{-1} \\ &= \Sigma^{-1} - \frac{1}{1-\alpha}\Sigma^{-1}((\alpha(1-\alpha))^{-1}(\alpha\Sigma^{-1} + (1-\alpha)\Lambda^{-1}))^{-1}\Sigma^{-1} \\ &= \Sigma^{-1} - \frac{1}{1-\alpha}\Sigma^{-1}(\frac{1}{1-\alpha}\Sigma^{-1} + \frac{1}{\alpha}\Lambda^{-1})^{-1}\Sigma^{-1} \\ \frac{1}{1-\alpha}(\Sigma^{-1} - \alpha\Sigma^{-1}M^{-1}\Sigma^{-1}) &= \frac{1}{1-\alpha}\Sigma^{-1} - \frac{1}{1-\alpha}\Sigma^{-1}(\frac{1}{1-\alpha}\Sigma^{-1} + \frac{1}{\alpha}\Lambda^{-1})^{-1}\frac{1}{1-\alpha}\Sigma^{-1} \\ &= ((\frac{1}{1-\alpha}\Sigma^{-1})^{-1} + (\frac{1}{\alpha}\Lambda^{-1})^{-1})^{-1} \\ &= ((1-\alpha)\Sigma + \alpha\Lambda)^{-1} \end{aligned}$$

$$\implies \Sigma^{-1} - \alpha\Sigma^{-1}M^{-1}\Sigma^{-1} = (1-\alpha)((1-\alpha)\Sigma + \alpha\Lambda)^{-1}$$

Similarly, we can solve the inside of the second part of equation (3) to get:

$$\begin{aligned}
\Lambda^{-1} - (1-\alpha)\Lambda^{-1}M^{-1}\Lambda^{-1} &= \Lambda^{-1} - (1-\alpha)\Lambda^{-1}(\alpha\Sigma^{-1} + (1-\alpha)\Lambda^{-1})^{-1}\Lambda^{-1} \\
\frac{1}{\alpha}(\Lambda^{-1} - \alpha\Lambda^{-1}M^{-1}\Lambda^{-1}) &= \frac{1}{\alpha}\Lambda^{-1} - \frac{1}{\alpha}\Lambda^{-1}(\frac{1}{1-\alpha}\Sigma^{-1} + \frac{1}{\alpha}\Lambda^{-1})^{-1}\frac{1}{\alpha}\Lambda^{-1} \\
&= (\alpha\Lambda + (1-\alpha)\Sigma)^{-1}
\end{aligned}$$

$$\implies \Lambda^{-1} - (1-\alpha)\Lambda^{-1}M^{-1}\Lambda^{-1} = \alpha((1-\alpha)\Sigma + \alpha\Lambda)^{-1}$$

Therefore, we can rewrite equation (3):

$$\begin{aligned}
(\mathbf{3}) = \alpha(1-\alpha)\mu^\top((1-\alpha)\Sigma + \alpha\Lambda)^{-1}\mu \\
-2\alpha(1-\alpha)\mu^\top((1-\alpha)\Sigma + \alpha\Lambda)^{-1}\eta \\
+\alpha(1-\alpha)\eta^\top((1-\alpha)\Sigma + \alpha\Lambda)^{-1}\eta
\end{aligned} \tag{C.5}$$

We can combine this using the binomial theorem. To do this first define the following:

$$T = (1-\alpha)\Sigma + \alpha\Lambda$$

Then we can rearrange (5) into the following form:

$$\begin{aligned}
&= \alpha(1-\alpha)(\mu^\top T^{-1}\mu - 2\mu^\top T^{-1}\eta + \eta^\top T^{-1}\eta) \\
&= \alpha(1-\alpha)(\mu-\eta)^\top T^{-1}(\mu-\eta) \\
&= \alpha(1-\alpha)(\mu-\eta)^\top((1-\alpha)\Sigma + \alpha\Lambda)^{-1}(\mu-\eta)
\end{aligned}$$

Substituting this back into $\exp(-\frac{1}{2}(\cdot))$ and taking the $\frac{1}{\alpha(1-\alpha)}\log(\cdot)$ of the exponential we get:

$$\begin{aligned}
&\frac{1}{\alpha(\alpha-1)}\log\Big\{\exp\big[-\frac{1}{2}(\alpha(1-\alpha)(\mu-\eta)^\top((1-\alpha)\Sigma + \alpha\Lambda)^{-1}(\mu-\eta))\big]\Big\} \\
=&\frac{1}{\alpha(\alpha-1)}\log\Big\{\exp\big[\frac{1}{2}(\alpha(\alpha-1)(\mu-\eta)^\top((1-\alpha)\Sigma + \alpha\Lambda)^{-1}(\mu-\eta))\big]\Big\} \\
=&\frac{1}{\alpha(\alpha-1)}\big[\frac{1}{2}(\alpha(\alpha-1)(\mu-\eta)^\top((1-\alpha)\Sigma + \alpha\Lambda)^{-1}(\mu-\eta))\big] \\
=&\frac{(\mu-\eta)^\top((1-\alpha)\Sigma + \alpha\Lambda)^{-1}(\mu-\eta)}{2}
\end{aligned}$$

Hence, combining **(1)** and **(2)** we get a closed form expression for the ARD between two Multivariate Gaussian Distributions:

$$\begin{aligned}\mathrm{D}_{AR}^{(\alpha)}(q(\theta)||\pi(\theta)) &= -\frac{1}{2\alpha(\alpha-1)}\log\frac{|\alpha\Lambda+(1-\alpha)\Sigma|}{|\Sigma|^{1-\alpha}\,|\Lambda|^{\alpha}} \\ &\quad+\frac{(\mu-\eta)^{\top}((1-\alpha)\Sigma+\alpha\Lambda)^{-1}(\mu-\eta)}{2} \\ \mathrm{D}_{AR}^{(\alpha)}(q(\theta)||\pi(\theta)) &= \frac{(\mu-\eta)^{\top}(\alpha\Lambda+(1-\alpha)\Sigma)^{-1}(\mu-\eta)}{2} - \frac{1}{2\alpha(\alpha-1)}\log\frac{|\alpha\Lambda+(1-\alpha)\Sigma|}{|\Sigma|^{1-\alpha}\,|\Lambda|^{\alpha}}\end{aligned}$$

Therefore proving the Lemma. ■

C.2.2 In the case of Isotropic Gaussians

We posit the normal distributions $\mathcal{N}(\mu,\sigma I_D)$ and $\mathcal{N}(\eta,\lambda I_D)$, which means we can simplify Lemma 5.

Corollary 33 *In the case of isotropic Gaussians, i.e. where the covariance matrices are given by some constants times the identity matrix, equation 2 reduces to the following:*

$$\mathrm{D}_{AR}^{(\alpha)}(q(\theta)||\pi(\theta)) = \frac{(\mu-\eta)^{\top}(\mu-\eta)}{2(\alpha\lambda+(1-\alpha)\sigma)} - \frac{D}{2\alpha(\alpha-1)}\log\frac{\alpha\lambda+(1-\alpha)\sigma}{\sigma^{1-\alpha}\,\lambda^{\alpha}} \tag{C.6}$$

Proof We can use the equation found earlier in our proof and simply plug the covariance matrices into it. Taking the first term we can see that:

$$\begin{aligned}((1-\alpha)\Sigma+\alpha\Lambda)^{-1} &= (\alpha\lambda I_D+(1-\alpha)\sigma I_D)^{-1} \\ &= ((\alpha\lambda+(1-\alpha)\sigma)I_D)^{-1} \\ &= (\alpha\lambda+(1-\alpha)\sigma)^{-1}I_D\end{aligned}$$

$$\implies \frac{(\mu-\eta)^{\top}(\alpha\lambda I_D+(1-\alpha)\sigma I_D)^{-1}(\mu-\eta)}{2} = \frac{(\mu-\eta)^{\top}(\mu-\eta)}{2(\alpha\lambda+(1-\alpha)\sigma)}$$

The second part notices that the determinant of a constant c times an $n\times n$ dimensional matrix M can be written as $|cM| = c^n|M|$ and that the determinant of the identity matrix is $|I_D| = 1$, therefore $|cI_D| = c^D$. Then the second part becomes:

$$\begin{aligned}\frac{1}{2\alpha(\alpha-1)}\log\frac{|\alpha\Lambda+(1-\alpha)\Sigma|}{|\Sigma|^{1-\alpha}\,|\Lambda|^{\alpha}} &= \frac{1}{2\alpha(\alpha-1)}\log\frac{|\alpha\lambda I_D+(1-\alpha)\sigma I_D|}{|\sigma I_D|^{1-\alpha}\,|\lambda I_D|^{\alpha}} \\ &= \frac{1}{2\alpha(\alpha-1)}\log\frac{|(\alpha\lambda+(1-\alpha)\sigma)I_D|}{\sigma^{D(1-\alpha)}\,\lambda^{D\alpha}} \\ &= \frac{1}{2\alpha(\alpha-1)}\log\frac{(\alpha\lambda+(1-\alpha)\sigma)^{D}}{\sigma^{D(1-\alpha)}\,\lambda^{D\alpha}} \\ &= \frac{1}{2\alpha(\alpha-1)}\log\Big(\frac{\alpha\lambda+(1-\alpha)\sigma}{\sigma^{1-\alpha}\,\lambda^{\alpha}}\Big)^{D}\end{aligned}$$

$$\implies \frac{1}{2\alpha(\alpha-1)}\log\frac{|\alpha\lambda I_D+(1-\alpha)\sigma I_D|}{|\sigma I_D|^{1-\alpha}\,|\lambda I_D|^{\alpha}}=\frac{D}{2\alpha(\alpha-1)}\log\frac{\alpha\lambda+(1-\alpha)\sigma}{\sigma^{1-\alpha}\,\lambda^{\alpha}}$$

Hence, the desired result follows. ■

Appendix C.3. Beta Loss Function

Proof of lemma 25, which states the closed form of the Beta Loss function with a misspecified normal distribution likelihood.

Proof Recall that the $\mathcal{L}_B^{(\beta)}$ is given as:

$$\mathcal{L}_B^{(\beta)}(p_m(\boldsymbol{x}_m|\theta),\{\theta_s\}_{s=1}^S)=-\frac{1}{\beta-1}p_m(\boldsymbol{x}_m|\theta)^{\beta-1}+\frac{\int_{\mathcal{X}}p_m(\boldsymbol{y}|\theta)^{\beta}d\boldsymbol{y}}{\beta}$$

The first part is given directly by this definition. Then for $p_m(\boldsymbol{y}|\theta)\sim\mathcal{N}(\boldsymbol{y}|\theta,\Sigma)$ we can evaluate the integral.

$$\begin{aligned}\int_{\mathcal{X}}p_m(\boldsymbol{y}|\theta)^{\beta}d\boldsymbol{y}&=\int_{\mathcal{X}}\Big(\frac{1}{(2\pi\det(\Sigma))^{1/2}}\exp\Big\{-\frac{1}{2}(\theta-\boldsymbol{y})^{\top}\Sigma^{-1}(\theta-\boldsymbol{y})\Big\}\Big)^{\beta}d\boldsymbol{y}\\&=\int_{\mathcal{X}}\frac{1}{(2\pi\det(\Sigma))^{\beta/2}}\exp\Big\{-\frac{\beta}{2}(\theta-\boldsymbol{y})^{\top}\Sigma^{-1}(\theta-\boldsymbol{y})\Big\}d\boldsymbol{y}\\(\spadesuit)\qquad&=\int_{\mathcal{X}}\frac{1}{(2\pi\det(\Sigma))^{\beta/2}}\exp\Big\{-\frac{1}{2}(\boldsymbol{y}-\theta)^{\top}\Big(\frac{1}{\beta}\Sigma\Big)^{-1}(\boldsymbol{y}-\theta)\Big\}d\boldsymbol{y}\end{aligned}$$

Then we recall that the integral over a normalised Gaussian distribution equals 1, and that the covariance does not depend on $\boldsymbol{y}$.

$$\begin{aligned}1&=\int\frac{1}{(2\pi\det(\Lambda))^{1/2}}\exp\Big\{-\frac{1}{2}(\boldsymbol{y}-\nu)^{\top}\Lambda^{-1}(\boldsymbol{y}-\nu)\Big\}d\boldsymbol{y}\\1&=\frac{1}{(2\pi\det(\Lambda))^{1/2}}\int\exp\Big\{-\frac{1}{2}(\boldsymbol{y}-\nu)^{\top}\Lambda^{-1}(\boldsymbol{y}-\nu)\Big\}d\boldsymbol{y}\\(2\pi\det(\Lambda))^{1/2}&=\int\exp\Big\{-\frac{1}{2}(\boldsymbol{y}-\nu)^{\top}\Lambda^{-1}(\boldsymbol{y}-\nu)\Big\}d\boldsymbol{y}\end{aligned}$$

We can therefore substitute this result into $(\spadesuit)$ with $\Lambda=\frac{1}{\beta}\Sigma$.

$$\begin{aligned}(\spadesuit)\qquad&=\frac{1}{(2\pi\det(\Sigma))^{\beta/2}}\Big(2\pi\det\Big(\frac{1}{\beta}\Sigma\Big)\Big)^{1/2}\\&=\frac{\Big(2\pi\det\big(\Sigma\big)\Big)^{1/2}}{(2\pi\det(\Sigma))^{\beta/2}}\Big(\frac{1}{\beta}\Big)^{D/2}\\&=\frac{1}{(2\pi\det(\Sigma))^{(\beta-1)/2}\beta^{D/2}}\end{aligned}$$

This follows by basic linear algebra, where D is the dimension of $\boldsymbol{y}$. Substituting this back into the β–loss function, yields the desired result.

$$\mathcal{L}_B^{(\beta)}(p_m(\boldsymbol{x}_m|\{\theta_s\}_{s=1}^S)) = -\frac{1}{\beta-1}p_m(\boldsymbol{x}_m|\theta)^{\beta-1} - \frac{1}{(2\pi\det(\Sigma))^{(\beta-1)/2}\beta^{D/2}}\frac{1}{\beta}$$

Hence, proving lemma 25. ■

Appendix C.4. PGVI code implementation in BNNs

We include our Python code for the calculation of the Alpha–Rényi divergence for the Bayesian Neural Networks of chapter 5.2, which uses the existing code base of Ashman et al. (2022) for existing methods, such as calculating the log partition function $A(\cdot)$. This part can be found in the file `pvi/distributions/base.py` and is used in the calculation of the local ELBO in the file `pvi/clients/base.py`.

```
def ar_divergence(self, p, alpha=0.5, calc_log_ap=True):

        assert type(p) == type(self),
        if alpha == 0:
                alpha = 0.01

        frac_1 = (alpha * (alpha - 1)) ** (-1)
        frac_2 = (alpha - 1) ** (-1)
        frac_3 = alpha ** (-1)


        log_a = self.log_a().squeeze()

        np1_q = self.nat_params["np1"]
        np2_q = self.nat_params["np2"]

        np1_p = p.nat_params["np1"]
        np2_p = p.nat_params["np2"]

        np1_combi = alpha * np1_q + (1 - alpha) * np1_p
        np2_combi = alpha * np2_q + (1 - alpha) * np2_p

        try:
                np1_combi = torch.reshape(np1_combi, [1])
                np2_combi = torch.reshape(np2_combi, [1])
        except RuntimeError as e:
                pass
```

```
batch_dims = len(self.nat_params["np1"].shape) - 1

if batch_dims == 0:
        d = 1
else:
        d = np1_combi.shape[-1]

log_a_combi = -0.5 * np.log(np.pi) * d
log_a_combi += (-(np1_combi ** 2) / (4 * np2_combi)
                - 0.5 * (-2 * np2_combi).log()).sum(-1)



ar = frac_1 * log_a_combi
ar -= frac_2 * log_a

if calc_log_ap:
        ar += frac_3 * p.log_a()

return ar
```

Listing C.1: Python code snippet of the implmentation for the Alpha–Rényi divergence in PGVI for BNNs. This is inserted into the existing code base for BNNs of Ashman et al. (2022) in the file `pvi/distributions/base.py`.

Bibliography

Virginia Aglietti, Edwin V Bonilla, Theodoros Damoulas, and Sally Cripps. Structured variational inference in continuous cox process models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Sungjin Ahn, Babak Shahbaba, and Max Welling. Distributed stochastic gradient MCMC. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1044–1052, Bejing, China, 2014. PMLR.

Maruan Al-Shedivat, Jennifer Gillenwater, Eric Xing, and Afshin Rostamizadeh. Federated learning via posterior averaging: A new perspective and practical algorithms. In *International Conference on Learning Representations*, 2021.

Shun-ichi Amari. *Information Geometry and Its Applications.* Springer, Tokyo, Japan, 2016. ISBN 9784431559771.

Matthew Ashman, Thang D. Bui, Cuong V. Nguyen, Stratis Markou, Adrian Weller, Siddharth Swaroop, and Richard E. Turner. Partitioned variational inference: A framework for probabilistic federated learning. *arXiv preprint arXiv:2202.12275*, 2022.

Hagai Attias. Inferring parameters and structure of latent variable models by variational bayes. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 21–30, 1999.

Robert H. Berk. Limiting behavior of posterior distributions when the model is incorrect. *The Annals of Mathematical Statistics*, 37(1):51 – 58, 1966.

José M. Bernardo and Adrian F. M. Smith. *Bayesian theory.* Wiley Series in Probability and Statistics, Chichester, England, 2000. ISBN 9780470316870.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 9780387310732.

Pier G. Bissiri, Chris Holmes, and Stephen G. Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 78(5):1103–1130, 2016.

David Blei, Alp Kucukelbir, and Jon McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–87, 01 2016.

Andrea Braides. *Gamma-Convergence for Beginners*. Oxford University Press, 2002. ISBN 978019850784.

Alexander Buchholz, Daniel Ahfock, and Sylvia Richardson. Distributed computation for marginal likelihood based model choice. *Bayesian Analysis*, 18(2):607 – 638, 2023.

Thang D. Bui, Cuong V. Nguyen, Siddharth Swaroop, and Richard E. Turner. Partitioned variational inference: A unified framework encompassing federated and continual learning. *arXiv preprint arXiv:1811.11206*, 2018.

Wei-Ning Chen, Christopher A Choquette-Choo, Peter Kairouz, and Ananda Theertha Suresh. The fundamental price of secure aggregation in differentially private federated learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 3056–3089. PMLR, 17–23 Jul 2022.

Andrzej Cichocki and Shun-ichi Amari. Families of alpha- beta- and gamma- divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568, 2010.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

Donald L. Cohn. *Measure Theory: Second Edition*. Springer New York, New York, NY, 2013. ISBN 9781461469568.

Luca Corinzia, Ami Beuret, and Joachim M. Buhmann. Variational federated multi-task learning. *arXiv preprint arXiv:1906.06268*, 2021.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory, 2nd Ed*. Wiley-Interscience, Hoboken, N.J, 2006. ISBN 9780471241959.

Imre Csiszar. I-Divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146 – 158, 1975.

Giorgos Felekis, Theo Damoulas, and Brooks Paige. Probabilistic deep learning with generalised variational inference. In *4th Symposium on Advances in Approximate Bayesian Inference*, 2022.

Futoshi Futami, Issei Sato, and Masashi Sugiyama. Variational inference based on robust divergences. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 813–822. PMLR, 09–11 Apr 2018.

Christian Genest. A characterization theorem for externally bayesian groups. *The Annals of Statistics*, 12(3):1100–1105, 1984.

Eric Grivel, Roberto Diversi, and Fernando Merchan. Kullback-leibler and rényi divergence rate for gaussian stationary ARMA processes comparison. *Digital Signal Processing*, 116: 103089, 2021.

Han Guo, Philip Greengard, Hongyi Wang, Andrew Gelman, Yoon Kim, and Eric Xing. Federated learning as variational inference: A scalable expectation propagation approach. In *The Eleventh International Conference on Learning Representations*, 2023.

Jenny Hamer, Mehryar Mohri, and Ananda Theertha Suresh. FedBoost: A communication-efficient algorithm for federated learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3973–3983. PMLR, 2020.

Leonard Hasenclever, Stefan Webb, Thibaut Lienart, Sebastian Vollmer, Balaji Lakshminarayanan, Charles Blundell, and Yee Whye Teh. Distributed bayesian learning with stochastic natural gradient expectation propagation and the posterior server. *Journal of Machine Learning Research*, 18(1):3744–3780, 2017.

Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Xinghua Zhu, Jianzong Wang, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. FedML: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

Mikko Heikkilä, Matthew Ashman, Siddharth Swaroop, Richard E Turner, and Antti Honkela. Differentially private partitioned variational inference. *Transactions on machine learning research*, 2023(4), 2023.

Harold V. Henderson and Shayle R. Searle. On deriving the inverse of a sum of matrices. *SIAM Review*, 23(1):53–60, 1981.

James Hensman, Max Zwiessele, and Neil D. Lawrence. Tilted variational bayes. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 356–364, Reykjavik, Iceland, 22–25 Apr 2014. PMLR.

Ralf Herbich. Minimising the kullback-leibler divergence. Technical report, Microsoft Research, 2005.

José Miguel Hernández-Lobato. *Balancing Flexibility and Robustness in Machine Learning: Semi-parametric Methods and Sparse Linear Models.* PhD thesis, Universidad Autonoma de Madrid, Spain, 2010.

Tom Heskes, Manfred Opper, Wim Wiegerinck, Ole Winther, and Onno Zoeter. Approximate inference techniques with expectation constraints. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11015, 2005.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(40):1303–1347, 2013.

Jack Jewson, Jim Q. Smith, and Chris Holmes. Principles of bayesian inference using general divergence criteria. *Entropy*, 20(6):442, 2018.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Rahif Kassab and Osvaldo Simeone. Federated generalized bayesian learning via distributed stein variational gradient descent. *IEEE Transactions on Signal Processing*, 70:2180–2192, 2022.

Mohammad Emtiyaz Khan and Håvard Rue. The bayesian learning rule. *Journal of Machine Learning Research*, 24(281):1–46, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

Bas J. K. Kleijn and Aad W. van der Vaart. Misspecification in infinite-dimensional Bayesian statistics. *The Annals of Statistics*, 34(2):837 – 877, 2006.

Bas J.K. Kleijn and Aad W. van der Vaart. The Bernstein-Von-Mises theorem under misspecification. *Electronic Journal of Statistics*, 6:354–381, 2012.

Jeremias Knoblauch. Frequentist consistency of generalized variational inference. *arXiv preprint arXiv:1912.04946*, 2019.

Jeremias Knoblauch and Theodoros Damoulas. Spatio-temporal Bayesian on-line changepoint detection with model selection. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2718–2727. PMLR, 10–15 Jul 2018.

Jeremias Knoblauch, Jack E Jewson, and Theodoros Damoulas. Doubly robust bayesian inference for non-stationary streaming data with β-divergences. In *Advances in Neural Information Processing Systems*, volume 31, pages 64–75. Curran Associates, Inc., 2018.

Jeremias Knoblauch, Hisham Husain, and Tom Diethe. Optimal continual learning has perfect memory and is NP-hard. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5327–5337. PMLR, 13–18 Jul 2020.

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. An optimization-centric view on bayes' rule: Reviewing and generalizing variational inference. *Journal of Machine Learning Research*, 23(132):1–109, 2022.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Solomon Kullback. *Information Theory and Statistics*. Wiley, New York, 1959. ISBN 9780486696843.

Solomon Kullback and Richard A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yingzhen Li and Richard E Turner. Rényi divergence variational inference. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

David J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, USA, 2003. ISBN 9780521642989.

Grigory Malinovsky, Dmitry Kovalev, Elnur Gasanov, Laurent Condat, and Peter Richtarik. From local SGD to local fixed-point methods for federated learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6692–6701. PMLR, 2020.

Geoffrey J. McLachlan and David Peel. *Finite Mixture Models*. Wiley, New York, 2000. ISBN 0471006262.

Geoffrey J. McLachlan, Sharon X. Lee, and Suren I. Rathnayake. Finite mixture models. *Annual Review of Statistics and Its Application*, 6(1):355–378, 2019.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017.

Marco Avella Medina, José Luis Montiel Olea, Cynthia Rush, and Amilcar Velez. On the robustness to misspecification of α-posteriors and their variational approximations. *Journal of Machine Learning Research*, 23(147):1–51, 2022.

Khaoula el Mekkaoui, Diego Mesquita, Paul Blomstedt, and Samuel Kaski. Federated stochastic gradient langevin dynamics. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1703–1712. PMLR, 2021.

Diego Mesquita, Paul Blomstedt, and Samuel Kaski. Embarrassingly parallel MCMC using deep invertible transformations. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 1244–1252. PMLR, 2020.

Jeffrey W. Miller. Asymptotic normality, concentration, and coverage of generalized posteriors. *Journal of Machine Learning Research*, 22(168):1–53, 2021.

Thomas Minka. Power EP. Technical report, Microsoft Research, Cambridge, 2004.

Thomas Minka. Divergence measures and message passing. Technical report, Microsoft Research, Cambridge, 2005.

Thomas P. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, Cambridge, USA, 2001a.

Thomas P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, page 362–369, San Francisco, CA, USA, 2001b.

Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. ISBN 9780262046824.

Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. ISBN 9780262048439.

Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. Variational bayesian unlearning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16025–16036. Curran Associates, Inc., 2020.

Frank Nielsen. An elementary introduction to information geometry. *Entropy*, 22(10):1100, 2020.

Frank Nielsen. Fast approximations of the jeffreys divergence between univariate gaussian mixtures via mixture conversions to exponential-polynomial distributions. *Entropy*, 23 (11), 2021.

Frank Nielsen. A simple approximation method for the fisher–rao distance between multivariate normal distributions. *Entropy*, 25(4), 2023.

Frank Nielsen and Kazuki Okamura. On the f-divergences between densities of a multivariate location or scale family. *Statistics and Computing*, 34(1):60, 2024.

Leandro Pardo Llorente. *Statistical inference based on divergence measures.* Chapman & Hall/CRC, 2006. ISBN 9781584886006.

F. J. Pinski, G. Simpson, A. M. Stuart, and H. Weber. Kullback-leibler approximation for probability measures on infinite dimensional spaces. *SIAM Journal on Mathematical Analysis*, 47(6):4091–4122, 2015.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006. ISBN 9780262182539.

Sashank Reddi, Zachary Burr Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

Simon Rogers and Mark Girolami. *A First Course in Machine Learning, 2nd Ed.* Chapman & Hall/CRC, 2016. ISBN 9781498738484.

Walter Rudin. *Real and Complex Analysis, 3rd Ed.* McGraw-Hill, Inc., USA, 1987. ISBN 9780070542341.

Steven L. Scott, Alexander W. Blocker, Fernando V. Bonassi, Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bayes and big data: The consensus monte carlo algorithm. *International Journal of Management Science and Engineering Management*, 11:78–88, 2016.

Cosma Rohilla Shalizi. Dynamics of Bayesian updating with dependent data and misspecified models. *Electronic Journal of Statistics*, 3:1039–1074, 2009.

Bingqing Song, Prashant Khanduri, Xinwei Zhang, Jinfeng Yi, and Mingyi Hong. FedAvg converges to zero training loss linearly for overparameterized multi-layer neural networks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 32304–32330. PMLR, 23–29 Jul 2023.

Irene Tenison, Sai Aravind Sreeramadas, Vaikkunth Mugunthan, Edouard Oyallon, Irina Rish, and Eugene Belilovsky. Gradient masked averaging for federated learning. *Transactions on Machine Learning Research*, 2023.

Volker Tresp. A bayesian committee machine. *Neural computation*, 12:2719–41, 2000.

Isidoros Tziotis, Zebang Shen, Ramtin Pedarsani, Hamed Hassani, and Aryan Mokhtari. Straggler-resilient personalized federated learning. *Transactions on Machine Learning Research*, 2023.

Aki Vehtari, Andrew Gelman, Tuomas Sivula, Pasi Jylänki, Dustin Tran, Swupnil Sahai, Paul Blomstedt, John P. Cunningham, David Schiminovich, and Christian P. Robert. Expectation propagation as a way of life: A framework for bayesian inference on partitioned data. *Journal of Machine Learning Research*, 21(1), 2020.

Stephen G. Walker. Bayesian inference with misspecified models. *Journal of Statistical Planning and Inference*, 143(10):1621–1633, 2013.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Arnold Zellner. Optimal information processing and bayes's theorem. *The American Statistician*, 42(4):278–280, 1988.

Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):2008–2026, 2019.

Zihao Zhao, Mengen Luo, and Wenbo Ding. Deep leakage from model in federated learning. In *Conference on Parsimony and Learning*, volume 234 of *Proceedings of Machine Learning Research*, pages 324–340. PMLR, 2023.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.