Abstract

To be written ...

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Akash Kumar Dhaka, Michael Riis Andersen, Pablo Garcia Moreno, Aki Vehtari. Scalable Gaussian Process for Extreme Classification. *The International Workshop on Machine Learning for Signal Processing MLSP* , Espoo, Finland, pages 1–6, October 2020.

II Eero Siivola, Akash Kumar Dhaka, Michael Riis Andersen, Pablo Garcia Moreno, Javier Gonzalez, Aki Vehtari. Preferential Batch Bayesian Optimization. *The International Workshop on Machine Learning for Signal Processing MLSP* , Gold Coast, Australia, November 2021.

III Akash Kumar Dhaka, Alejandro Catalina, Michael Riis Andersen, Mans Magnusson, Jonathan Huggins and Aki Vehtari. Robust, Accurate Stochastic Optimization for Variational Inference. *Advances in Neural Information Processing Systems*, Volume 33, pages 10961–10973, 2020.

IV Akash Kumar Dhaka, Alejandro Catalina, Michael Riis Andersen, Manushi Welandawe, Jonathan Huggins, Aki Vehtari. Challenges and Opportunities in High-dimensional Variational Inference. *Advances in Neural Information Processing Systems*, Volume 34, 2021.

Author's Contribution

Publication I: "Scalable Gaussian Process for Extreme Classification"

Dhaka had the main responsibility in writing the article. Dhaka designed and implemented the models and methods. The original idea was Moreno's. Moreno contributed to coding the algorithm (about 20 percent of the program code). Andersen helped in debugging the code and designing experiments. Andersen, Moreno and Vehtari helped in writing the article (about 25 percent of the text). Moreno and Vehtari supervised the research.

Publication II: "Preferential Batch Bayesian Optimization"

Siivola had the main responsibility in writing the article. Siivola designed and implemented the models and methods. The original idea was Siivola's. Dhaka had the main responsibility in implementing the variational Bayes algorithm (about 15 per cent of the program code). Andersen, Gonzalez, Moreno and Dhaka helped in writing the article (about 25 percent of the text). Gonzalez, Moreno and Vehtari supervised the research.

Publication III: "Robust, Accurate Stochastic Optimization for Variational Inference"

Dhaka had the main responsibility in writing the article. Dhaka designed and implemented the models and methods. The original idea was by Vehtari and Dhaka. Catalina and Huggins contributed to coding the algorithm (about 15 per cent of the program code). Catalina and Magnusson ran experiments on large scale dataset and compiled final results. Andersen, Magnusson, Huggins and Vehtari helped in writing the article (about 25 percent of the text). Magnusson, Huggins and Vehtari supervised the research.

Publication IV: "Challenges and Opportunities in High-dimensional Variational Inference"

Dhaka and Catalina had equal contribution. Dhaka and Catalina designed and implemented the models and methods and ran experiments. The original idea was by Vehtari and Huggins. Welandawe debugged the code and improved existing implementations of optimisers (about 15 percent code). Andersen, Welandawe, Huggins and Vehtari helped in writing the article (about 40 percent of the text). Andersen, Huggins and Vehtari supervised the research.

List of Tables

Abbreviations

ADVI Automatic Differentiation Variational Inference

BBVI Black Box Variational Inference

CAVI Coordinate-Ascent Variational Inference

CLT Central Limit Theorem

DSVI Doubly Stochastic Variational Inference

EC Extreme Classification

EP Expectation Propagation

GP Gaussian process

GPR Gaussian process regression

GPU graphics processing unit

LDA Linear Discriminant Analysis

L-BFGS limited-memory Broyden-Fletcher-Goldfarb-Shannon

MC Monte Carlo

MCMC Markov chain Monte Carlo

MoG Mixture of Gaussians

NSSM Non Linear State Space Models

PBBO Preferential Batch Bayesian optimization

PSIS Pareto Smoothed Importance Sampling

SGD Stochastic Gradient Descent

SVI Stochastic Variational Inference

1. Introduction

There are many ways of learning functions from data, one particularly elegant way to do it is by ’probabilistic modelling’. Concepts such as noise in data and uncertainty hold the key in probabilistic modelling. In most situations, we can not claim to have perfect knowledge of the systems we find around us. Even if we do, the interactions among the physical variables could be so complex that not all factors can be taken into account. Consider a toy problem of estimating the distance a car travels when brakes are applied. This problem can be solved using the laws of physics given we know perfectly the initial speed of car, friction coefficient of the road, condition of tyres etc., but this information is generally not available and would vary from place to place. Another way to solve this problem can be using a data driven approach. By collecting noisy data from experiments with a set of different tyre and road conditions, the function between initial speed and distance travelled can be modelled. The measurements made from sensors and used in the data analysis can be expected to be noisy.

Uncertainty is usually classified into two categories, *aleatoric* and *epistemic*. There are many possible sources of aleatoric uncertainty. It arises due to inherent random variability between the members of a population which we sample from like a shuffled pack of cards or a bag of poker chips. The data used to train the models can be seen as a realization of a random process. It also comes in because of random error in measurements. In contrast, *epistemic* uncertainty is due to factors one could in principle know but does not in practice and can be explained away with more training data. For example, there maybe many possible ways of fitting a function to given data.

It has been argued that reasoning with the help of probability rules is the only coherent way to perform inference from data under uncertainty [Cox, 1946, Jaynes, 2003]. This work considers data analysis through a probabilistic view point and using concepts from Bayesian probability theory. Bayesian inference allows to determine a distribution over underlying model parameter values after observing data, which is known as the posterior distribution. The other way, ’deterministic modelling’ offers no clear and coherent way of incorporating the notion of uncertainty into analysis. Most *Deep Learning models* can be seen as deterministic functions where the user gets point estimates of underlying model parameters and predictions. Unfortunately, in practice obtaining posteriors is generally intractable. Then a user has to resort to approximate inference. One particular approximate inference scheme is known as variational inference and it will be the focus of this thesis. Recently there has been a plethora of work where variational inference has been used to compute approximations to intractable posterior and elaborate models.

More robust and reliable inference makes data analysis for decision-making by scientists and organizations (e.g., corporations, governments, and foundations) more reliable and reproducible. In next section, we describe the general setting and key objects in Bayesian methods which sets up the background for introduction of approximate inference and variational inference in detail.

1.1 A case for Bayesian methods

Many machine learning and deep learning methods reduce inference to an *optimization* problem, which is to determine the optimal value of parameters (or weights) which minimise a loss function such as the mean squared error for a regression problem or cross entropy for classification. These approaches do not take uncertainty into account. Bayesian methods allow us to express the uncertainty described above in the form of distributions over both parameters and predictions. The key aspect defining Bayesian methods is *marginalisation* which gives us solutions on the parameter space weighted by a density, generally the posterior density. Having a distribution over the parameters enables us to compute a distribution over predictions in the output space. This is an elegant way of transferring the uncertainty in estimation of parameters to predictions. Certain problems often associated with models such us neural networks, including robustness and data scarcity can then be handled in a more principled and elegant manner. Another attractive attribute of Bayesian methods is that they often subsume other methods. Methods which find point estimates of model parameters is a case, where the uncertainty associated with inference of parameter has vanished completely. In Bayesian terms this can be denoted by the posterior taking the form of a Dirac mass function.

1.2 Mechanics of Exact Bayesian Inference

Consider we have set of observations $D = \{x_n, y_n\}$, where the $x_n \in \mathbb{R}^k$ is the fixed input and generally a vector of attributes (such as an image or a sentence), and y_n is the scalar output. The relation of the output to the input is modelled by means of some function $f(\cdot)$. The function could be a simple linear model, a hierarchical model or even some complex non linear function such as a neural network, having some parameters denoted by $\boldsymbol{\theta}$. Then the model is set as: a prior over the parameter, $p(\boldsymbol{\theta})$, and a likelihood, $p(D \mid \boldsymbol{\theta})$, which is the conditional probability density of observing the given configuration of data D over the parameter space after making some suitable transformation. The parameter space can be, discrete, continuous or a mixture of both. The object of interest for us is then the posterior: $p(\boldsymbol{\theta} \mid D)$ which is the conditional probability or probability density over the parameter space *after* observing the data. Given the posterior we can answer questions such as: what is the most likely value of θ, and what is the probability that θ equals certain value given the data in case of discrete parameter space or correspondingly in continuous space, what is the probability that $\boldsymbol{\theta}$ lies between two values.

Inference, i.e. computing the posterior, can be done following Bayes' Rule:

$$p(\theta \mid D) = \frac{p(D \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(D)}, \tag{1.1}$$

where the denominator, $p(D)$, is the normalizing constant. $p(D)$ is known as the marginal likelihood because its computations involves marginalizing, or integrating out or summing for the discrete case, the term $\boldsymbol{\theta}$. That is, $p(D) = \int p(D \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$. This integral results in a density for cases where the data D is continuous and probability mass where D is discrete. The integral has to be a converging one for the posterior to be a proper distribution.

Apart from the integral discussed above, we commonly need a few more integrations, for example when computing the summary statistics of high dimensional posteriors, we often look at the first two moments: the posterior mean

$$\boldsymbol{\mu} = \int \boldsymbol{\theta} p(\boldsymbol{\theta} \mid D) d\boldsymbol{\theta},$$

and the posterior covariance:

$$\Sigma = \int (\boldsymbol{\theta} - \boldsymbol{\mu})(\boldsymbol{\theta} - \boldsymbol{\mu})^\top p(\boldsymbol{\theta} \mid D) d\boldsymbol{\theta}.$$

Another example is the computation of the posterior predictive density for on an unseen data point x^*:

$$p(y^*|D, x^*) = \int p(y^*|\boldsymbol{\theta}, D, x^*)p(\boldsymbol{\theta} \mid D) d\boldsymbol{\theta}. \tag{1.2}$$

All these integrals can be generalised as integrals or expectations with respect to the posterior density

$$\mathrm{E}_{p(\boldsymbol{\theta}|D)}(I(\boldsymbol{\theta})) = \int I(\boldsymbol{\theta}) p(\boldsymbol{\theta} \mid D) d\boldsymbol{\theta}, \tag{1.3}$$

where $I(\boldsymbol{\theta})$ is the object of interest or integrand.

The central computation for Bayesian inference is performing the integration in Equation (1.3). The complexity of which grows with the dimensionality of the parameter space being integrated, even more so when those integrals do not have a closed form solution. If we could generate draws from the posterior, $\boldsymbol{\theta}_s \sim p(\boldsymbol{\theta}|D)$), the integrals can be approximated by Monte Carlo (MC) as

$$\mathrm{E}_{p(\boldsymbol{\theta}|D)}(I(\boldsymbol{\theta})) \approx \frac{1}{K} \sum_{s=1}^{S} I(\boldsymbol{\theta}_s). \tag{1.4}$$

Since it is not easy in general to draw sample from the posterior distribution, approximate methods find an approximate density $q(\boldsymbol{\theta})$ replacing the posterior such that it is either possible to solve the integrals analytically or it is easy to generate MC draws from the approximate density. This sets up the need to study the various approximate inference techniques available in literature.

1.3 Inference Methods

Bayesian inference methods can be divided in two main categories:

- Simulation-based, such as Hamiltonian Monte Carlo (HMC) and importance sampling (IS) , which generate samples in the parameter space, which are then used for further computations.
- Approximation based, where one places a tractable distribution in place of the true posterior. Techniques such as Laplace approximation, expectation propagation (EP) and variational inference (VI) fall in this category. Having a Gaussian as the tractable distribution is the most common and prevalent choice.

1.4 Hierarchical Models

Many categories of objects can be organised under subordinate or superordinate classes: e.g. birds, cats, dogs, crocodiles are all animals, but cats and dogs can be further grouped into 'mammals'. Similarly mammals like leopards, tigers, lions and cats are part of a 'Felidae' family, while deers, elks, moose form part of the 'Cervidae' family. Hierarchical models can capture the shared latent structure among observations commonly found in many real datasets.

As hierarchical models seek to find out the shared characteristics of systems while filtering away the irrelevant details, they can be of help in making better predictions on unseen new systems. A hierarchical model can make a more informed prediction about the quality of a hypothetical seventh machine, than the alternative separate model and pooled models.

Posteriors in hierarchical models often have a complex shape and geometry and provide a difficult test for any algorithm which aims to improve and produce accurate posterior computations.

This thesis lays a special emphasis on Bayesian hierarchical models. For example, publications III–IV use the eight schools model and radon model presented by Rubin [1981] and Lin et al. [1999] respectively, as case studies for accurate posterior approximation. These are also covered in details in related papers: [Yao et al., 2018, Huggins et al., 2019], which also served as motivation for the work done in this thesis.

1.5 Probabilistic Programming

Bayesian inference requires the modellers to approximate the posterior on account of intractability and computational constraints. Computational efficiency forces the user to think of a model so that model-specific derivation procedures can be handled and do not become overly complex. This conflation of model and inference is not desirable. In response to this restrictions, probabilistic programming has emerged as a set of tools allowing to specify models and then solve them by performing automated Bayesian inference. Ideally a user of such tools should be able to specify the probabilistic models with some lines of code without having to worry about the inference procedure. For a much more encompassing, and complete definition of probabilistic programming, see book by Goodman and Stuhlmüller [2014].

Recently, interest in probabilistic programming language (PPLs) has grown fast and Automatic differentiation variational inference (ADVI)/Black box variational inference (BBVI) along with Hamiltonian Monte Carlo (HMC) have emerged as the inference engines. The specifics of ADVI algorithms are described later.

Some prominent modern frameworks include Stan [Team, 2021], PyMC [Salvatier et al., 2016] MXFusion [Dai et al., 2019], TensorFlow Probability [Dillon et al., 2017], Pyro [Bingham et al., 2019]. Some older PPLs include Church [Goodman et al., 2008], Anglican [Wood et al., 2014], Infer.net [Minka et al., 2018].

1.6 Objectives and Scope

This thesis comprises of four publications: Publication I–IV, and this introduction. Each publication either uses variational inference as the tool for doing Bayesian inference or studies its properties from a more theoretical perspective or diagnoses and highlights the challenges and problems with variational inference in its current methodological form. This dissertation addresses four research questions presented here in this section. This section also describes how the research questions are related to the publications constituting this thesis.

Research Question 1: *Multi-class classification problems are seen in many fields. Problems like recommendations, object recognition, speech recognition can be seen as cases of multi-class classification. With ever increasing dataset size, it is becoming common for datasets to have a huge number of categories / classes. Consider the dataset EUR-lex which contains 57k English EU legal documents, each of which are tagged with one of 4.3k concepts / labels. Gaussian Process models allow to incorporate prior information and provide uncertainty estimate which could be useful when making decisions based on model predictions. Multi-class classification with Gaussian Process models use the softmax function which provides probabilities over classes by taking as input the latent functions for each class modelled by a GP, coupling together all link functions at once. This makes inference algorithms hard to scale when the number of classes is very large. A recently introduced approximation of softmax function factorises the likelihood such that each factor only involves two latent functions. Can we make use of stochastic approximations of softmax function to make inference tractable and scalable ?* Research question 1 is addressed in Publication I, which uses recently introduced approximations of softmax function combining it with the well known variational inducing point framework. These approximations have additional variables which when optimized can tighten the lower bound to softmax function. The inference is intractable, so a variational approximation is introduced, resulting in a single variational objective which can be written as a double sum over data points and classes. This helps GP models to scale where the number of classes and data points is large.

Research Question 2: *In many applications in real world, we are trying to model an unknown function, but it is either difficult / expensive to get its value at a location, but rather it is possible to know which location has a higher value in comparison to another location. Such a setting, where it is only possible to make queries to an unknown function can only be done in pair of points or 'duels' naturally occurs in recommendation systems, A / B testing etc. For example, a user of a movie streaming service might* *find it easier to pick his favorite movie from a list of movies in place of assigning them numerical scores. Although, some previous works have provided approaches for dealing with preferential feedback for two points, is it possible to extend it to a batch setting, where more than two points in the space are compared ?*

Publication II uses the factorised likelihood as the key tool in presenting the Preferential Batch Bayesian Optimisation framework, where choices are ranked in place of being assigned numerical scores. Three different acquisition functions are presented. The work uses two different parameterizations of the Gaussian approximations, one using a structured diagonal covariance matrix and the other using a full rank covariance matrix and compares their performance against each other and other inference methods such as Expectation Propagation and HMC. Experiments are set up to compare different inference methods and acquisition functions for different values of batch size. While Chapter 2 gives an introduction of variational inference, its application on GP models is presented in Chapter 3 which forms the background for these two publications.

Research Question 3: *Black box variational inference has emerged as a promising alternative to MCMC. It has been made scalable and widely applicable by application of stochastic optimization where the gradients can be estimated using MC draws and mini-batching. The performance of the stochastic optimization algorithm is paramount to the success of BBVI. How can we diagnose if stochastic optimization has worked well for BBVI ? This comes with some sub-problems. Can we design a better stopping rule than just running it for a fixed amount of iterations ? Can we diagnose if the optimization has failed to converge ?*

Publication III presents an algorithm for diagnosing convergence and improving the quality of final solution by averaging the iterates only after a certain iteration count. While this framework improves the performance of the stochastic optimization algorithm to improve the quality of posterior approximation, measured in terms of distance between first and second moment, the publication shows even this might not be enough as the dimension of posterior increases. The framework can also diagnoses convergence issues for a single optimizer and is also useful for detecting multi-modality in posterior using multiple optimizer runs. Chapter 2 presents a background into black box variational inference and discusses the algorithm in detail.

Research Question 4: *Black box variational inference requires users to make decisions from a wide variety of choices: divergence measure, approximating family. This is made possible because the integrals involved for computing objectives and their gradients can be approximated with finite*

average over MC draws. How does the finite sample bias affect optimization for the choice of divergence measure and approximating family? Several different divergences have been proposed claiming benefits over others, but how feasible it is to optimise for objectives associated with these divergences algorithmically and practically in high dimensions or difficult posterior geometry ?
Publication IV utilizes a recently introduced Pareto-$\widehat{k}$ diagnostic to analyse how finite sample bias resulting due to MC approximations of integrals affect stochastic optimisation algorithms for different divergence objectives with different approximating distributions. The analysis that while mass covering divergence measures offer improvements over the canonical exclusive Kullback-Leibler divergence in theory, in practice the bias for these divergence measures is much larger, making the stochastic optimization extremely challenging and thus failing to converge to the optimum.

The bias goes higher as posterior dimension increases. These results challenge the idea of using mass-covering divergences to obtain variational approximations in high dimensions. Based on this analysis with $\widehat{k}$, the publication finally gives some recommendations to the user about how to make these choices.

1.7 Structure of Thesis

Publication I–II, use variational inference for applications. Publication IV deals with identifying the algorithmic issues with canonical Black Box VI and suggesting ways to diagnose and ameliorate them to some extent, while Publication III looks at the robustness of SGD algorithm for optimising VI divergences commonly used in literature. All the publications are of methodological or analytical nature, either bringing improvements over current methods, solving scalablity challenges or then safeguarding the use of VI algorithms. Chapter 2 gives an introduction to variational inference, providing the theoretical background for all the research questions and presenting contributions of III and IV. Chapter 3 discusses Gaussian process models briefly and then focuses on application of variational inference to Gaussian process models as done in Publication I and II. Chapter 4 provides a one page summary of each publication. Finally, Chapter 5 concludes the introductory part of the thesis by discussing some relevant recent research by others as well as limitations and directions for future research.

2. Variational inference: Review and recent advances

This chapter gives a background on theory of variational inference starting from algorithms in its early application to new algorithms in more contemporary research. First, sections 2.1-2.3 provide some general background and motivation. Section 2.4- 2.5 discuss some popular different VI algorithms roughly in a chronological order, which are either used in one of the publications or important in this context. Section 2.6-2.7 provide the background specifically for research questions 2 and 3. Section 2.8, 2.9, 2.10, 2.11 analyse research question 3 and presents contribution of Publication III. Sections 2.12, 2.13, 2.14 analyse research question 4 and presents contribution of Publication IV. Section 2.15 discusses the software packages containing the tools developed in Publications III and IV.

2.1 Background and History

Variable Quantities called 'functional' were thought of at least three centuries ago by Euler and others when they tried to mathematically formulate several problems occurring in mechanics, geometry and physics. A 'functional' is a function where the variable of the function is a function itself. This means that it assigns a real definite value to a function belonging to some class. Quoting the example problem from the book 'Calculus of variations' by Gelfand et al. [2000] used to motivate the topic; Consider all possible paths a particle can take from point A to point B in a two dimensional plane. The particle has a definite velocity $v(x, y)$ for each point in the plane. The time taken by the particle to reach point B along each path can be considered as a *functional* of the particular path. A relevant question then could be which path takes the least time. The branch of mathematics for finding the maxima and minima of functional is called 'calculus of variations'. An interesting and important such variational problem can be stated as: Among all the curves of length l, find the curve enclosing the maximum area. This problem was solved by Euler and the answer is the circle. Euler found a way of reducing the problem of finding the extrema of a functional to the more familiar calculus problem of finding the extrema of a function of n variables. He then obtained the exact solution by taking the limit : $\lim n \to \infty$.

2.2 Modern history of variational inference

One of the first applications of variational methods in Bayesian inference was inspired from work in physics Parisi [1988], and it was the approximation of an intractable distribution with a simple approximating distribution having some factorial form. The intractability problem can emerge from one of the three sources:

- it may happen that the target distribution does not have a closed analytical expression, which generally happens because the normalization constant (marginal likelihood) integral does not have a closed form,
- it might be the case that it is NP-hard to evaluate the distribution in the worst case.
- thirdly, even when evaluation has polynomial time, the power of the polynomial might be too high, making the evaluation extremely expensive for contemporary computers.

Early researchers favoured using simple approximating distributions having some factorial form because they were easy to interpret, and more importantly yielded a tractable approximation. The solutions were derived by hand using calculus of variations technique with Lagrange multipliers, quite similar to a standard optimization problem. Jordan et al. [1999] provides a tutorial on the early advances in VI.

The key idea in VI is to find an approximate distribution, which is close to the target distribution. In context of Bayesian inference, the target distribution is the posterior distribution. The approximate distribution should belong to a family of tractable distributions for example Gaussian density, such that resulting approximate integrals discussed in Chapter 1 are easier to estimate. These approximating integrals are also sometimes referred as variational expectations, which are much easier to handle than the original expectations since it is easy to get a MC estimates of variational expectations, because it is much simpler to generate draws from the approximate density. The measure of 'closeness' can be defined by divergences, the most prominent of them being the exclusive Kullback-Leibler divergence. For continuous distributions this is given as

$$\mathrm{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|Y)) = \int \ln \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} q(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{2.1}$$

This divergence is so popular and commonly used, that if there is no prefix before KL divergence, most authors refer to 'exclusive KL' divergence (also known as reverse KL). This is an important point since KL divergence is an asymmetric measure, i.e. in general KL$(q||p) \neq$ KL$(p||q)$. The KL divergence in the other direction KL$(p||q)$ is referred to as inclusive KL (also forward KL).

VI with exclusive KL divergence provides a lower bound on the marginal likelihood, commonly known as the evidence lower bound (ELBO), which can be used as a proxy objective to optimize hyperparameters.

Early efforts focused almost exclusively on using the exclusive Kullback-Leibler (KL) divergence. More recently, many other algorithms for minimizing other statistical divergences have been proposed as alternatives by researchers, such as :χ^2-divergence [Dieng et al., 2017], α-divergenceHernandez-Lobato et al. [2016], inclusive KL divergence [Prangle, 2019, Naesseth et al., 2020], adaptive f-divergence[Wang et al., 2018]. How the divergences differ theoretically and practically is explained in Publication IV and discussed in the later sections of this work.

In modern variational inference, the algorithms to minimize these divergence measures do not use calculus of variations techniques. Instead, they have been replaced with automatic differentiation (abbreviated as autodiff) frameworks, which offer much more representation capacity for the approximating distributions. This is because, the gradients needed for optimisation do not need to be analytical but can be approximated with Monte Carlo estimates and mini-batching if the dataset is large. Inspired by research in physics [Thouless et al., 1977, Parisi, 1988], the early application of VI was focused on Bayesian neural networks [Peterson and Anderson, 1987] and probabilistic inference in graphical models [Ghahramani, 1994].

Nowadays, VI is almost as popular as MCMC as inference method. This thesis is also a step towards safe, robust application of VI for a wide class of models. Some of the popular and prominent algorithms emerging in VI literature are given in Table 2.1.

2.3 Integration through optimization

I give here a more general introduction to variational inference. Let $p(\boldsymbol{\theta}, Y)$ be the joint distribution of a probabilistic model, where $\boldsymbol{\theta} \in \mathbb{R}^D$ is a vector of model parameters and Y is the observed data. In Bayesian analysis, the posterior $p(\boldsymbol{\theta} \mid Y) \propto p(Y \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$ is typically the object of interest, but most posterior summaries of interest are not accessible because the normalizing integral, in general, is intractable. Variational inference approximates the exact posterior $p(\boldsymbol{\theta} \mid Y)$ using a distribution $q \in \mathcal{Q}$ from a family of tractable distributions $\mathcal{Q}$. The best approximation is determined by minimizing a divergence $D(p \parallel q)$, which measures the discrepancy between p and q:

$$q_{\lambda^*} = \arg\min_{q_{\boldsymbol{\lambda}} \in \mathcal{Q}} D(p \parallel q), \tag{2.2}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^K$ is a vector parameterizing the variational family $\mathcal{Q}$. Thus, the properties of the resulting approximation q are determined by the choice of variational family $\mathcal{Q}$ as well as the choice of divergence D. In cases where the optimization is stochastic in nature, there is no fixed point solution and, the solution then also depends on the type of optimization algorithm and algorithm parameters.

The family $\mathcal{Q}$ is often chosen such that quantities of interest (e.g., moments of q) can be computed efficiently. For example, q can be used to compute Monte Carlo or importance sampling estimates of the quantities of interest. We here introduce importance sampling since it is a recurring concept in this thesis.

2.3.1 Connections with importance sampling

Let $w(\boldsymbol{\theta}) := p(\boldsymbol{\theta}, Y)/q(\boldsymbol{\theta})$ denote the density ratio between the joint and approximate distributions. For a function $\phi : \mathbb{R}^D \to \mathbb{R}$, the *self-normalized importance sampling* (SNIS) estimator for the posterior expectation $\mathbb{E}_{\boldsymbol{\theta} \sim p}[\phi(\boldsymbol{\theta})]$ is given by

$$\hat{I}(\phi) := \sum_{s=1}^{S} \frac{w(\boldsymbol{\theta_s})}{\sum_{s'=1}^{S} w(\boldsymbol{\theta_{s'}})} \phi(\boldsymbol{\theta_s}),$$

where $\theta_1, \ldots, \theta_S \sim q$ are independent. Importance sampling estimates allow for computation of more accurate posterior summaries and to go beyond the limitations of the variational family. For example, it makes it possible to estimate the posterior covariance even when using a mean-field variational family. The SNIS is an important quantity of interest which is discussed in Publications III–IV. The SNIS estimator is consistent but has a finite sample size bias of $O(1/S)$ [Owen, 2013].

Often due to mismatch between true density and approximation, the density ratios will have a highly right-skewed distribution. This is explained in detail in Section 2.8

Now we introduce the different flavours of VI, starting with classical VI to modern black box variational inference (BBVI).

2.4 Coordinate-Ascent(Classical) variational inference (CAVI)

The standard recipe for this classical kind of VI is to use the exclusive KL divergence and the mean field Gaussian as the approximating family. The optimization problem is well specified and corresponds to:

$$q_{\lambda^*} = \arg\min_{q_\lambda \in \mathcal{Q}} \mathrm{KL}(q \parallel p).$$

Table 2.1. Prominent algorithms in VI literature with their characteristics

Reference	Popular Name	optimization	Models	Remarks
Classical VI				
[Beal, 2003]	VB	Analytic	conjugate	-
[Honkela et al., 2010]	-	Natural Gradient	non-conjugate	NSSM, NFA
[Hoffman et al., 2013]	SVI	Stochastic Natural Gradient	conjugate	-
Others				
[Winn and Bishop, 2005]	VMP	Analytical	conjugate	-
[Knowles and Minka, 2011]	NC-VMP	Analytical	Non-conjugate	-
Online VI				
[Sato, 2001]	Online VB	Analytical	conjugate	approximating family: MoG
[Broderick et al., 2013]	Streaming VI	analytical	conjugate	application on LDA models
Black Box VI				
[Challis and Barber, 2013]	-	LBFGS	non conjugate	-
[Titsias and Lázaro-Gredilla, 2014]	DSVI	Stochastic Gradient with MC	non-conjugate	Variable selection
[Ranganath et al., 2014]	BBVI	Stochastic gradient	non-conjugate	uses score gradients
[Kucukelbir et al., 2015]	ADVI	Stochastic gradient	non-conjugate	reparameterised gradients
[Hernandez-Lobato et al., 2016]	α-BBVI	Stochastic gradient	non-conjugate	-

where λ corresponds to the parameters of the approximating distribution. As explained earlier, this problem is not solvable, since we do not know the posterior itself, the problem is converted to an equivalent optimisation problem where a functionally equivalent quantity ELBO denoted by $\mathcal{L}$ is maximized. After some manipulations, the following relation is obtained:

$$\mathrm{KL}(q \parallel p) + \mathcal{L} = \log p(\mathbf{Y}). \tag{2.3}$$

Since $p(\mathbf{Y})$ is a constant irrespective of the approximating density one chooses, minimizing one of the terms on left is equivalent to maximizing the other, and the problem in its equivalent formulation is given as:

$$q_{\lambda^*} = \arg\max_{q_\lambda \in \mathcal{Q}} \mathcal{L}.$$

This is solved analytically using gradient based optimization algorithms. Since the variational family is mean field Gaussian in this case, the idea here is that we can optimize the ELBO with respect to a single variational factor at each iteration, keeping others fixed. The optimal density $q^i_\lambda(\boldsymbol{\theta})$ for the i-th variational factor is proportional to the exponentiated expected log of the conditional density,

$$q^i_\lambda(\boldsymbol{\theta}) \propto \exp\Big(\mathrm{E}_{q^{-i}}[\log p(Y, \boldsymbol{\theta})]\Big), \tag{2.4}$$

where the q^{-i} denotes all factors excluding the i-th factor $q_\lambda(\boldsymbol{\theta})$. Once the variational factor i is updated, the ELBO is recomputed with the updated factor and $i+1$ factor is computed with the same formula with the updated i-th variational factor.

2.4.1 ELBO and posterior multimodality

The ELBO in general is a non-convex function of the flattened variational parameter vector:$\boldsymbol{\lambda} = (\boldsymbol{m}, \boldsymbol{C})^\top$ [Blei et al., 2017], it is even non-smooth in general because the entropy term in its formulation is a non-smooth function [Domke, 2020] of $\boldsymbol{C}$, the covariance matrix for a location-scale family. This means it is sensitive to initialization, even in algorithms using analytical updates and is likely to get trapped in a local maxima, which is often the case in practise. In practice it has been observed that, there exist many models which have multiple posterior modes. It is conjectured that, the existence of multiple posterior modes can happen due to model mis-specification and overparameterization. This thesis and publications III and IV also keep this conjecture as a key theme for motivation and analysing results and deriving conclusions. Some recent work Zhang and Blei [2021] also point to this as a possible explanation for 'no-best phenomenon' empirically observed by researchers in topic modelling. To press the point further, multimodality is often seen in mixture models due to label-assignment switching phenomenon (many equally plausible explanations of data-generating process). For such models it is often considered sufficient to capture just one of the modes since, they have identical properties and can be all equally suited for prediction. However,there are many models where the modes can have very different properties and explain the data in very different ways. One mode for example might show that the observation noise is high while the other mode might have lower observation noise and high signal variance from Rasmussen and Williams [2006] Chapter 5. There are also models which may have tough posterior geometry such as Neal's funnel density [Neal, 2003] or multiple minor modes, all of which pose problems to VI. While these problems also exist for MCMC, recent innovations like dynamic HMC, NUTS [Hoffman and Gelman, 2011] sampling are more robust and easier to diagnose, for example with divergences. Such robustness and diagnostics are lacking from current VI implementations and serve as a motivation for this thesis.

When using CAVI the ELBO is monotonically increased to a local maximum and the ELBO on the whole dataset is monitored, which can often be expensive to evaluate on large datasets. To ameliorate this problem, it has been suggested to monitor the ELBO over a much smaller held out test set. However, this proxy objective is not guaranteed to increase monotonically. Hence, assessing convergence can be hard irrespective if the optimization is stochastic or deterministic.

2.5 Stochastic variational inference

The main paper which introduced SVI to the community was by Hoffman et al. [2013] and can be thought of as the first work which did general Bayesian inference with variational inference, solving the optimization problem with stochastic optimization [Robbins and Monro, 1951], and in the process making VI scale to larger datasets than ever before. The 'stochasticity' is due to computing gradients and objectives using 'minibatches' of data. This algorithm was successfully applied to the field of topic modelling for document collections. This paper used natural gradients and did not use Monte Carlo estimates for ELBO quantities and gradients, which was later introduced in many papers in 2014 [Ranganath et al., 2014, Titsias and Lázaro-Gredilla, 2014, Rezende and Mohamed, 2015]. So, the stochasticity was limited to the 'mini-batching' source only.

A common and generic kind of model found in Bayesian statistics involves: observations $Y = y_{1:N}$, some latent global variables $\boldsymbol{\beta}$, some latent local variables, one for each datapoint $z = z_{1:N}$ and some fixed parameters. Each one of the observation y_n, z_n could be a collection of random variables. The joint distribution is a product of a global density and product of local terms.

$$p(Y, z, \boldsymbol{\beta} \mid \boldsymbol{\alpha}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(y_n, z_n \mid \boldsymbol{\beta}), \tag{2.5}$$

where α is a vector of hyperparameters controlling the global hidden variables β. The distinction between the two set of latent variables is that each observation and local variable are independent of all other observations and local variables given the global latent variables

$$p(y_n, z_n \mid \boldsymbol{\lambda}, x_{-n}, z_{-n}) = p(y_n, z_n \mid \boldsymbol{\lambda}), \tag{2.6}$$

while the global latent variables have a prior on them: $p(\boldsymbol{\lambda})$ which could well be correlated such as a multivariate normal distribution with a non-diagonal covariance matrix.

A popular example is Bayesian mixture of Gaussians. The local variables are the means and variances of mixture components, while the mixture coefficients can be considered as global variables.

It is also assumed that both the prior density $p(\boldsymbol{\lambda})$ and the complete conditional $p(z_{n,j} \mid y_n, z_{n,-j}, \boldsymbol{\lambda})$ belong to the exponential family. Many useful and popular models in statistics and machine learning literature are examples of such models. These include: hierarchical linear regression models, factorial models, probabilistic matrix factorisation models like recommendation engines, hidden Markov models (HMM) used in speech and signal processing. SVI uses natural The natural gradient is computed on a small subset of data and is then scaled by N/B where N and B is the number of datapoints, and batch size respectively. The natural gradient [Amari, 1998] ensures that the optimization happens in Riemannian space and not in Euclidean space, avoiding the pathologies associated with Euclidean space optimization.

This gradient is used to update the global variational parameters. The local variational parameters which are implicit functions of global variational parameters, are then updated in expectation-maximization algorthm style [Dempster et al., 1977]. This method still requires some restrictions on the form of the approximating distribution(mean-field) and the complete conditionals.

2.5.1 Amortized variational inference

Given the problem where we want to do inference over the local latent variablez_i for a datapoint Y_i, but we also want to do inference over all the latent parameter/datapoint combinations. Since we are interested in distributions over point estimates, z_i is a normal distribution $z_i = \mathcal{N}(\mu_i, \sigma_i)$, and suppose we have found out the optimal parameters μ_i^*, σ_i^* for the i-th datapoint, is there a way we can predict the optimal parameters for a new datapoint? If we make a reasonable assumption here, that points closer in data space should also have similar latent parameters, that means we can obtain a deterministic mapping, which can be learnt using a multi layer perceptron (MLP) or a feedforward dense network. It should be noted that we hope this function is not too wiggly, otherwise the learnt mapping will not generalize well to unseen data points. This idea was famously used in variational auto-encoder Kingma and Welling [2014], which is now one of the de-facto generative modelling technique along with Generative Adversarial Networks (GAN), although the author believes the idea was first used by Lawrence and Quiñonero Candela [2006]. The complexity of the neural network is generally tuned empirically. This idea was also used in Publication I.

2.5.2 Automatic differentiation variational inference

ADVI is the key topic of this thesis and extensively used in publications I,III,IV. It is also known by other names like black box variational inference (BBVI) and doubly stochastic variational Inference (DSVI). They were proposed by different authors and have some minor, subtle differences. It is more general than the SVI method 2.5 introduced by Hoffman et al. [2013] as it does not require closed form coordinate updates. However, for the purpose of this thesis, we consider them to be the same. Originally, BBVI introduced by Ranganath et al. [2014] used score gradients requiring some elaborate control variate schemes to reduce the otherwise impractically high variance of gradients. ADVI [Kucukelbir et al., 2015] and DSVI [Titsias and Lázaro-Gredilla, 2014] both proposed to use reparameterised gradients, which were shown to have much lower variance and are the most preferred algorithms. Though commonly used, reparameterised gradients do not always have lower variance than score gradients. ADVI offers great flexibility and scalability since it involves computing gradients and objectives using MC samples and occasionally mini-batching. A key point to observe here is that while in the influential SVI paper by Hoffman and Gelman [2011] discussed earlier in 2.5, the only source of stochasticity is mini-batching, for ADVI the generic stochasticity is gradient computation by MC samples, and mini-batching is second source of stochasticity which is optional. Not all probabilistic models will have an objective which can be written as a sum over datapoints.

The key idea of BBVI is stochastic optimization, i.e. the gradients are expectations which are approximated by their MC estimates. It is hoped that they are approximately unbiased, but as we discuss later in PublicationIV, in pre-asymptotic regime this is rarely the case, and often the bias is so large that the solution never converges to the true solution. Another key-insight here is that the optimizer has two phases in its trajectory: when it is far away from the solution, and a stationary phase when it is oscillating around the solution. This is covered in Publication III of this thesis. Due to lack of space, some additional information was not described in detail in these publications, which we do here in this chapter.

2.6 Variational families

As discussed earlier, it was not possible to use a wide range of approximating families earlier due to intractibility. This has been made possible by advances made on automatic-differentiation software powered ADVI described in section 2.5.2, which provides non-analytical gradients for a wide range of distributions. This has made possible to use complex distributions as approximating distributions, some of which we give here. Let q_λ be an approximating family parameterized by a K-dimensional vector $\lambda \in \mathbb{R}^K$ for D-dimensional inputs $\theta \in \mathbb{R}^D$. Typical choices of q include mean-field Gaussian and Student's t families [Blei et al., 2017, Huggins et al., 2019], full and low rank Gaussians [Ong et al., 2018, Kucukelbir et al., 2015], mixtures of exponential families [Locatello et al., 2018, Miller et al., 2017], and normalizing flows [Rezende and Mohamed, 2015]. In Publication III, we mainly focus on full-rank approximations in the experiments, though the theory proposed is generally applicable to other approximate distributions. In Publication IV, we mainly use mean-field and normalizing flow families. Mean-field families assume independence across the D dimensions: $q(\theta) = \prod_{i=1}^{D} q_i(\theta_i)$, where each q_i typically belongs to some exponential family or other simple class of distributions. We take the q_i to be either Gaussian or Students' t with ν degrees of freedom. The distribution for the ith dimension has a mean parameter: μ_i and a scale parameter: σ_i. We parameterize the family by the means and log scales $\phi_i = \log \sigma_i$, so $\lambda = [\mu_i, \phi_i]_{i=1}^{D}$. For computational reasons, we transform the standard deviation to the unconstrained log space, so that the optimization problem becomes easier. These families are often chosen for computational reasons: they are typically easy to optimise, the densities are not expensive to compute, and drawing samples from them is straightforward. However, their lack of flexibility makes them too simple for many real world problems with complex posteriors.

Mixtures Mixtures of Gaussian or t distributions are common in statistics because they can model more complex datasets. They have not succeeded as variational families in BBVI mainly due to the difficulty of optimization and poor scaling to high dimensions. These models are particularly prone to overfitting and in general difficult to optimise. Some recent papers have suggested ways to improve their optimization, as in boosting VI [e.g. Guo et al., 2017, Locatello et al., 2018, Miller et al., 2017, Dresdner et al., 2021] To each component's individual parameters they add mixing weights to optimise. One also needs to decide the number of components C to fit a priori, which is usually not a trivial problem.

Full rank families Full rank families no longer assume independence across dimensions meaning that they can effectively capture correlated distributions with non-zero correlation. Consider the multivariate Gaussian distribution

$$q(\theta) = \mathcal{N}(\theta \mid \mu, \Sigma), \quad (\mu \in \mathbb{R}^D, \Sigma \in \mathbb{R}^{D \times D}),$$

which has a total of $D(D+3)/2$ parameters – in contrast to the $2D$ parameters of a mean field Gaussian. The quadratic scaling with increasing dimension makes these families rather difficult to optimise in high dimensions. We can also define a low rank approximation by decomposing Σ as $\Sigma = U + VV^\top$, where $U \in \mathbb{R}^{D \times D}$ is a diagonal matrix and $V \in \mathbb{R}^{D \times d}$, with $d \ll D$ [Ong et al., 2018]. This results in fewer parameters, i.e. $Dd + 2D$ parameters but leaving us with a problem of choosing the hyperparameter d at the price of less flexibility. While mixtures of low rank approximations can be a good compromise between richness and scalability, they require additional hyperparameters like mixture weights and number of components to be optimised or chosen apriori which significantly detracts from the benefits of using BBVI. Having better initialisation and putting a prior on the number of components is still an active area of research.

Normalizing Flows Normalizing flows provide more flexible families that can capture correlation and non-linear dependencies. A normalizing flow is defined via the transformation of a probability density through a sequence of invertible mappings. By repeatedly applying the change of variables formula, the initial density *flows* through the sequence of transformations [Rezende and Mohamed, 2015]. If we use an invertible, smooth mapping $f : \mathbb{R}^D \to \mathbb{R}^D$ with inverse $f^{-1} = g$, to transform a random variable θ with distribution $q(\theta)$, the resulting random variable $\theta' = f(\theta)$ has a distribution:

$$q(\theta') = q(\theta) \left| \det \frac{\partial f^{-1}}{\partial \theta'} \right| = q(\theta) \left| \det \frac{\partial f}{\partial \theta} \right|^{-1}. \tag{2.7}$$

By composing several maps, a simple distribution such as a mean-field Gaussian can be transformed into a more complex one. Here we use planar flows [Rezende and Mohamed, 2015] and non-volume preserving (NVP) flows [Dinh et al., 2017].

Planar flows are defined as transformations given by :

$$f(\theta) = \theta + uh(w^T\theta + b), \tag{2.8}$$

where h is a smooth, non linear element-wise function.

NVP flows are defined as

$$f(\theta_{1:d}) = \theta_{1:d},$$

$$f(\theta_{d+1:D}) = \theta_{d+1:D} \odot \exp(s(\theta_{1:d})) + t(\theta_{1:d}),$$

where s and t stand for scale and translation, and are functions from $\mathbb{R}^d \to \mathbb{R}^{D-d}$, and $\odot$ is the elementwise (or Hadamard) product operator.

In our experiments in Publication IV, we stack fully connected neural networks for both scale s and translation t operators. After the final layer of s we place a hyperbolic tangent non-linearity.

The advantage NVP offers over other types of flows is that this construction allows functions s and t to be as complex as one desires, as the Jacobian of this transformation is lower triangular [Dinh et al., 2017].

2.7 Optimization with Monte Carlo draws and Mini-batching

All the divergences used in the publications of this thesis can be seen as special cases of f-divergence which we define here. For a convex function f satisfying $f(1) = 0$, the f-divergence is given by

$$D_f(p \parallel q) := \mathbb{E}_{\theta \sim q}\left[f\left(\frac{p(\theta \mid Y)}{q(\theta)} \right) \right].$$

The exclusive Kullback–Leibler (KL) divergence corresponds to $f(w) = -\log(w)$, the inclusive KL divergence corresponds to $f(w) = w\log(w)$, the χ^2 divergence corresponds to $f(w) = w^2/2$ and, finally, general α-divergences correspond to $(w^\alpha - w)/\alpha(\alpha-1)$. We also consider the tail-*adaptive* f-divergence, proposed by Wang et al. [2018] in publication IV. We note in some literature, inclusive and exclusive KL divergences are known as forward and reverse KL divergence respectively (from left to right).

Though earlier we only talked about ELBO and exclusive KL divergence relations, it has been shown in numerous works that minimizing the f-divergence is equivalent to minimizing the loss function

$$\mathcal{L}_f(p \parallel q) := \mathbb{E}_{\theta \sim q}[f(w(\theta))]. \tag{2.9}$$

Let $L(\lambda) := \mathcal{L}_f(p \parallel q_\lambda)$ denote the loss as a function of the variational parameters denoted by the vector:$\boldsymbol{\lambda}$. Many commonly used objectives such as the ELBO [Bishop, 2006] and CUBO [Dieng et al., 2017] can also be formulated this way. The loss and its gradient can both be approximated using, respectively, the Monte Carlo estimates

$$\widehat{L}(\lambda) = \frac{1}{S}\sum_{s=1}^{S} f(w(\theta_s)) \quad \text{and} \quad \widehat{G}(\lambda) = \frac{1}{S}\sum_{s=1}^{S} g(\theta_s), \tag{2.10}$$

where $\theta_1, \ldots, \theta_S$ are independent draws from q_λ and $g : \mathbb{R}^K \to \mathbb{R}^K$ is an appropriate gradient-like function that depends on f and w.

The solution:$\boldsymbol{\lambda}^*$ is found using the stochastic optimization recursive scheme

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \eta\gamma_t\hat{\boldsymbol{g}}_t\,, \tag{2.11}$$

where $\hat{\boldsymbol{g}}_t$ is an unbiased, stochastic estimator of the gradient of the objective:$\mathcal{L}$ at $\boldsymbol{\lambda}_t$ (i.e., $\mathbb{E}\left[\hat{\boldsymbol{g}}_t\right] = \nabla\mathcal{L}\left(\boldsymbol{\lambda_t}\right)$), η is a base step size, and $\gamma_t > 0$ is the learning rate at iteration t, which may depend on current and past iterates and gradients. This optimisation routine is also popularly known as stochastic gradient descent (SGD). In practice, adaptive gradient schemes like ADAM [Kingma and Ba, 2014], RMSProp [Hinton and Tieleman, 2012], Adagrad [Duchi et al., 2011] are used. In some literature, the above equation is generalized such that the updates take place in a direction $\boldsymbol{d}_t$ where $\boldsymbol{d}_t$ is the direction for SGD update.

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \eta\gamma_t\boldsymbol{d}_t. \tag{2.12}$$

This covers cases which use history of gradient in a form of moving average or other schemes. Standard SGD then corresponds to $\boldsymbol{d}_t = \hat{\boldsymbol{g}}_t$. Adaptive method like RMSProp, for instance, keeps a moving average of the squared gradient: $l^{t+1} = \beta l^t + (1-\beta)\hat{\boldsymbol{g}}_t \odot \hat{\boldsymbol{g}}_t$ which is used to rescale the current stochastic gradient $\boldsymbol{d}^{t+1} = \hat{\boldsymbol{g}}_{t+1}/\sqrt{l^{t+1}}$, making scale of gradient less sensitive to its current iterate. This makes it possible to use a bigger step size than would be possible with simple SGD.

The noise in the gradients is a consequence of using mini-batching, or approximating the expectations using Monte Carlo estimators, or both Mohamed et al. [2019], Hoffman et al. [2013], Ranganath et al. [2014]. For standard stochastic gradient descent (SGD), γ_t is a deterministic function of t only and converges asymptotically if γ_t satisfies the Robbins–Monro conditions $\sum_{t=1}^{\infty}\gamma^t = \infty$ and $\sum_{t=1}^{\infty}\gamma_t^2 < \infty$ [Robbins and Monro, 1951]. Under Robbins–Monro-type conditions, many stochastic optimization algorithms converge asymptotically to the exact solution $\boldsymbol{\lambda}^*$ [Robbins and Monro, 1951, Li and Orabona, 2019], but any iterate $\boldsymbol{\lambda}_t$ obtained after a finite number of iterations will be a realization of a diffuse probability distribution π_t (i.e., $\boldsymbol{\lambda}_t \sim \pi_t(\boldsymbol{\lambda}_t)$). This key viewpoint helps us to analyse the behaviour of stochastic optimization algorithms in III and given in details later. SGD is very sensitive to the choice of step size since too large of a step size will result in the algorithm diverging, while too small of a step size will lead to very slow convergence.

2.7.1 Fixed step-size optimization and Iterate Averaging

If the step size becomes fixed after some iteration say t, i.e. $\gamma_{t,\cdots,T} = \gamma$, then the iterates obtained by SGD (2.12), $\lambda_{t,\cdot,T}$ form a Markov chain, which under some mild conditions will have a stationary distribution, the characteristics of which are determined by the initial position λ_t and the step size which is now fixed i.e. π_γ. This is even true when the step size is fixed after following an adaptive schedule initially, for faster convergence. As shown in Publication III, we should not expect a single iteration $\boldsymbol{\lambda}_{t+i}$ to be close to the optimal: $\boldsymbol{\lambda}^*$ in high-dimensional settings, the expected value of $\boldsymbol{\lambda}_t$ is close to $\boldsymbol{\lambda}^*$. This led us to the utilize the idea of *iterate averaging* (IA) to construct a more accurate estimate of $\boldsymbol{\lambda}^*$ given by

$$\bar{\boldsymbol{\lambda}} \equiv \tfrac{1}{T}\textstyle\sum_{i=1}^{T-t} \boldsymbol{\lambda}_{t+i}, \tag{2.13}$$

where we should aim to choose $t \geq t_0$. t_0 is the point where the optimiser has entered stationary phase. The idea of *iterate averaging* has been proposed before in Markov chain literature [Meyn et al., 2009], in context of convex optimization [Ruppert, 1988, Moulines and Bach, 2011] and more recently to improve generalization in deep learning [Izmailov et al., 2018]. The key idea here is how to determine when to start averaging. There are important details to consider here, which I explain later.

2.8 Tail-index and $\hat{k}$

A tail index can be used to determine the number of finite moments a random variable has. Estimating tail index is a common research theme in extreme value theory. In this thesis Publications III–IV, tail index is used for carrying out analysis and diagnosis following modelling and inference. Here we give a short introduction to thick tails and tail-index. A non negative variable has thick tail when the tail decays at a slower rate than the exponential function $\exp^{-t}$. The probability mass in tails as $F_w(t) = P(w \geq t)$ is asymptotically equivalent to $t^{-\alpha^*}$ as $t \to \infty$ for some positive α^*. This is formally written as

$$F_w(t) = L(t)t^{-\alpha^*}, \tag{2.14}$$

where L is a slow varying function such that $\lim_{t\to\infty} L(ct)/L(t) = 1$ for any $c > 0$. Here α^* denotes the tail index of the density ratio defined above: w, and $E[w^\alpha]$ is finite only if $\alpha < \alpha^*$. In other words, the density ratio: w has α^* finite moments. This is the motivation for using thick tailed distributions such as t density as the approximation q in literature, which makes the density ratios bounded from above theoretically. It is another matter that in high dimensions, as shown in IV, the bound can be so high that this idea stops working in practice (limited MC draws). The $\hat{k}$ used in this work [Vehtari et al., 2016, Yao et al., 2018] is an estimator for the inverse of α^*. Hill [1975], De Haan and Peng [1998] are important references proposing or comparing tail index estimators. Simsekli et al. [2019] used the tail index estimator given in Mohammadi et al. [2015] to analyse gradient noise for deep learning models on typical deep learning datasets, while I used the estimator in Vehtari et al. [2016] for the estimation of commonly used variational objectives and also divergences.

One challenge with variational inference is assessing how good of an approximation, the obtained variational approximation after optimization $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$ is to the true posterior distribution $p(\boldsymbol{\theta}|Y)$. Let $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_S \sim q_{\boldsymbol{\lambda}}$ denote draws from the variational posterior. If the proposal distribution is far from the true posterior, the weights $w(\boldsymbol{\theta}_s)$ will have high or practically infinite variance. The number of finite moments of a distribution can be estimated using the shape parameter k in the generalized Pareto distribution (GPD) [Vehtari et al., 2019b]. Theoretical and empirical results show that values below 0.7 indicate that the approximation is close enough to be used for importance sampling, while values above 1 indicate that the approximation obtained as a weighted average is very poor [Vehtari et al., 2019b].

Interestingly, we can also use $\hat{k}$ to assess the behaviour of the stochastic optimization algorithm used to obtain the VI solution. Recent work by Gurbuzbalaban et al. [2020] suggests that SGD iterates with fixed step size may converge towards a heavy tailed stationary distribution with infinite variance for even simple models (i.e. linear regression). Characterizing noise distributions with momentum based optimisers is still very active area of research. Furthermore, even in cases that don't show infinite variance, the heavy tailed distribution may not be consistent for the mean, i.e. the mean of the stationary distribution π_γ might not coincide with the mode of the objective. This would imply that some of the assumptions in Mandt et al. [2017] wouldn't hold in practice for some parameters in the optimization, which in turn would make iterate averaging unreliable. We again rely on $\hat{k}$ to provide an estimate of the tail index of the iterates (at convergence) and warn the user when the empirical tail index indicates a very poor approximation.

VI for all its success still has many challenges in the prevalent algorithms which need to be acknowledged, and possibly resolved in future research. First, we describe the challenges in more detail in the next sections, at the same time presenting ideas to fix them or atleast diagnose them for the end user. This is covered in Publications III and IV, but I summarise them here in some detail in this chapter in Sections 2.10, 2.11, 2.13 and 2.14.

2.9 Convergence for stochastic optimization in VI

While asymptotically (in the number of iterations and Monte Carlo sample size S) there may be no issues with stochastic optimization or divergence estimation, in practice black-box variational inference operates in the pre-asymptotic regime, since we do not have infinite computing resources at disposal. As we cannot run the optimization for ever, the lack of a good stopping criterion is a major challenge for these methods. The high variance in stochastic gradients makes the users go for a smaller step-size and adaptive gradient methods (which use a moving average over the past gradients) like AdaGrad, RMSProp etc. these methods may introduce bias due to premature stopping. Moreover, these optimisers may not even be consistent, which means that given an optimization scheme, if they are run for an impractically large time (infinite), the solution obtained may still not be one of the optima (the member of the variational family, which is closest to the target in terms of divergence measure chosen).

Some authors recommend monitoring the ELBO or the predictive log-likelihood of a test-set and stopping the algorithm when the relative ELBO drops below a certain threshold 10^{-2} [Kucukelbir et al., 2015] or 10^{-4} [Yao et al., 2018]. Taking such a snapshot based approach can be highly misleading. While no diagnostic is perfect, ΔELBO is highly susceptible to noise. Similarly, many strong arguments can be given against the predictive log likelihood approach which seems to be the preferred and recommended approach in literature Blei et al. [2017].

We show in Publication III, that the ELBO estimates might be so noisy that we might stop too early even when the true solution is relatively far away. In our experiments, we found the ΔELBO value oscillates around a quantity which is problem and model dependent. Having a very strict threshold might result in the condition never triggering. For some models and with some optimization schemes, even 10^{-2} might be too strict of a threshold (Publication III) and for some other models it might be too weak of a condition [Yao et al., 2018]. Even when monitoring predictive likelihood to detect convergence, there is no guarantee that it will increase monotonically and the optimum may even have a worse predictive log-likelihood than a point in its vicinity, since it is estimated only on a subset of data. It is possible to get a worse predictive likelihood on a validation set at the optimum solution(i.e.say true posterior) due to model misspecification. Another issue with this approach is that it takes away a key attractive property of Bayesian methods of not having to throw any data away from training.

Although I focus on exclusive KL divergence objective, this should broadly apply to all BBVI methods which use stochastic optimization. The problem gets only more challenging with other variational objectives such as CUBO, the objective for minimizing χ^2 divergence. In case of ELBO, it is guaranteed that the objective evaluated at any point, $\lambda \neq \lambda^*$ other than the optimum is still a lower bound to the log marginal likelihood, since MC estimation can rarely over-estimate ELBO on expectation. For objectives such as CUBO which is an upper bound over marginal likelihood, the objective estimated might be higher or lower than the log marginal likelihood, for sub-optimal variational parameters, due to huge under-estimation of objectives with polynomial dependence on density ratios. Same holds for variational objective for inclusive KL divergence. Please refer to Publication 4 Figure 2. This is a big challenge for mass-covering divergences, since in practice with MC estimation, because of downward bias, their variational objectives are not necessarily upper bounds.

For objectives associated with other mass-covering divergences, we show that the variance and bias of gradients is likely to be much higher than for the ELBO. We propose a robust VI algorithm in Publication III, which uses the perspective of seeing the optimiser as a Markov chain and then apply known diagnostics from MCMC literature to know if the optimiser (Markov chain) has converged approximately to its stationary distribution, the theory for which is backed by Dieuleveut et al. [2020].

2.10 Detecting convergence to stationarity

The stochastic process induced by the SGD iterates of a variational parameter $\boldsymbol{\lambda} : \boldsymbol{\lambda}_{t0,\cdot,T}$ is stationary if the joint distribution of any subset of the sequence is invariant with respect to identical shifts in the indices. Informally speaking, if we imagine a distribution over the region of the variational parameter space, where the optimiser tends to reside, after a point it will become stable and fixed.

Stochastic optimization with a fixed learning rate displays two distinct phases: a transient phase during which the optimization makes rapid updates towards the optimum, which is followed by a stationary phase during which iterates oscillate around the mean of a stationary distribution: $\lambda = \int \lambda \pi_{\gamma}(d\lambda)$. These phases are akin to the 'warmup' and 'mixing' phases of MCMC runs. Before describing more about the approach developed in this work, we first summarize some recent methods and algorithms to detect stationarity. One prominent line of work is based on using a statistical test to determine if a suitable function (invariant) has expectation zero by checking if the empirical mean is close to 0, under the stationary distribution of the iterates [Lang et al., 2019, Yaida, 2019]. This is the key idea behind the algorithms SASA and SASA+ by Lang et al. [2019] and Zhang et al. [2020] respectively, which have been recently proposed for detecting convergence in training of neural networks, the optimization for which is highly non-convex. Such methods should also transfer well for monitoring convergence for BBVI.

The invariant function is defined as: $\Delta =< \boldsymbol{d_t}, \boldsymbol{\lambda_t} > -\frac{\alpha}{2}||\boldsymbol{d_t}||^2$. While it is easy and computationally cheap to compute this expectation with inner products, one issue with this test is that all variational parameters are used as once, making it hard to know which variational parameter and related model parameter can be the cause of convergence issue. This can be extremely important information as shown in Publication III, where the scale parameter was found to have problematic convergence in the eight school centred model. When the model was re-parameterised to non-centred parameterization [Papaspiliopoulos et al., 2007], the convergence issue was resolved.

The above test can be improved to a multivariate hypothesis test by applying a separate hypothesis test for each of the variational parameter, with convergence triggered after all tests confirm stationarity in the same way $\hat{R}$, ESS and MCSE are checked for all variational parameters. In case the dimensionality of the variational parameters is very high, then it is also possible take a random subset of them to perform check at the cost of some robustness. The split-$\hat{R}$ diagnostic checks the whole optimization trajectory after splitting it into two parts. This approach while being computationally more expensive than the ΔELBO stopping rule is much more robust and looks at all variational parameters.

2.11 MCMC diagnostics for VI optimization

As discussed in Section 2.7.1, if the learning rate is fixed after a point, such that $\gamma_{t,\cdots} = \gamma$, then we can view the iterates $\lambda_t, \lambda_{t+1}, \cdots, \lambda_T$ as a Markov chain (MC), which under certain conditions will have a stationary distribution π_γ. As we found it in publication III, the connections between constant step size SGD and MC dynamics lead us to use convergence diagnostics for MCMC, which have received much more rigorous treatment and testing over the past two decades, compared to convergence in BBVI. The diagnostics and their formulae are given in the Table 2.2. It is important to state that we use multiple convergence diagnostics and not just one, which makes it harder to falsely detect convergence at the cost of being slightly too pessimistic and conservative. The diagnostics used for robust VI optimiation are: $\widehat{R}$ [Vehtari et al., 2019a], Effective sample size (ESS) and Monte Carlo Standard Error (MCSE). These can be read in detail in Vehtari et al. [2019a] but we describe them here briefly in context of MCMC. It is recommended to run MCMC with multiple independent chains, such that the sampler distribution should converge to the target probability. If the chains have mixed up well, then the variance of all the chains mixed together should not be much higher than the variance of individual chains. This can be done for each parameter individually alerting the user about which particular parameters are problematic for convergence (the user can consider reparameterizing those as a remedy). $\widehat{R}$ is computed as the standard deviation of a parameter denoted as: $(\hat{\mathbb{V}})^{1/2}$ taken one at a time divided by the root mean square of the separate within-chain standard deviations $(\hat{\mathbb{W}})^{1/2}$. When this number is close to 1, it means the variances are very close to each other. The variances computed above do not give the full picture since there is usually very high autocorrelation among the chains. The effective sample size (ESS) of a quantity of interest shows how many independent draws contain the same amount of information as the dependent(due to autocorrelation) sample obtained by the MCMC algorithm. The higher the ESS the better it is. To compute ESS, it is then important to know the autocorrelation values ρ_t at lag values from 1 to some lag index T where T is taken to be 100 in practice. If all the draws are independent , the ESS should come out to be JN, where J is the number of chains and N is the number of draws in each chain. The MCSE quantity can be used as a stopping rule in MCMC, i.e. when it drops below a certain threshold. Determining the threshold requires domain expertise as it is scale dependent unlike ESS. MCSE can be used to diagnose the efficacy of iterate averaging Equation (2.13) the same way it is used to determine the accuracy of empirical average of a parameter of interest:$\bar{\theta}$ as an estimator for posterior mean $\mathrm{E}[\theta]$.

These diagnostics can be used on multiple runs or even with a single run after splitting the *chain* into two equal parts. Since we do not want to compute these values too often, we only estimate them after a fixed amount of iterations W. We found $W = 200$ and $W = 100$ to be good default values. These diagnostics are coupled with iterate averaging as also recommended in deep learning and optimization literature [Garipov et al., 2018][Dieuleveut et al., 2020]. The crucial difference in our work is that iterate averaging is done only after convergence diagnostic is triggered and stopped when ESS and MCSE reach a certain threshold. The threshold values for ESS and $\hat{R}$ are less conservative compared to corresponding values from MCMC literature, while the value for MCSE was chosen empirically and taken to be of the same order as the initial step size. These diagnostics worked well with adaptive optimizers like Adagrad, ADAM, and RMSProp.

These statistics are computed for each variational parameter. In case the user is happy with point estimates without additional uncertainty estimates, the convergence check can be made only on location parameters. If convergence is not detected even after running the optimization for a given amount of iterations, we should either change the approximating family or reparameterize the model. If convergence is achieved, the user can put more trust in the results.

MCMC diagnostics is still an active area of research with recent innovations like the recently introduced: split-$\hat{R}$ and ESS for tail quantities [Vehtari et al., 2019a]. These ideas fit well in the *Bayesian workflow* scheme [Gelman et al., 2020].

2.12 Challenges with exclusive KLVI and mis-specification

Algorithms for minimizing the exclusive KL divergence are the canonical algorithm of variational inference. This is because of its tractability compared to other divergence measures and good theoretical properties of

Table 2.2. Convergence diagnostics and recommended thresholds

Diagnostic	Formulae	Recommended value
split $\hat{R}$	$(\hat{\mathbb{V}}/\hat{\mathbb{W}})^{1/2}$	1.1-1.2
ESS	$JN/(1+\sum_{t=1}^{\infty} 2\rho_t)$	20
MCSE	$\{\mathbb{V}(\lambda_i)/\text{ESS}(\lambda_i)\}^{1/2}$	0.02

ELBO with respect to maximum likelihood solution. However, the family of approximation Q is very often mis-specified, which means there is no member q^* which has zero KL$(q||p)$ divergence. It is also difficult to make a decision whether the exclusive KL divergence of the optimal KL$(q^*||p)$ is small enough for a downstream application[Huggins et al., 2019, Kuśmierczyk et al., 2019, Kuśmierczyk et al., 2020]. This has motivated application of Importance Sampling (IS) to reduce the bias from posterior summaries [Yao et al., 2018, Huggins et al., 2019]. Using the solution as a proposal for IS also provided the $\hat{k}$ as a diagnostic. While a small $\hat{k}$ is not a guarantee of a good approximation accuracy, it guarantees that the approximation posterior summaries are close to true posterior summaries after IS/PSIS corrections. Moreover, we can obtain a new proposal by computing moments from the MC draws weighted by their importance weights. This procedure can be iterated to obtain an increasingly better IS proposal [Paananen et al., 2021]. Huggins et al. [2019] gave an algorithm to bound the error in summaries such as posterior mean, posterior std. dev. and posterior covariance. While being useful, however, these bounds can still be loose and they may require computing the χ^2-divergence (and optimising CUBO) which itself might not be reliable if estimated stochastically, especially in high dimensions. This has encouraged researchers to look for alternatives mass-seeking divergences such as inclusive KL divergence, α-divergence, Renyi-divergence, which can possibly cover the posterior mass better, a claim which is contested in publication IV for high dimensional non-Gaussian posteriors.

2.13 Underestimation and Overestimation of Marginal Variances

In case where the true posterior does not lie in the same family as the approximation, the approximating posterior q^* will be different from p when the optimization has converged. If the objective divergence is chosen to be exclusive KL, then the margin al variances (and uncertainty) are typically underestimated. This is because of its mode-seeking properties. The opposite is true for inclusive KL divergence as the variances are typically overestimated IV. We write 'typically' because this is not always the case.

As shown in Yao et al. [2018]and III, for instance, in hierarchical models, like the eight schools models, a positive bias in estimation of the scale parameter will lead to overestimation in variance of other parameters.

2.14 High dimensional geometry and low overlap in typical sets

Here, we introduce the concept of 'typicality', which is central to high dimension probability distributions where normal intuition gives away. Expectation values as given in Equation (1.3) are given by accumulating the integrand over a volume of parameter space and, while the density is largest around the mode(where the density is the highest), there is not much volume there. To identify the regions of parameter space that dominate expectations we need to consider the behavior of both the density and the volume. In high-dimensional spaces the volume behaves very differently from the density, resulting in a tension. The probability mass is given by the product of the density and the volume around that point $d\boldsymbol{\theta}$, then to find regions where probability mass lies, we cannot consider regions where either is extremely low. Thus a point near the mode will have high density but the neighbourhood region around the mode has extremely small volume. Similarly far from the mode, the neighbourhood regions have very large volumes but extremely low density, the regions between the two extremes is the only place to capture probability mass. If we draw samples from a distribution in high dimensional space, it becomes extremely unlikely to draw any samples close to neighbourhood of the mode. The density values alone do not tell where probability mass is concentrated and the density values of the samples drawn from the approximate density in high dimensions will have much lower densities compared to mode. The only significant contribution to the integral in Eq. (1.3) comes from some region where neither is extremely low, known as the typical set. A much more formal definition and information theory connections is given in Cover and Thomas [2006]. Betancourt [2018] is another excellent reference to get intuition for high dimensional geometry. As the dimension increases, the typical set gets narrower, for example an isotropic multivariate Gaussian density $\mathcal{N}(0, \mathbb{I}_d)$ will be concentrated in a thin shell of radius $\sigma\sqrt{d}$. This is due to 'concentration of measure' phenomenon.

As the dimensionality of the posterior increases, it becomes more and more difficult to find a good proposal distribution for Importance sampling. This brings us to another question: how fast or slow does then, the estimation of divergence objectives(and gradients) deteriorate with increasing dimensionality compared to importance sampling. The low overlap of typical sets in high dimensions causes the estimation of objective and gradients with MC samples to have pre-asymptotic bias, which can massively slow

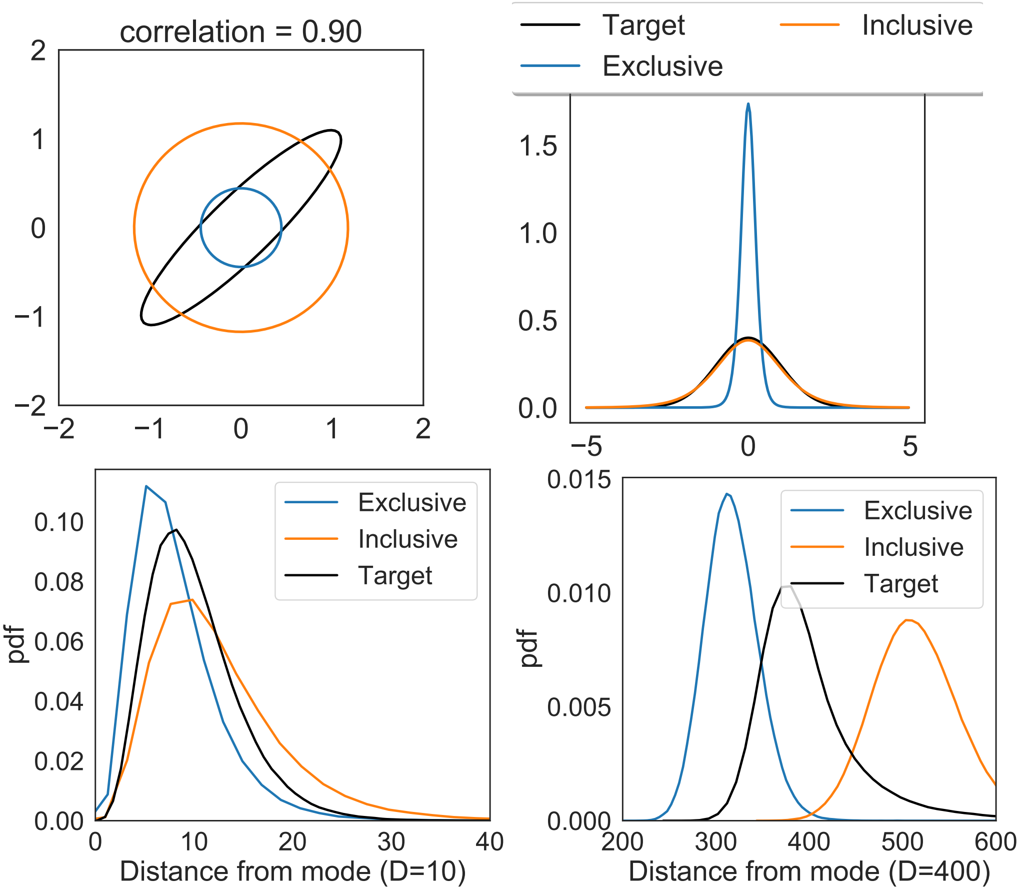


Figure 2.1. Illustration of a mean-field approximation with exclusive (mode-seeking) and inclusive (mass-covering) divergences. The target is a normal distribution and the approximate distribution is a t distribution with 7 degrees of freedom. (a) The typical 2D illustration (correlation 0.9) gives impression that the inclusive divergence would provide a better approximation. (b) The marginal distribution of the 2D illustration, showing the heavier tails of the approximate distribution. Heavier tails guarantee that the importance ratios are bounded, and thus the importance sampling estimate has asymptotically finite variance. (c,d) The marginal densities as a function of the distance from the mode in 10- and 400-dimensional examples (correlation 0.10). These demonstrate that, even for much lower correlation levels, the intuition from the low-dimensional examples does not carry over to higher dimensions: although the importance ratios are still bounded and importance sampling has asymptotically finite variance, the overlap in typical sets of the target and the approximations gets worse both for exclusive and inclusive divergences.

Table 2.3. Required finite moments to estimate divergences, for $\delta > 0$ and $\alpha > 1$ [Epifani et al., 2008, Table 2]. The last column gives the formulae required to compute the Pareto-k index to the gdpfitlw function in `arviz`

Objective	$f(w)$	Moments	Formulae
Exclusive KL	$\log(w)$	δ	$\log(\log(w) - \min(\log(w)) + 1)$
Inclusive KL	$w\log(w)$	$2+\delta$	$\log(w\log(w) - \min(w\log(w)) + 1)$
χ^2	$(w^2 - w)/2$	4	$2 \times \log(w)$
α-divergence	$(w^\alpha - w)/(\alpha(\alpha-1))$	2α	$\alpha \times \log(w)$

down stochastic optimization speed and can even result in divergences in optimization. Another ramification is that the low overlap causes the distribution over the density ratio to have a heavy right skew, implying that even after averaging over a large number of samples, most empirical estimates $\sum_{s=1}^{S} w_s$ will be smaller than the true mean. This causes the variational parameter solutions obtained after minimizing mass-covering divergences, algorithmically done by maximizing corresponding divergence objectives) like χ^2, α to be biased towards exclusive KL divergence estimates due to underestimation, as also reported by [Geffner and Domke, 2020a] empirically. As dimensions increase, the overlap in typical sets of true posterior and approximation decreases and optimizing mass-covering divergences becomes extremely challenging with standard existing stochastic optimization algorithms [Geffner and Domke, 2020b].

As explained in Publication IV, this depends on the function $f(w)$ applied to the density ratios, for the computation of objectives and gradients. If the function f is strongly concave i.e. grows sublinearly like $f(w) = \log(w)$ as in case of exclusive KL divergence, we can expect all the moments to be finite, hence the tail index value is quite small, and computation of Exclusive KL divergence with MC samples is more reliable than the corresponding IS estimate. If the function grows super-linearly, then the estimation will get worse at a faster rate than IS. Since selecting a variational family that can match the typical set tends to be more difficult in higher dimensions, we should expect $\hat{k}$ to be larger for higher-dimensional posteriors. We can hope to reduce low overlap in typical sets either by improving on the variational family(for example, replacing mean field Gaussian density by mixture of Gaussian densities) or by reparameterising the model [Yao et al., 2018, Dhaka et al., 2021].

In table 2.3 we show a summary of the divergences we study in our work and the required number of finite moments for each of them. The faster $f(w)$ increases as w increase, the more finite moments that are required – which only get harder to estimate in high-dimensional cases.

2.15 Software

The convergence diagnostics for MCMC and $\hat{k}$ from importance sampling literature can be found in the `viabel` [Huggins et al., 2019], `arviz` [Kumar et al., 2019] and `PyMC3` [Salvatier et al., 2016] package. All these packages are in Python. Another notable package in R programming language with diagnostic is `posterior` Bürkner et al. [2020] .

2.16 Other inference schemes

2.16.1 MCMC and HMC

Hamiltonian Monte Carlo is a class of MCMC family of algorithms and widely considered as the gold standard. There have been many recent advances which have made HMC computationally more efficient and popular so much so that it is the main inference engine for modern probabilistic programming frameworks and languages such as Stan, PyMC, Pyro, and TensorFlow probability. The new dynamic U-turn HMC sampler aided with the automatic differentiation engine of these modern frameworks saves the user from the cumbersome and time-taking task of manually tuning algorithm's hyper-parameters, and manually computing gradients.

2.16.2 Laplace approximation

In Laplace approximation, the true posterior is approximated by a multivariate Gaussian with its mean as the mode of the true posterior and the inverse covariance matrix as the Hessian of the posterior at the mode.

$$p(\theta|Y,\phi) \approx q(\theta|\phi) := \mathcal{N}(\theta^*, \Sigma^*)$$

where θ^* is the mode and Σ^{-1} is the curvature of the posterior density.

2.16.3 Expectation propagation

Expectation propagation has been successfully deployed by researchers and has yielded excellent results with Gaussian processes. It has important and good theoretical qualities and can sometimes overcome some issues associated with standard VI, since it minimises the inclusive KL divergence between the target and the approximation, both of which are updated at each step.

Publication II also uses EP as one of the main inference techniques. Some recent work on deep Gaussian processes also uses EP [Bui et al., 2017]. EP is an iterative algorithm which minimizes the inclusive KL divergence (as opposed to the conventional exclusive KL divergence), has no theoretical convergence guarantees and in practice often suffers from numerical issues like underflow requiring bespoke implementations and numerical tricks. For lower dimensional and moderate data settings, it can often produce better results than VI as also shown empirically in II. Moreover, EP requires model specific derivations and can be difficult to implement when the moment matching sub-problem can not be solved in closed form.

3. Gaussian process and variational inference

This chapter introduces the concepts and background for Publications I and II and how these publications contribute towards research question 1 and 2. Section 3.1- 3.2 provide a brief introduction to Gaussian process and discuss the challenges in GP modelling. Section 3.3 provides an introduction to techniques developed in recent research aimed at solving these challenges, which the publications also make use of. Section 3.4 provides a short introduction of Bayesian optimization for a better understanding of Publication II. Finally, Section 3.5 concludes by listing software for these models.

3.1 Introduction

Gaussian processes are a type of stochastic process, suitable for defining flexible prior distributions for functions in a Bayesian setting. Gaussian process (GP) models have found applications in many domains such as signal processing [Wiener, 1949], geostatistics [Matheron, 1973], state-space modelling, reinforcement learning, Bayesian optimization among others. It has been successfully used in tasks such as drug-dose modelling for infants, predicting energy in atoms etc. Several methods used in other disciplines like Kalman Filter, Kriging, Wiener-Kolmogorov filtering are equivalent to GP. Gaussian processes became popular because they could be seen as a generalisation of neural network with infinite units as shown in the seminal work of Radford Neal [Neal, 1996]. They have been widely adopted in many research fields where uncertainty quantification is of paramount importance such as sequential decision making/active learning [Srinivas et al., 2010], model based planning [Deisenroth and Rasmussen, 2011], unsupervised data analysis and dimensionality reduction [Lawrence, 2004a], Bayesian optimization (BO) [Snoek et al., 2012, Gonzalez et al., 2016], applications of BO such as hyperparameter optimization of machine learning algorithms [Klein et al., 2017], approximate bayesian computation (ABC) for estimating horizontal gene transfer [Järvenpää et al., 2018, Lintusaari et al., 2018], human pose estimation [Ek, 2007], survival analysis [Saul et al., 2016], finding minimum energy paths in atoms [Koistinen et al., 2019], learning user preferences [Mikkola et al., 2020], [Siivola et al., 2020], time series [Frigola et al., 2014] and state-space modelling [Svensson et al., 2016].

Lawrence used GPs for non-linear dimensionality reduction, a model which came to be known as Gaussian process latent variable model (GPLVM) [Lawrence, 2004b]. GPLVMs generalised existing techniques for dimensionality reduction such as PCA (principal component analysis) a linear dimensionality reduction technique. At about the same time, Gaussian process were also used for modelling binary and multi-class classification [Kuss and Rasmussen, 2005, Nickisch and Rasmussen, 2008]. The inference procedures for GPs require inversion of a $N \times N$ matrix which has cubic complexity. More recently, researchers have been able to scale GPs to big data and learn richer representations building on recent advances in Sparse GPs and stochastic variational inference by Hoffman et al. [2013], allowing training with large and complex data.

This chapter reviews the basics of Gaussian process regression and classification, sparse GPs and application of variational inference and stochastic variational inference, forming the theory and background for Publication I–II.

A more thorough review of the Bayesian approach to Gaussian process regression can be found in the book of Rasmussen and Williams [2006].

The observation model stated above assumes that the observations are conditionally independent of each other given the latent function values f at the inputs X. This is the standard setting found commonly, but Publication II does not follow this, and the standard inference algorithms have to be modified to take this into account.

3.2 Gaussian processes

Formally defining, a Gaussian process (GP) is a collection of random variables with a multivariate Gaussian distribution for any finite set of these random variables. The random variables are often indexed in a continuous domain such as time or space. Gaussian random variables are closed under marginalisation and conditioning, which allows us to make a leap from the infinite dimensional object to a multivariate Gaussian distribution. A GP can also be seen as a generalisation of a multivariate Gaussian distribution to infinite number of dimensions. The marginalisation property of Gaussian distributions offers tractability and allows us to work only with a finite set of function instantiations at the training points $\mathbf{f} = [f(x_1), \cdots, f(x_n)]$ with which we can then make predictions for function instantiations at the test points($\mathbf{X}^*$): $\mathbf{f}^* = [f(x^*)]$. Bold letters indicate vectors.

A Gaussian process model for the probability distribution of function $f : \mathbb{R}^D \to \mathbb{R}$ is specified by a mean function $m : \mathbb{R}^D \to \mathbb{R}$ and a covariance function $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$. The mean function specifies the mean level of the distribution of $f(\mathbf{x})$ at a given input point $\mathbf{x} \in \mathbb{R}^D$, i.e., $\mathrm{E}[f(\mathbf{x})] = m(\mathbf{x})$, and the covariance function specifies how the values of f at any two input points, $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^D$, correlate with each other, more precisely, $\mathrm{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] = k(\mathbf{x}, \mathbf{x}')$. Given an arbitrary set of input points $\mathbf{X} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}]^\top$, the joint probability distribution of function values $\mathbf{f} = [f(\mathbf{x}^{(1)}), f(\mathbf{x}^{(2)}), \ldots, f(\mathbf{x}^{(N)})]^\top$ is defined as a multivariate Gaussian distribution

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{m}, K(\mathbf{X}, \mathbf{X})), \tag{3.1}$$

with mean vector

$$\mathbf{m} = [m(\mathbf{x}^{(1)}), m(\mathbf{x}^{(2)}), \ldots, m(\mathbf{x}^{(N)})]^\top,$$

and covariance matrix

$$K(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} k(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}) & k(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) & \cdots & k(\mathbf{x}^{(1)}, \mathbf{x}^{(N)}) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}^{(N)}, \mathbf{x}^{(1)}) & k(\mathbf{x}^{(N)}, \mathbf{x}^{(2)}) & \cdots & k(\mathbf{x}^{(N)}, \mathbf{x}^{(N)}) \end{bmatrix}.$$

The mean function of the prior GP model is assumed to be set to zero, which is a common practice and applied also in Publications I–II after suitable data transformation. To summarise, we can formulate GP as a hierarchical model here:

$$\begin{aligned} \text{Hyper prior} &:= \phi \sim p(\phi) \\ \text{Gaussian Process Prior} &:= \mathbf{f} \sim p(\mathbf{f}|\mathbf{X}, \phi) = \mathcal{N}(0, K(X, X)) \\ \text{Likelihood} &:= \mathbf{Y}|\mathbf{f}, \eta \sim p(\mathbf{Y}|\mathbf{f}, \beta) = \prod_{i=1}^{N} p(y_i|f_i, \beta). \end{aligned}$$

3.2.1 Covariance functions

The key part of a Gaussian process model is the covariance function, which is used to encode favourable properties of the unknown function. From the perspective of machine learning, it has a particularly important role in defining what can be learned about the function based on observed values. If a covariance function $k(\mathbf{x}, \mathbf{x}')$ depends only on the vector between the two points, $\mathbf{x} - \mathbf{x}'$, it is called stationary since it behaves similarly in all parts of the input space. If a covariance function is also isotropic, it can be written simply as a function of the distance $||\mathbf{x} - \mathbf{x}'|| = \sqrt{\sum_{d=1}^{D} (x_d - x'_d)^2}$, which means that the behaviour is similar in all directions.

For this thesis, we only consider the exponentiated quadratic (also called as radial basis function (r.b.f) or Gaussian kernel in some literature) kernel function as the only assumption we make is that the functions are smooth and continous. This is given as:

$$k(\mathbf{x},\mathbf{x}') = \sigma_{\mathrm{m}}^2 \exp\biggl(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2l^2}\biggr) \tag{3.2}$$

where the lengthscale l and the variance σ_{m} are the hyperparameters of the covariance function. The covariance is larger when the two input points are closer to each other and decreases with increasing distance. The magnitude σ_{m} defines the process variance, i.e., how much the values of f tend to deviate from the mean function, and the length scale l defines how far the effect of the covariance function fades out. In the isotropic form, the length scale is the same in all directions, but it is also possible to give separate length scales l_d for each input coordinate $d = 1, \ldots, D$: This covariance function is infinitely differentiable, meaning that sample functions/path drawn from the probability model are also infinitely differentiable.

3.2.2 Posterior as conditional Gaussian process

In Gaussian process regression, the observation model is given as: $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f},\sigma^2\mathbf{I})$. The posterior is also a GP, given as:

$$\mathbf{f}|X,y,\phi \sim \mathcal{N}(\mathbf{K}_{ff}(\mathbf{K}_{ff}+\sigma^2 I)^{-1}y, \mathbf{K}_{ff} - \mathbf{K}_{ff}(\mathbf{K}_{ff}+\sigma^2 I)^{-1}\mathbf{K}_{ff}).$$

The predictive distribution for the test datapoints $\mathbf{X}^*$ is given as:

$$p(\mathbf{f}^*|\mathbf{y},\mathbf{X},\mathbf{X}^*) = \mathcal{N}(\mathbf{f}^*|\mathbf{T}\mathbf{y}, K_{**} - \mathbf{T}\mathbf{K}_{\mathbf{f}*}),$$

where $\mathbf{T} = \mathbf{K}_{*f}(\mathbf{K}_{ff}+\sigma^2\mathbf{I})^{-1}$. The matrix K_{*f} denotes the cross-covariance matrix between the training data points X and test data points: X^*. The hyperparameters ϕ are optimised by maximising the marginal likelihood $p(\mathbf{y}|\phi) = \mathcal{N}(\mathbf{y}\ |0, \mathbf{K}_{ff}+\sigma^2\mathbf{I})$.

3.2.3 Challenges with Gaussian process modelling

There are two main challenges generally encountered when using Gaussian process as prior. When the likelihood is non-Gaussian, the posterior and predictive densities do not have a closed form. To overcome this, approximation methods discussed in section 2.16 are also applicable here. The second challenge is that for the general case, the computation operations require $O(N^3)$ operations and $O(N^2)$ memory complexity. Both the challenges are encountered in Publication I and II. As we see in those publications, both the problems can be overcome with the use of variational inference.

3.2.4 Other inference methods for GP

Other popular inference methods for GP include expectation propagation, Laplace approximation, and Hamiltonian Monte Carlo. Among these HMC is quite slow and cannot be used for complex large scale data applications.

Laplace approximation in context of GP models has been extensively used in packages like INLA (integrated nested Laplace integration)[Rue et al., 2009] and the `GPStuff` package by Vanhatalo et al. [2013]. For many models, this scheme offers fast inference without losing much accuracy. Laplace approximation can be used as a tool for marginalising parameters or/and hyper-parameters as its general idea and design can be applied in many situations.

3.3 Overcoming GP scalability and intractability

GP models are non-parametric, there is a latent function value f for each input point, which gives them the flexibility to model different forms of functions. There are several methods which have been proposed over time to tackle computational complexity challenges of GPs. Our focus in this thesis will be on sparse GPs with inducing points framework using stochastic variational inference (used in Publication I), which is also the most preferred technique with most impressive results. A related approach which uses inducing points to approximate the covariance matrix (K_{ff}) with a lower rank approximation $\mathbf{K}_{ff} \approx \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}$ where $K_{uu} \in \mathbb{R}^{M \times M}$ and $M \ll N$. A recent class of literature has scaled GPs with exact inference via matrix vector matrix multiplications (MVM) which has recently gained renewed interest among the community. These approaches initially used structure in data using a structured kernel matrix with data lying in a regularly spaced grid [Wilson and Nickisch, 2015]. More recently there has been significant success in applying MVMs to more general setting with the help of recent innovations such as distributed Cholesky factorization, faster estimation of log determinant [Ubaru et al., 2017], GPU acceleration and preconditioned conjugate gradients [Gardner et al., 2018, Wang et al., 2019].

Other inference methods like expectation propagation[Bui et al., 2017] and MCMC [Hensman et al., 2015a] have also been extended to be used with sparse GPs, variational inference has emerged as the most popular inference technique mainly because of its black box style optimization character and compatibility with gradient utilising auto-diff based frameworks like TensorFlow and PyTorch.

Other noteworthy approaches include mixture of expert models and distributed computations Deisenroth and Ng [2015], kernel expansions Yang et al. [2015] and basis function decomposition [Lázaro-Gredilla et al., 2010].

3.3.1 Opper's variational Gaussian process approximation

This section provides a summary of the variational method proposed by Opper and Archambeau [2009], which solves the intractability due to non Gaussian likelihoods and when used along with the variational inducing point framework has served as the basis for modern stochastic variational Gaussian process methods (SVGP) [Hensman et al., 2015b,a]. These methods have helped to scale GP models for large data classification. In GP context, research on VI has mostly sought algorithms which minimise the exclusive KL divergence $D_{\text{exclusive KL}} = \text{KL}(q(\mathbf{F})||p(\mathbf{F}|\mathbf{Y}))$ The exclusive KL penalises heavily the regions where the target probability density is low, which has the effect that the probability mass of the approximation concentrates around one of the modes of the target. The approximating distribution is a multivariate normal distribution $N(\mathbf{F}|\mathbf{m}, \mathbf{S})$ where $\mathbf{m}, \mathbf{S}$ are the variational parameters.

For the posterior : $p(\mathbf{F}|\mathbf{Y}) = \frac{p(\mathbf{Y},\mathbf{F})}{p(Y)} = \frac{p(\mathbf{Y}|\mathbf{F})p(\mathbf{F})}{p(\mathbf{Y})}$, The Exclusive KL divergence is given as:

$$\text{KL}(q(\mathbf{F})||\mathbf{F}|\mathbf{Y}) = \int q(\mathbf{F}) \log \frac{q(\mathbf{F})}{p(\mathbf{F}|\mathbf{Y})} d\mathbf{F} \quad = \log p(\mathbf{Y}) + \int q(\mathbf{F}) \log \frac{q(\mathbf{F})}{p(\mathbf{F},\mathbf{Y})} d\mathbf{F}$$

After some elementary manipulations, this can be written as

$$\log p(Y) = \int q(\mathbf{F}) \log p(\mathbf{Y}|\mathbf{F}) d\mathbf{F} - \text{KL}(q(\mathbf{F})||p(\mathbf{F})) + \text{KL}(q(\mathbf{F})||p(\mathbf{F}|\mathbf{Y})).$$

After some basic manipulations, we arrive as the inequality is the evidence lower bound (ELBO) for GP models and a new formulation to the one we saw in the previous chapter. This form of the bound is the one generally used in variational Gaussian process models, this is equivalent to the importance sampling flavoured ELBO formulation given in the previous chapter:

$$\log p(Y) \geq \int q(\mathbf{F})[\log p(\mathbf{Y},\mathbf{F}) - \log q(\mathbf{F})] d\mathbf{F}$$

The first term in the integral can be seen as encouraging variational parameters that give high density to configuration of latent variables which can explain the observations Y better . The second term encourages variational parameters which give rise to higher entropy distributions so that the distribution spreads its mass across many configurations [Ranganath et al., 2014] . The variational parameters are obtained by maximising the ELBO as formulated in Opper and Archambeau [2009]. For a factorised likelihood $p(\mathbf{Y}|\mathbf{F}) = \prod_{i=1}^{n} p(y_i|f_i)$, the lower bound denoted by $\mathcal{L}$ can be written as

$$\mathcal{L}(q) = \frac{1}{2}Tr(\mathbf{K}_{ff}\mathbf{S}) + \frac{1}{2}\mathbf{m}^\top\mathbf{K}_{ff}^{-1}\mathbf{m} - \frac{1}{2}\log|S| + \log Z - \frac{n}{2}\ln(2\pi e)$$
$$-\sum_{i=1}^{n}\int q(\mathbf{F})\log p(y_i|f_i)d\mathbf{F}$$

.

It is important to note that in the last term each of the term in the sum: $\int q(\mathbf{F})\log p(y_i|f_i)d\mathbf{F}$ depends only on the corresponding variational parameters: $\mathbf{m}_i$ and $\mathbf{S}_{ii}$. Applying another identity : $\mathbf{S}^{-1} = -2\nabla_{\mathbf{S}}\mathbb{E}_{q(\mathbf{F})}\Big[\log p(\mathbf{Y}|\mathbf{F})\Big]$ makes the optimal covariance matrix available to us as

$$\mathbf{S} = (\mathbf{K}_{ff}^{-1} + \beta\mathbf{I})^{-1},$$

where $\beta \in \mathbb{R}^{n\times 1}$ is a vector. This formulation uses only $2N$ variational parameters N for mean m and N for covariance matrix S, while the naive implementation requires $N(N+3)/2$ variational parameters. This is a major gain in terms of computation and space complexity. The variational parameters are optimised by gradient descent (not SGD!). The reduced number of variational parameters means higher chance of avoiding local minima. This is the formulation which is used in Publication II and while the likelihood is not factorised, we used both a full rank matrix and the above parameterisation.

3.3.2 Sparse GP

The cubic complexity of GPs limits their application to problems with $N \approx 10000$. The main computational bottleneck happens due to covariance matrix inversion. To overcome these scalablity challenges, *sparse approximations* have been proposed in the literature Titsias [2009], Quiñonero Candela and Rasmussen [2005] which reduce the computational cost to $O(NM^2)$ where $M \ll N$. The idea is to augment the probability space with M points of auxiliary(inducing) input/output pairs of variables Z(which lie in the same space as $\mathbf{X}$) and $\mathbf{U}$ (which lie in the same space as $\mathbf{F}$). The original covariance matrix K_{ff} with rank N is then replaced by a low rank approximation K_{uu} alleviating the computation cost of inverting the original matrix by inversion of a smaller matrix $O(M^3)$. the way K_{ff} is approximated is what distinguishes these methods from each other. The most popular and prominent methods are Deterministic training conditional (DTC) [Seeger et al., 2003], fully independent and training conditional (FITC) [Snelson and Ghahramani, 2005] and the variational sparse GP approach of Titsias [2009] which was further extended by Hensman et al. [2013, 2015b] and Damianou [2015] using further innovation of conjugate gradients.

Quiñonero Candela and Rasmussen [2005] and later Bui et al. [2017] made a comprehensive summary and unification of all the methods under sparse approximations.

The new problem is then to define a good approximation and find good inducing point locations Z. A common way to select the location is to optimise the inducing point location as done by Snelson and Ghahramani [2005]. Optimising so many new hyper-parameters can make us prone to overfitting. Another problem with these sparse approaches is that they modify the model, are non rigorous since the objective used in place of log marginal likelihood to optimise hyperparameters and inducing points/locations is an approximation having no theoretical guarantees, which can lead to severe overfitting. All these issues are resolved by the sparse variational Gaussian process models.

3.3.3 Sparse variational Gaussian process

This section summarises the variational sparse Gaussian process approach. SVGP models minimise the exclusive KL divergence between the approximation and joint posterior over $\mathbf{F}, \mathbf{U}$: KL$(q(\mathbf{F})||p(\mathbf{F}, \mathbf{U}|\mathbf{Y}))$. The key idea here is to replace the true posterior $p(\mathbf{F}|\mathbf{Y}, \mathbf{U})p(\mathbf{U}|\mathbf{Y})$ with the approximation having a specific factorisation: $q(\mathbf{F}) = p(\mathbf{F}|\mathbf{U})q(\mathbf{U})$. This factorisation helps to obtain the following variational bound

$$p(\mathbf{Y}) \geq \mathcal{L}(q) = \int q(\mathbf{U}) \Big[p(\mathbf{F}|\mathbf{U}) \log p(\mathbf{Y}|\mathbf{F}) d\mathbf{F} \Big] d\mathbf{U} - \text{KL}(q(\mathbf{U})||p(\mathbf{U})).$$

This is the new sparse variational ELBO, maximising this quantity wrt variational parameters brings the variational posterior close to the true joint posterior. Other hyper parameters such as length scale, noise variance are also optimised using this quantity. This is more principled way than the approaches discussed earlier since the risk of overfitting is significantly reduced as we are optimising a lower bound of the true log marginal likelihood as a result of which this is the default sparse variational Gaussian process approach for modern GP libraries.

If the likelihood factorises this can be further simplified to

$$\mathcal{L}(q) = \int q(\mathbf{U}) \Big[\sum_{i=1}^{N} \int p(f_i|\mathbf{U}) \log p(y_i|f_i) df_i \Big] d\mathbf{U} - \text{KL}(q(\mathbf{U})||p(\mathbf{U})).$$

To find the optimal variational distribution $q(\mathbf{U})$, Titsias differentiated the above bound with respect to $q(\mathbf{U})$. The optimal distribution also turns out to be a Gaussian

$$q(\mathbf{U}) = \mathcal{N}(\mathbf{U}|\mathbf{K}_{uu}(\mathbf{K}_{uu} + \frac{1}{\sigma^2}\mathbf{K}_{uf}\mathbf{K}_{fu})^{-1}\mathbf{K}_{fu}\mathbf{y}/\sigma^2, \mathbf{K}_{uu}(\mathbf{K}_{uu} + \frac{1}{\sigma^2}\mathbf{K}_{uf}\mathbf{K}_{fu})^{-1}\mathbf{K}_{uu}).$$

By placing this distribution in the above bound and performing the integration the variational lower bound becomes

$$\log p(\mathbf{y}|\mathbf{F}) \geq \log \mathcal{N}(\mathbf{Y}|0, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I) - \frac{1}{2\sigma^2} tr(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uf}).$$

The above bound has complexity $O(NM^2 + M^3)$, since $M \ll N$, the first term is the dominant one. As we discussed earlier, the other method in variational distribution is to fix the form of the variational approximation as a Gaussian and then optimise its parameters. This kind of model inspired by Hoffman et al. [2013] came to be known as SVI GP [Hensman et al., 2013] which was later extended for non-Gaussian likelihoods byHensman et al. [2015b].

3.4 Bayesian optimization

Bayesian optimization is a technique for finding the optimal input for an unknown function that is expensive to evaluate. It is closely tied to the popular *exploration vs exploitation dilemma*.

Expensive evaluation means that a clever strategy has to be made to know to make decisions based on the unknown function with a limited computation budget. This is done in a sequential manner using the information gained from past analysis to determine where to evaluate the function next. The unknown function, also known as the black box function is modelled using a Bayesian surrogate model, which in turn is then input to an 'acquisition function' which decides where the function has to be evaluated. The first use cases emerged due to a need to efficiently design experiments and systems. Now, BO has been applied to many use cases from a diverse set of fields like product design of material and drugs, finding new materials and molecules and in hyperparameter optimization of machine learning [Feurer et al., 2015] and deep learning algorithms [Snoek et al., 2012]. A more thorough and introductory tutorial is given by Shahriari et al. [2016].

Here I introduce the key concepts in BO so that we can explain in detail our contributions later. Let $\mathcal{X}$ be the input space and $f : \mathcal{X} \to \mathbb{R}$ be a continuous and expensive to evaluate *black-box* function. BO finds the minimum $\mathrm{x}_{\min}$ of this unknown function :

$$\mathrm{x}_{\min} = argmin_{\mathbf{x} \in \mathcal{Z}} f(\mathbf{x}),$$

where $\mathcal{Z} \in \mathcal{X}$ is the feasible optimization space. The key idea in BO is to transform this optimization problem into a sequence of decision problems of where to evaluate the function next . This is done with the help of an acquistion function: $a : \mathcal{X} \to \mathbb{R}$ that uses the posterior distribution of the latent function to score how useful an observation associated to input $x_{i}{}_{i=1}^{M}$is. BO can be seen as an iterative algorithm with the following steps:

1) Fit a GP model to the available observations 2) Observe the black-box function at the maximum point of the acquisition function, 3) Repeat steps 1 and 2 until the *stopping criterion* is not met.

A prespecified number of function evaluations also referred to as *computational budget* is a common stopping criterion.

3.4.1 Gaussian Process with comparative observation models

In literature, the most common likelihood function $p(Y|F)$ we observe is factorised where each observation y_i is independent of all other latent functions $f_{\ i}$ when conditioned on the latent function value f_i. One such case where this is not followed is preferential Bayesian optimization, where the observation model takes into account the ordering of two noisy latent function values at two input locations. This can be formulated as :$D^i = 1_{y^{i,1} > y^{i,2}}$. The likelihood when assuming Gaussian corruption in y is given as,

$$\begin{aligned} p(D^i|\mathbf{f}, \sigma^2) &= \int\int 1_{y^{i,1}>y^{i,2}} \mathcal{N}(y^{i,1}|f(x^{i,1}), \sigma^2)\mathcal{N}(y^{i,2}|f(x^{i,2}), \sigma^2) dy^{i,1} dy^{i,2} \\ &= \Phi\Big((2D^i - 1)\frac{f(x^{i,1}) - f(x^{i,2})}{\sqrt{2}\sigma}\Big), \end{aligned}$$

This likelihood is the central idea for Publication II. Publication I also uses a similar comparison, but it is between the function evaluations of two independent Gaussian process.

3.4.2 Preferential batch Bayesian optimization

In some cases, the requirement might be to select multiple evaluation inputs with the acquisition function. This situation arises if each observation is related to multiple inputs or if multiple observations are made for a single query.

If we have a batch of input locations: $\mathbf{x}_{i=1}$ and a set of observations $\mathbf{D} \in N^{m\times 2}$, such that $y_{\mathbf{D}_{i,1}} \leq y_{\mathbf{D}_{i,2}}, i \in [1, \cdots, m]$, the likelihood of observing $\mathbf{D}$ is

$$p(\mathbf{D}|\mathbf{f}) = \int \cdots \int \Big(\prod_{i=1}^{m} \mathbf{1}_{y_{\mathbf{D}_{i,1}} \leq y_{\mathbf{D}_{i,2}}}\Big)\Big(\prod_{k=1}^{B} \mathcal{N}(y_k|f_k, \sigma^2)\Big) dy_1 \cdots dy_B.$$

Publication II also proposes three acquisition functions which can be applied in this setting a) Batch Expected improvement (q-EI), b) batch Thompson sampling which is equivalent to regular Thompson sampling performed B times sequentially where B is the size of the batch and c) sum of variances (SV), which is given by:

$$\sum_{i=1} \text{Var}(p(y_i \leq y_j \forall i \neq j)) = \sum_{i=1}^{B} \Big(\mathbb{E}_f[p(y_i \leq y_j \forall i, j)] - \mathbb{E}_f[p(y_i \leq y_j \forall i, j)] \Big). \tag{3.3}$$

3.5 Gaussian process libraries

There are many existing publically available Gaussian Process libraries differing in their objectives and focus on inference schemes. GPStuff [Vanhatalo et al., 2013], GPML(MATLAB) [Rasmussen and Nickisch, 2010], INLA(R) [Rue et al., 2009] were written in the first decade of this century. The Python package GPy [since 2012] with its object oriented design made GP models available to a larger audience. It is written in Python and had a substantial impact on the later packages. Publication II uses GPy to implement the main algorithm. GPFlow [de G. Matthews et al., 2017] and GPyTorch [Gardner et al., 2018] are the modern GP libraries using variational inference as the main approximation method and using auto differentiation to overcome challenges of non-conjugacy and speed. Publication I uses GPFlow to get baseline results. MXFusion [Dai et al., 2019] used in Publication I and Stan [Team, 2021] used in publication II, III, IV are general purpose auto-diff using probabilistic programming frameworks using VI and HMC as the main approximation methods respectively. While GPFlow and GPytorch contain useful computation tricks to make GP computations fast, it is easier and natural to use different priors and marginalisation of hyper-parameters in Stan.

4. Summary of Contributions

4.0.1 Publication 1

A number of researchers are working to increase the performance of Gaussian process models performance beyond neural networks for different tasks. The cubic complexity in inference for Gaussian processes is a major challenge in achieving this objective. Extreme classification is an emerging field in machine learning where the number of classes can be very high. A standard approach in multi-class classification with GP models is to have one latent function for each class, and having all the latent functions coupled by the softmax link function which produces probabilities over the classes. This makes the complexity as $O(KN^3)$, prohibiting application of Gaussian process to this task for number of classes, $K > 10$ and number of training points, $N > 1000$. This work proposes several likelihood approximations of the common link functions such as softmax, logit functions which were proposed in context of linear models [Ruiz et al., 2018] combining with the well known variational inducing point framework given by Hensman et al. [2015b], Titsias [2009] to obtain a tractable lower bound on the marginal likelihood that is a sum over both data points and classes. This essentially means that at each step in optimization, the objective and gradients can be computed approximately after drawing samples over data points and the negative classes (all the classes except the true class). This helps to scale Gaussian process models for extreme classification for datasets where the number of classes is as large as 355 and $N \approx 13000$ in the EUR-lex dataset, a dataset where legal documents are classified into multiple legal concepts or categories. This means a combined three order of magnitude improvement. The results show that the GP models give higher test set accuracy compared to linear models using the same likelihood approximations and other GP stochastic likelihood functions. The approximation introduces some extra latent variables in the model (one per datapoint), whose inference is intractable, which makes us propose amortized variational inference algorithms to solve for these parameters in addition to the variational parameters like inducing point values, their locations with the help of auto-differentiation framework.

4.0.2 Publication 2

When modelling human preferences, recommender systems, A/B testing, it is often observed that the human subjects do not give a score or have a hard time in assigning a numerical score to the choices he or she is looking into. It is easier for them to rank them or make a one-to-one (pairwise) comparisons. This type of feedback is known as *preferential feedback* which has not received as much attention as the *direct feedback* methods. This work presents a framework called preferential batch Bayesian optimization (PBBO) that finds the optimum of a latent function of interest, given any type of preferential feedback for a group of two or more choices.

The latent function is modelled by a GP. The posterior is intractable and is approximated with EP and MCMC. VI can also be applied if it is possible to query the batch winner i.e. the member with the smallest value in the batch, the likelihood reduces to that of one-vs-each [Titsias, 2016] and a direct treatment of variational inference, as given by Opper and Archambeau [2009] becomes possible. This work used two different parameterizations of the Gaussian approximation, one having a factorial form (diagonal covariance matrix) and the other having a full rank covariance matrix. The full rank parameterization was able to capture the uncertainty better and performed as good as the EP approach while being more stable and fast than the other two techniques. This paper also proposes three acquisition functions which can be applied in this setting.

We found out, that the value of batch-size does not significantly affect the optimization performance when measuring the performance as a function of total number of points associated to the observations. The framework is applied to the problem of finding the best ingredients for sushi and best candy based on ratings given by users.

4.0.3 Publication 3

This publications results and conclusions are useful for applications which require accurate posterior estimates. In this publication, we first discuss the challenges related to obtaining a variational approximation using stochastic optimization methods. We show that black box VI can fail even when the true posterior lies in the same family as the approximation due to sub-optimal stochastic optimization. The stochasticity comes from one of the two/both sources: a) mini-batching b) evaluation of objective and gradients using Monte Carlo draws. We give an algorithm which makes the stochastic optimization more robust and accurate, where the choice of divergence to be minimised is the exclusive KL divergence. The key observation made in this publication is to view the optimization trajectory using SGD as a Markov chain. Though the theory holds strictly for SGD algorithm, it should also work for adaptive gradient based optimisers like ADAGRAD, RMSPROP etc. as long as the iterates have a small historical dependence so that departure from Markov property is not significant. This make us available a lot of diagnostic tools which have been so rigorously tried and tested in the MCMC diagnostics literature since 1990s namely: $\hat{R}$ for diagnosing convergence, effective sample size for obtaining independent draws, autocorrelation and Monte Carlo standard error (MCSE). We then use iterate averaging after detecting convergence to obtain a robust estimate of the quantity of interest. We recommend running many optimizers in parallel, the same way as MCMC and then combining the results. 'when to stop reliably' is a crucial question for the success of stochastic optimization in BBVI. This is really important when we want highly accurate posterior estimates. We also observed that analysing the optimizer behaviour is informative of the shape of the posterior and can guide the user towards models which are computationally easier to approximate.

The better accuracy and robustness in posterior estimates should be reflected in the way that the obtained variational distribution should have its moments closer to the one obtained using HMC taken to be the gold standard. The results are also analysed by comparing the tail index of the estimate. An approximation which covers the whole posterior mass in the space will results in bounded density ratios leading to the tail index being very small (or even negative). We also use the expected log predictive density (ELPD) as a metric to evaluate the proposed algorithm. However using this metric alone is also not a guarantee that the estimate is better, since a poorer approximation of the true posterior can still give better predictive results due to model-misspecification.

4.0.4 Publication 4

In black box VI, the user is required to make many choices such as: divergence measure(variational objective), approximating family, and the choice of stochastic optimiser(and related hyperparameters). This work focuses on the first two parts.

We show in this paper it is possible to use the Pareto-$\hat{k}$ diagnostic from the importance sampling literature to evaluate if the estimation of different divergence objectives and their gradients with MC draws are even reliable or not. This diagnostic index then helps us to evaluate and analyse if the black box VI procedure with the particular divergence and family choice has failed or not. If it has, then we can change our choices or even reparameterize the model.

This is important since BBVI assumes that replacing the integrals with finite average over MC samples follows the central limit theorem. However, we show that in practice this is not the case since BBVI operates in pre-asymptotic regimes and the CLT never really kicks in for problems with high dimensional and difficult highly non-Gaussian posteriors. We focus on the density ratios, which can also be interpreted as importance weights, evaluated at MC draws. When there is a mismatch between the true posterior density and variational density, the distribution over density ratio is heavily skewed to the right. Because of the heavy right skew, most of the mass of $w(\theta)$ is below its mean.

We show that even increasing the sample size is not going to help since the sample size which is required for a reliable estimate is too large and practically unfeasible for contemporary computers. For example, when the requirement is 10^{12} samples, raising the sample size say from 100 to 10^6 is not going to help.

While this can already be encountered in low dimensions with posteriors which are hard to approximate with Gaussian, an example being the eight schools model with the funnel shape posterior [Neal, 2003, Papaspiliopoulos et al., 2007]. This problem becomes more common when the dimensionality of the posterior increases due to curse of dimensionality. It has been suggested to use a multivariate t distribution with a small degree of freedom as the proposal density in literature, to make the density ratios bounded. We show empirically this only works practically for $D < 10$, since otherwise the bound on the weights is so large, that it can be treated as infinite for all practical reasons. The work finally suggests to use exclusive KL divergence objective as its estimation is the most reliable in high dimensional models.

5. Discussion

5.1 Recent Research by Others

The publications I–II in this thesis shows application of SVI and VI for Gaussian Process models for extreme classification

Galy-Fajou et al. [2020] have also proposed an approach similar to the one in Publication I where they introduce a new likelihood function to tackle the problem of evaluating the denominator of softmax function. The intractable sum in the denominator is replaced by an integral comprising an additional latent variable per datapoint. Some further augmented variables are introduced in a way such that the model is conditionally conjugate and the updates of variational parameters are closed-form, resulting in a fast and stable algorithm. Compared to the likelihood augmentations introduced by Ruiz et al. [2018] used in Publication I, the number of variational parameters per datapoint are more but the model being conditionally conjugate allows coordinate ascent style optimization (CAVI), which is not the case for Augment and Reduce likelihood.

Publication III addresses the problem of detecting convergence in stochastic VI algorithms and gave several diagnostics for detecting optimization issues. In some relatively recent work, the algorithms SASA [Lang et al., 2019] and SASA+ Zhang et al. [2020] have been recently proposed for detecting convergence in training of neural networks, the optimization for which is highly non-convex. It will be interesting to compare these methods with the one given by us.

A natural follow-up will be to visually investigate the landscape of ELBO and looking at its geometry in a similar way, as has been done for neural network landscape recently [Garipov et al., 2018, Draxler et al., 2018]. Zhang and Blei [2021] have done some preliminary research into this topic, but it deserves more thorough investigation.

Some recent work by Geffner and Domke [2020a] has shown that minimization of α divergence using unbiased gradient becomes more challenging as dimensions increase due to extremely high variance. Geffner and Domke [2020b] show some empirical results of minimizing α divergence with biased gradient estimators and summarise their findings as

- In high dimensions, an impractically large amount of computation in the form of MC draws is needed to mitigate this bias and obtain solutions that actually minimize the α-divergence.
- Solutions returned by algorithms using biased gradient estimators appear to be strongly biased towards minimizers of the traditional "exclusive" KL-divergence, $\mathrm{KL}(q||p)$.

Our work in Publication IV supports these findings, and also shows how the severe underestimation of density ratio with Monte Carlo estimate biases the algorithm towards solution of exclusive KL, conjectured in the second point above.

5.2 Limitations and recommendations for future research

For Publication I, the work did not cover extreme multi-label classification, where a data point can have multiple classes as output. Latest state of the art work in deep l earning and non-probabilistic methods can scale to 500K labels. In order to scale Bayesian methods to that scale will require distributed storage and algorithms and learning representations

A limitation for work carried out in II is that the datasets used were low dimensional, it is then a natural next step to scale the framework to higher dimensions. Using some ideas from some recent work such as [Siivola et al., 2021] could be helpful. The work also only focussed on the batch winner case, extending to other types of feedbacks can also be useful and almost straightforward to extend in the current framework.

VI has many benefits over other inference schemes like MCMC and EP, which explains its increased popularity in recent years, as illustrated in I–II. VI has made possible to use complicated models like Bayesian neural networks (BNN) on large datasets because of its speed, scalability, and good theoretical properties. However, the current practices have many potential pitfalls, which need to be guarded against and kept into account by the user. We hope, Publications III–IV will be a step in this direction, as there is a lot of scope for improving these approaches and introducing more checks and diagnostics for confident use of these powerful techniques. In Publication III, we propose to look at the distribution of the iterates and estimate the tail-index as tool to diagnose convergence for stochastic optimisation in VI. However, this procedure can be computationally tedious to scale for very high dimensional problems, since $\hat{k}$ has to be estimated for each dimension. It will be interesting to benchmark the performance of other multivariate tail index estimators [Mohammadi et al., 2015] for this problem. Although the experiments in Publication IV, included a range of common statistical model types, the findings may not generalize to all types of posteriors or to other variational families like semi-implicit variational distribution.

5.3 Practical implications

While Publication I has suggested new ideas for applying GP models to extreme classification problems. Publication II proposes a new method for doing Bayesian optimization with preferential feedback. Publication III–IV are useful to BBVI users. Publication III proposes a new robust algorithm for carrying out BBVI which should lead to a solution closer to optimum than the current algorithm or at least warns about non-reliable stochastic optimization. The users can then either reparameterise the model (c.f. eight school model) or change the optimization schedule. As discussed earlier, probabilistic programming will massively benefit from improved and robust inference algorithms. This will increase the user's trust in the workflow, making her free to make any modelling choices.

Publication III shows how to extend the tail index diagnostic ($\hat{k}$) to assess if the MC approximations to different divergence objectives are reliable and help the end users to select a divergence metric in combination with an approximating family. The work done in publication III and IV has inspired further research recently [Welandawe et al., 2022] where the authors build on the indeas introduced and propose an even more robust and adaptive algorithm.

The algorithm for robust stochastic optimization should be used to make the whole Bayesian workflow robust Gelman et al. [2020].

References

Shun-ichi Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276, 02 1998. ISSN 0899-7667. doi: 10.1162/089976698300017746.

Matthew J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University College London, 2003.

Michael Betancourt. A conceptual introduction to hamiltonian monte carlo, 2018.

Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul A. Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20:28:1–28:6, 2019.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112 (518):859–877, 2017.

Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan. Streaming variational bayes. In *In NIPS*, 2013.

Thang D. Bui, Josiah Yan, and Richard E. Turner. A unifying framework for gaussian process pseudo-point approximations using power expectation propagation. volume 18, pages 3649–3720, 2017.

Paul C. Bürkner, Jonah Gabry, M. Kay, and Aki Vehtari. *posterior: Tools for Working with Posterior Distributions.*, 2020. R package version 1.5-0.

Edward Challis and David Barber. Gaussian kullback-leibler approximate inference. *Journal of Machine Learning Research*, 14(32):2239–2286, 2013.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

R. T. Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1), 1946.

Zhenwen Dai, Eric Meissner, and Neil D. Lawrence. Modular deep probabilistic programming. In *International Conference on Learning Representations*, 2019.

Andreas Damianou. Deep gaussian processes and variational propagation of uncertainty. *PhD Thesis, University of Sheffield*, 2015.

Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. Gpflow: A Gaussian process library using tensorflow. volume 18, pages 1–6, 2017.

L. De Haan and L. Peng. Comparison of tail index estimators. *Statistica Neerlandica*, 52(1):60–70, 1998.

Marc Deisenroth and Jun Wei Ng. Distributed gaussian processes. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1481–1490, Lille, France, 07–09 Jul 2015. PMLR.

Marc Peter Deisenroth and Carl Edward Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 465–472, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

A.P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

Akash Kumar Dhaka, Alejandro Catalina, Manushi Welandawe, Michael Riis Andersen, Jonathan Huggins, and Aki Vehtari. Challenges and opportunities in high-dimensional variational inference. 2021.

Adji Bousso Dieng, Dustin Tran, Rajesh Ranganath, John Paisley, and David Blei. Variational inference via \chi upper bound minimization. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2732–2741. Curran Associates, Inc., 2017.

Aymeric Dieuleveut, Alain Durmus, and F Bach. Bridging the Gap between Constant Step Size Stochastic Gradient Descent and Markov Chains. *The Annals of Statistics*, 48(3):1348–1382, 2020.

Joshua V. Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matthew D. Hoffman, and Rif A. Saurous. Tensorflow distributions. *CoRR*, abs/1711.10604, 2017.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

Justin Domke. Provable smoothness guarantees for black-box variational inference. 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 2587–2596. PMLR, 13–18 Jul 2020.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. 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 1309–1318. PMLR, 10–15 Jul 2018.

Gideon Dresdner, Saurav Shekhar, Fabian Pedregosa, Francesco Locatello, and Gunnar Rätsch. Boosting variational inference with locally adaptive step-sizes. 2021.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July 2011. ISSN 1532-4435.

Carl Henrik Ek. Gaussian process latent variable models for human pose estimation. In *In 4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI 2007), volume LNCS 4892*, pages 132–143. Springer-Verlag, 2007.

Ilenia Epifani, Steven N MacEachern, and Mario Peruggia. Case-deletion importance sampling estimators: Central limit theorems and related results. *Electronic Journal of Statistics*, 2:774–806, 2008.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Roger Frigola, Yutian Chen, and Carl Edward Rasmussen. Variational gaussian process state-space models. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Théo Galy-Fajou, Florian Wenzel, Christian Donner, and Manfred Opper. Multi-class gaussian process classification made conjugate: Efficient inference via data augmentation. 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 755–765. PMLR, 22–25 Jul 2020.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8789–8798. Curran Associates, Inc., 2018.

Tomas Geffner and Justin Domke. Empirical evaluation of biased methods for alpha divergence minimization. In *Symposium on Advances in Approximate Bayesian Inference, AABI 2020*, 2020a.

Tomas Geffner and Justin Domke. On the difficulty of unbiased alpha divergence minimization. In *ICML 21*, 2020b.

Izrail Moiseevitch Gelfand, Richard A Silverman, et al. *Calculus of variations*. Courier Corporation, 2000.

Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C. Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. Bayesian workflow, 2020.

Zoubin Ghahramani. Factorial learning and the em algorithm. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, NIPS'94, page 617–624, Cambridge, MA, USA, 1994. MIT Press.

Javier Gonzalez, Zhenwen Dai, Philipp Hennig, and Neil Lawrence. Batch bayesian optimization via local penalization. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 648–657, Cadiz, Spain, 09–11 May 2016. PMLR.

Noah Goodman, Vikash Mansinghka, Daniel Roy, Keith Bonawitz, and Joshua Tenenbaum. Church: a language for generative models. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, UAI'08. AUAI Press, 2008.

Noah D Goodman and Andreas Stuhlmüller. *The Design and Implementation of Probabilistic Programming Languages*. 2014. Accessed: 2021-7-15.

GPy. GPy: A gaussian process framework in python. http://github.com/SheffieldML/GPy, since 2012.

Fangjian Guo, Xiangyu Wang, Kai Fan, Tamara Broderick, and David B. Dunson. Boosting variational inference. 2017.

Mert Gurbuzbalaban, Umut Simsekli, and Lingjiong Zhu. The heavy-tail phenomenon in sgd. 2020.

J Hensman, A Matthews, and Z Ghahramani. Scalable Variational Gaussian Process Classification. In *AISTATS 15*, volume 38 of *PMLR*, pages 351–360, 2015a.

J Hensman, AG Matthews, M Filippone, and Z Ghahramani. MCMC for variationally sparse Gaussian processes. In *NeurIPS*, pages 1648–1656, 2015b.

James Hensman, Nicolò Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 282–290, Arlington, Virginia, USA, 2013. AUAI Press.

Jose Hernandez-Lobato, Yingzhen Li, Mark Rowland, Thang Bui, Daniel Hernandez-Lobato, and Richard Turner. Black-box alpha divergence minimization. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1511–1520. PMLR, 20–22 Jun 2016.

Bruce M. Hill. A Simple General Approach to Inference About the Tail of a Distribution. *The Annals of Statistics*, 3(5):1163 – 1174, 1975.

G. E. Hinton and Tijmen Tieleman. Lecture 6.5 – Rmsprop: Divide the gradient by a running average of its recent magnitude. In *Coursera: Neural networks for machine learning*, 2012.

Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo, 2011.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. volume 14, pages 1303–1347, 2013.

Antti Honkela, Tapani Raiko, Mikael Kuusela, Matti Tornio, and Juha Karhunen. Approximate riemannian conjugate gradient learning for fixed-form variational bayes. *Journal of Machine Learning Research*, 11(106):3235–3268, 2010.

Jonathan H Huggins, Mikolaj Kasprzak, Trevor Campbell, and T. Broderick. Validated Variational Inference via Practical Posterior Error Bounds. In *AISTATS*, October 2019.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Wilson. Averaging weights leads to wider optima and better generalization. *Uncertainty in Artificial Intelligence - Proceedings, UAI 2018*, 2018.

E. T. Jaynes. *Probability theory: The logic of science*. Cambridge University Press, 2003.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, November 1999. ISSN 0885-6125.

Marko Järvenpää, Michael Gutmann, Aki Vehtari, and Pekka Marttinen. Gaussian process modeling in approximate bayesian computation to estimate horizontal gene transfer in bacteria. *The Annals of Applied Statistics*, 12, 12 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets. 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 528–536, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

David A. Knowles and Thomas P. Minka. Non-conjugate variational message passing for multinomial and binary regression. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, page 1701–1709. Curran Associates Inc., 2011. ISBN 9781618395993.

Olli-Pekka Koistinen, Vilhjálmur Ásgeirsson, Aki Vehtari, and Hannes Jónsson. Nudged elastic band calculations accelerated with gaussian process regression based on inverse interatomic distances. *Journal of Chemical Theory and Computation*, 15(12):6738–6751, 2019.

Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, and David Blei. Automatic variational inference in stan. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 568–576. Curran Associates, Inc., 2015.

Ravin Kumar, Colin Carroll, Ari Hartikainen, and Osvaldo Martin. Arviz a unified library for exploratory analysis of bayesian models in python. *Journal of Open Source Software*, 4(33):1143, 2019. doi: 10.21105/joss.01143.

Tomasz Kuśmierczyk, Joseph Sakaya, and Arto Klami. Variational bayesian decision-making for continuous utilities. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Malte Kuss and Carl Edward Rasmussen. Assessing approximate inference for binary gaussian process classification. volume 6, pages 1679–1704, 2005.

Tomasz Kuśmierczyk, Joseph Sakaya, and Arto Klami. Correcting predictions for approximate bayesian inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4511–4518, Apr. 2020.

Hunter Lang, Lin Xiao, and Pengchuan Zhang. Using statistics to automate stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 32, pages 9540–9550, 2019.

Neil Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2004a.

Neil D. Lawrence. Gaussian process latent variable mod- els for visualisation of high dimensional data. In *Advances in neural information processing systems*, NIPS'04, page 329–336, 2004b.

Neil D. Lawrence and Joaquin Quiñonero Candela. Local distance preservation in the gp-lvm through back constraints. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 513–520, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832.

Miguel Lázaro-Gredilla, Joaquin Quiñonero Candela, Carl Edward Rasmussen, and Aníbal R. Figueiras-Vidal. Sparse spectrum gaussian process regression. *J. Mach. Learn. Res.*, 11:1865–1881, August 2010. ISSN 1532-4435.

Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 983–992. PMLR, 16–18 Apr 2019.

Chiayu Lin, Andrew Gelman, Phillip Price, and David Krantz. Analysis of local decisions using hierarchical modeling, applied to home radon measurement and remediation. *Statistical Science*, 14, 08 1999.

Jarno Lintusaari, Henri Vuollekoski, Antti Kangasrääsiö, Kusti Skytén, Marko Järvenpää, Pekka Marttinen, Michael U. Gutmann, Aki Vehtari, Jukka Corander, and Samuel Kaski. Elfi: Engine for likelihood-free inference. *J. Mach. Learn. Res.*, 19(1):643–649, January 2018. ISSN 1532-4435.

Francesco Locatello, Rajiv Khanna, Joydeep Ghosh, and Gunnar Rätsch. Boosting Variational Inference: an Optimization Perspective. In *International Conference on Artificial Intelligence and Statistics*, 2018.

Stephan Mandt, Matthew D. Hoffman, and David M. Blei. Stochastic gradient descent as approximate bayesian inference. volume 18, pages 4873–4907. JMLR.org, January 2017.

G. Matheron. The intrinsic random functions and their applications. *Advances in Applied Probability*, 5(3):439–468, 1973.

Sean Meyn, Richard L. Tweedie, and Peter W. Glynn. *Markov Chains and Stochastic Stability*. Cambridge Mathematical Library. Cambridge University Press, 2 edition, 2009.

Petrus Mikkola, Milica Todorović, Jari Järvi, Patrick Rinke, and Samuel Kaski. Projective preferential bayesian optimization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6884–6892, Virtual, 13–18 Jul 2020. PMLR.

Andrew C. Miller, Nicholas J. Foti, and Ryan P. Adams. Variational boosting: Iteratively refining posterior approximations. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2420–2429. PMLR, 2017.

T. Minka, J.M. Winn, J.P. Guiver, Y. Zaykov, D. Fabian, and J. Bronskill. /Infer.NET 0.3, 2018. Microsoft Research Cambridge. http://dotnet.github.io/infer.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo gradient estimation in machine learning. 2019.

Mohammad Mohammadi, Adel Mohammadpour, and Hiroaki Ogata. On estimating the tail index and the spectral measure of multivariate α -stable distributions. *Metrika*, 78:549–561, 07 2015.

Eric Moulines and Francis Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

Christian A Naesseth, Fredrik Lindsten, and D. M. Blei. Markovian Score Climbing: Variational Inference with $\mathrm{KL}(p||q)$. 33, 2020.

Radford Neal. Slice sampling. *The Annals of Statistics*, 31:705 – 767, 2003.

Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996. ISBN 0387947248.

Hannes Nickisch and Carl Edward Rasmussen. Approximations for Binary Gaussian Process Classification. volume 9, pages 2035–2078, October 2008.

Victor M H Ong, David J Nott, and Michael S Smith. Gaussian Variational Approximation With a Factor Covariance Structure. *Journal of Computational and Graphical Statistics*, 27(3):465–478, 2018.

Manfred Opper and Cédric Archambeau. The variational Gaussian approximation revisited. *Neural Computation.*, 21(3):786–792, March 2009. ISSN 0899-7667.

Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.

Topi Paananen, Juho Piironen, Paul-Christian Bürkner, and Aki Vehtari. Implicitly adaptive importance sampling. *Statistics and Computing*, 31(2), Feb 2021. ISSN 1573-1375.

Omiros Papaspiliopoulos, Gareth O. Roberts, and Martin Skold. A general framework for the parametrization of hierarchical models. *Statistical Science*, 22(1): 59–73, 2007.

GI Parisi. *Statistical Field Theory*. Addison-Wesley, 1988.

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

Dennis Prangle. Distilling importance sampling. *arXiv.org*, arXiv:1910.03632 [stat.CO], October 2019.

Joaquin Quiñonero Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. volume 6, page 1939–1959. JMLR, December 2005.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black Box Variational Inference. 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 814–822, Reykjavik, Iceland, 22–25 Apr 2014. PMLR.

Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (gpml) toolbox. *J. Mach. Learn. Res.*, 11:3011–3015, dec 2010. ISSN 1532-4435.

CE Rasmussen and CKI Williams. *Gaussian Processes for Machine Learning*. MIT Press, 1 2006. ISBN 0-262-18253-X.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538. PMLR, 2015.

H. Robbins and S. Monro. A stochastic approximation method. In *The Annals of Mathematical Statistics.*, 1951.

Donald B. Rubin. Estimation in parallel randomized experiments. *Journal of Educational Statistics*, 6(4):377–401, 1981. ISSN 03629791.

Havard Rue, Sara Martino, and Nicholas Chopin. Models using integrated nested Laplace approximations. *Journal of the Royal Statistical Society B*, 71:319–392, 2009.

FJR Ruiz, MK Titsias, AB Dieng, and DM Blei. Augment and reduce: Stochastic inference for large categorical distributions. In *ICML 18*, 2018.

David Ruppert. Efficient estimations from a slowly convergent robbins-monro process. *Technical report, Cornell University Operations Research and Industrial Engineering*, 1988.

John Salvatier, Thomas V. Wiecki, and Christopher Fonnesbeck. Probabilistic Programming in Python Using PyMC3. *PeerJ Computer Science*, 2:e55, April 2016. ISSN 2376-5992.

Masa-Aki Sato. Online model selection based on the variational bayes. *Neural Comput.*, 13(7):1649–1681, July 2001. ISSN 0899-7667.

Alan D. Saul, James Hensman, Aki Vehtari, and Neil D. Lawrence. Chained gaussian processes. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1431–1440, Cadiz, Spain, 09–11 May 2016. PMLR.

Matthias W. Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse gaussian process regression. In Christopher M. Bishop and Brendan J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, volume R4 of *Proceedings of Machine Learning Research*, pages 254–261. PMLR, 03–06 Jan 2003. Reissued by PMLR on 01 April 2021.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Eero Siivola, Akash Kumar Dhaka, Michael Riis Andersen, Javier Gonzalez, Pablo Garcia Moreno, and Aki Vehtari. Preferential batch bayesian optimization. 2020.

Eero Siivola, Andrei Paleyes, Javier González, and Aki Vehtari. Good practices for bayesian optimization of high dimensional structured spaces. *Applied AI Letters*, 2(2):e24, 2021.

Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5827–5837, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, NIPS'05, page 1257–1264, Cambridge, MA, USA, 2005. MIT Press.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. ICML'10, page 1015–1022, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

Andreas Svensson, Arno Solin, Simo Särkkä, and Thomas Schön. Computationally efficient bayesian learning of gaussian process state space models. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 213–221, Cadiz, Spain, 09–11 May 2016. PMLR.

Stan Development Team. *Stan Modeling Language Users Guide and Reference Manual,*, 2021.

D. J. Thouless, P. W. Anderson, and R. G. Palmer. Solution of 'solvable model of a spin glass'. *The Philosophical Magazine: A Journal of Theoretical Experimental and Applied Physics*, 35(3):593–601, 1977.

Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1971–1979. PMLR, 2014.

MK Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *AISTATS 12*, 2009.

MK Titsias. One-vs-each approximation to softmax for scalable estimation of probabilities. In *NIPS*, 2016.

Shashanka Ubaru, Jie Chen, and Yousef Saad. Fast estimation of tr(f(a)) via stochastic lanczos quadrature. volume 38, pages 1075–1099, 2017.

Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. Gpstuff: Bayesian modeling with gaussian processes. *Journal of Machine Learning Research*, 14:1175–1179, April 2013. ISSN 1532-4435.

Aki Vehtari, Tommi Mononen, Ville Tolvanen, Tuomas Sivula, and Ole Winther. Bayesian leave-one-out cross-validation approximations for Gaussian latent variable models. *JMLR*, 17(1):3581–3618, January 2016. ISSN 1532-4435.

Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC. *arXiv preprint arXiv:1903.08008*, 2019a.

Aki Vehtari, Daniel Simpson, Andrew Gelman, Yao Yuling, and Jonah Gabry. Pareto smoothed importance sampling. *arXiv preprint arXiv:1507.02646*, 2019b.

Dilin Wang, Hao Liu, and Qiang Liu. Variational inference with tail-adaptive f-divergence. In *Advances in Neural Information Processing Systems*, volume 31, pages 5737–5747, 2018.

Ke Wang, Geoff Pleiss, Jacob Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact gaussian processes on a million data points. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Manushi Welandawe, Michael Riis Andersen, Aki Vehtari, and Jonathan H. Huggins. Robust, automated, and accurate black-box variational inference, 2022.

N. Wiener. Extrapolation, interpolation, and smoothing of stationary time series. 1949.

Andrew Gordon Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1775–1784. JMLR, 2015.

John Winn and Christopher M. Bishop. Variational message passing. *Journal of Machine Learning Research*, 6, 2005.

Frank Wood, Jan Willem van de Meent, and Vikash Mansinghka. A new approach to probabilistic programming inference. In *Proceedings of the 17th International conference on Artificial Intelligence and Statistics*, pages 1024–1032, 2014.

Sho Yaida. Fluctuation-dissipation relations for stochastic gradient descent. In *International Conference on Learning Representations (ICLR)*, 2019.

Zichao Yang, Andrew Wilson, Alex Smola, and Le Song. A la Carte – Learning Fast Kernels. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 1098–1106, San Diego, California, USA, 09–12 May 2015. PMLR.

Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Yes, but did it work?: Evaluating variational inference. 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 5581–5590, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Edith Zhang and David Blei. Unveiling mode-connectivity of the elbo landscape. In *Workshop on Bayesian Deep Learning, NeurIPS 2021*, 2021.

Pengchuan Zhang, Hunter Lang, Qiang Liu, and Lin Xiao. Statistical adaptive stochastic gradient methods, 2020.

Publication I

Akash Kumar Dhaka, Michael Riis Andersen, Pablo Garcia Moreno, Aki Vehtari. Scalable Gaussian Process for Extreme Classification. *The International Workshop on Machine Learning for Signal Processing MLSP* , Espoo, Finland, pages 1–6, October 2020.

SCALABLE GAUSSIAN PROCESS FOR EXTREME CLASSIFICATION

Akash Kumar Dhaka[1], *Michael Riis Andersen*[2], *Pablo Garcia Moreno*[3], *Aki Vehtari*[1]

[1]Aalto University, Dept. of Computer Science, [2] DTU Compute, Technical University of Denmark
[3]Amazon.com

ABSTRACT

We address the limitations of Gaussian processes for multiclass classification in the setting where both the number of classes and the number of observations is very large. We propose a scalable approximate inference framework by combining the inducing points method with variational approximations of the likelihood that have been recently proposed in the literature. This leads to a tractable lower bound on the marginal likelihood that decomposes into a sum over both data points and class labels, and hence, is amenable to doubly stochastic optimization. To overcome memory issues when dealing with large datasets, we resort to amortized inference, which coupled with subsampling over classes reduces the computational and the memory footprint without a significant loss in performance. We demonstrate empirically that the proposed algorithm leads to superior performance in terms of test accuracy, and improved detection of tail labels.

***Index Terms*—** Gaussian process classification, variational inference, augmented model.

1. INTRODUCTION

Multiclass classification refers to the supervised learning problem where each instance is labelled with a value chosen from a discrete set with cardinality $K > 2$. The goal of multiclass classification is to learn a mapping from an input space to the set of labels based on a set of input-output pairs $(\mathbf{x}_n, y_n)$, where $\mathbf{x}_n \in \mathbb{R}^D$ and $y_n \in \{1, 2, ..., K\}$. Extreme classification (EC) [1, 2] deals with the complexity introduced when the number of classes K is extremely large so that evaluation of the likelihood becomes prohibitively expensive using standard inference techniques. For example, consider the softmax function which maps K function values to a probability vector,

$$p(y = c|\mathbf{f}) = \frac{\exp(f_c)}{\sum_{i=1}^{K} \exp(f_i)}, \tag{1}$$

where $\mathbf{f} = [f_1, \ldots, f_K]$ is a vector of scores for each class for a given observation. Evaluating Eq. (1) and its gradients scales linearly with K. For very large data sets, this motivates the search for sub-linear, efficient, and accurate approximations.

Besides the computational challenges, the statistical challenges include 1) the average number of observations per class, N/K is small, 2) sparse data for a subset of classes, and 3) class imbalance in general. Bayesian methods in the setting where K is large, have received less attention than standard multi-class classification. Recently, Bayesian inference algorithms for extreme classification have been proposed for linear models [3, 4, 5].

While linear models have been shown to scale to very big data sets, non-linear models such as Gaussian processes (GPs) [6] can provide better performance by modeling non-linearities and covariate interactions. In the context of multi-class classification, imposing GP priors on each score function, f_i for $i = 1, \ldots, K$, allows modelling complex and non-linear dependencies in a probabilistic framework. Naive computations for GPs scale cubically with number of data points N, and for K-class GP classification the computation scales as $\mathcal{O}(KN^3)$. This makes it computationally non-trivial to apply GPs to scenarios where K is large.

There has been extensive work on how to reduce the computational cost arising due to large N, including sparse GPs using the inducing points framework [7, 8]. This reduces the computational cost per GP to $\mathcal{O}(\mathcal{B}M^2 + M^3)$, where M is the number of inducing points and $\mathcal{B}$ is the mini-batch size.

We propose a scalable GP framework for extreme classification by combining sparse GPs with recently proposed variational approximations of the likelihood terms. In particular, we study two different approximations: the One-vs-Each (OVE) approximation [4] and the *augment and reduce* (AR) approximation [5]. This allows us to approximate the likelihood and gradient for each observation using a small subset of the $(K - 1)$ negative classes such that the resulting cost will be independent of K. While AR offers better empirical performance than OVE, it introduces a set of local variational parameters for each observation. Since the number of variational parameters scales with N, the memory footprint can be prohibitively large for large datasets. We resolve this issue using amortized inference, where a neural network (NN) learns a mapping from the input space to the variational parameters. The NN is learnt jointly with the hyperparameters of the GP. We show that this solution does not degrade the performance of the AR approximation, but it keeps the memory footprint

978-1-7281-6662-9/20/$31.00 ©2020 IEEE

constant with respect to N. In addition, the optimisation problem is simplified as we tie up local parameters. Overall, this variational approximation performs better than previous GP approaches in literature on 4 out of 5 datasets in terms of accuracy and coverage. Finally, we share insights into how these likelihoods are related to each other.

1.1. Related Work

Relevant work on multiclass classification include [9, 10]. [10] use expectation propagation (EP) and an OVE-style bound that uses the probit function instead of the logistic function. EP is a fixed point algorithm which is hard to scale when the number of outcomes is large (in contrast to SVI). It does not offer a bound on the marginal likelihood, and it can suffer from convergence issues. Earlier variational approximations using augmented variables [11, 12] lack scalability. Sampling the latent function as done in [13] is not scalable for large K.

2. BACKGROUND

2.1. Gaussian processes for classification

GPs provide a principled way of imposing prior distributions over function spaces. We consider the problem where we have D-dimensional input vectors $\mathbf{x}_n \in \mathbb{R}^D$ associated with target class labels $y_n \in \{1, \ldots, K\}$ for $n = 1, \ldots, N$. We model the latent score function for each class $f_i \sim \mathcal{GP}(0, k)$ using a GP prior with covariance function $k(\cdot, \cdot, \boldsymbol{\theta})$. Given the set of input vectors $\mathbf{x}_n$, the joint prior distribution on the latent variables is given as

$$p(\mathbf{F}) = \prod_{i=1}^{K} p(\mathbf{f}^i), \quad p(\mathbf{f}^i) = \mathcal{N}(\mathbf{f}^i | \mathbf{0}, \mathbf{K}_{ff}),$$

where $\mathbf{f}^i = [f_i(\mathbf{x}_1), \ldots, f_i(\mathbf{x}_N)]$ and $[\mathbf{K}_{ff}]_{nm} = k(\mathbf{x}_n, \mathbf{x}_m, \boldsymbol{\theta})$. We will use $\mathbf{f}_n = [f_1(\mathbf{x}_n), \ldots, f_K(\mathbf{x}_n)]$ to denote the values of the latent functions for the n'th data point $\mathbf{x}_n$. We apply a link function $g : \mathcal{I}^k \mapsto \mathbb{R}^k$ that maps the probabilities of a categorical distribution that live in a K dimensional simplex $\mathbf{p}_n \in \mathcal{I}^K$ to the $\mathbf{f}_n \in \mathbb{R}^K$. The generative process is then

$$y_n \sim \text{Cat}(\mathbf{p}_n), \quad \mathbf{p}_n = g^{-1}(\mathbf{f}_n).$$

2.2. Inducing points and Stochastic Variational Inference

The coupled training points can be made conditionally independent given a set of inducing points $\mathbf{z}$ living in the same space as $\mathbf{x}$ [7, 14]. We augment the model with inducing output variables for each class, $\mathbf{u}^i = [f^i(\mathbf{z}_1), \ldots, f^i(\mathbf{z}_M)]$, i.e. the latent functions evaluated at the inducing points $\mathbf{z}$. The joint model for $(\mathbf{y}, \mathbf{f}, \mathbf{u})$ is then

$$\mathbf{u}^i \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{uu}}), \tag{2}$$

$$\mathbf{f}^i | \mathbf{u}^i \sim \mathcal{N}(\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{u}^i, \mathbf{K}_{ff} - \mathbf{Q}_{ff}), \tag{3}$$

$$y_n \sim \text{Cat}(g^{-1}(\mathbf{f}_n)), \tag{4}$$

where $\mathbf{Q}_{ff} = \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}$. The matrices $[\mathbf{K}_{uu}]_{ij} = k(\mathbf{z}_i, \mathbf{z}_j; \boldsymbol{\theta})$ and $[\mathbf{K}_{fu}]_{ij} = k(\mathbf{x}_i, \mathbf{z}_j; \boldsymbol{\theta})$ are the covariance matrix between inducing points and the cross-covariance matrix between the training points and inducing points, respectively.

Based on this generative model, [14] proposed to approximate the posterior distribution $p(\mathbf{F}, \mathbf{U}|\mathbf{X}, \mathbf{y})$ by $\prod_i q(\mathbf{f}^i, \mathbf{u}^i) = \prod_i p(\mathbf{f}^i|\mathbf{u}^i)q(\mathbf{u}^i)$, where $q(\mathbf{u}^i)$ is a variational multivariate Gaussian distribution $q(\mathbf{u}^i) = \mathcal{N}(\mathbf{u}^i|\mathbf{m}^i, \mathbf{S}^i)$. The variational parameters $\boldsymbol{\lambda} = \left\{\boldsymbol{\lambda}^i\right\}_{i=1}^K$, where $\boldsymbol{\lambda}^i = \left\{\mathbf{m}^i, \mathbf{S}^i\right\}$, and the kernel parameters θ are estimated by maximizing the evidence lower bound (ELBO)

$$\begin{aligned} \text{ELBO}(\boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{\eta}) = &\sum_{i=1}^{K} -\text{KL}(q(\mathbf{u}^i)||p(\mathbf{u}^i)) + \\ &\sum_{n=1}^{N} \text{E}_{q(\mathbf{f_n}|\boldsymbol{\lambda})} \log p(y_n|\mathbf{f_n}). \end{aligned} \tag{5}$$

Since the approximate posterior distribution $\prod_i q(\mathbf{f}^i|\boldsymbol{\lambda}) = \prod_i \int p(\mathbf{f}^i|\mathbf{u}^i)q(\mathbf{u}^i)d\mathbf{u}^i$ is a multivariate Gaussian, the marginals $q(\boldsymbol{f_n}|\boldsymbol{\lambda})$ are analytically available

$$q(\boldsymbol{f_n}) = \prod_i \mathcal{N}(f_n^i | m_n^i, (\sigma_n^i)^2), \tag{6}$$

$$m_n^i = \mathbf{k}_{nu}\mathbf{K}_{uu}^{-1}\mathbf{m}^i, \tag{7}$$

$$(\sigma_n^i)^2 = k_{nn} + \mathbf{k}_{nu}\mathbf{K}_{uu}^{-1}(\mathbf{S}^i - \mathbf{K}_{uu})\mathbf{K}_{uu}^{-1}\mathbf{k}_{un}. \tag{8}$$

The key idea is that conditioned on the inducing points, the training points become decoupled and the bound can be maximized using stochastic optimization. The ELBO contains two terms: the first is the sum of KL divergences between the prior distribution and $q(\mathbf{u}^i)$ for each class, which can be computed analytically. The second term is the sum of expectations of log likelihoods with respect to the vector of latent score function values $\mathbf{f_n} = [f^1, \ldots, f^K]$ at datapoint $\mathbf{x}_n$.

3. APPROXIMATE OBSERVATION MODELS

The second term of Eq. (5) involves a set of intractable expectations. In the binary classification unidimensional expectations can be approximated using quadrature methods [13]. In the multiclass scenario the link function $g(\cdot)$ couples all the latent variables $\mathbf{f}_n$, and for large K the high-dimensional integrals are not feasible with quadrature methods.

In this work, we consider two different approximations of the likelihood, where the high-dimensional integrals are replaced with a product of $(K-1)$ uni-dimensional integrals, each constituting a function operating on the pairwise differences $f_n^{ci} = f_n^c - f_n^i$ between the latent function values belonging to the target class c and one of the remaining classes i. As a result, we get approximations of Eq. (5) that also decomposes as a sum over classes

$$\mathrm{ELBO}(\boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{\eta}) \approx \sum_{i=1}^{K} \Big[- \mathrm{KL}(q(\mathbf{u}^i)||p(\mathbf{u}^i)) + \sum_{n=1}^{N} \mathrm{E}_{q(f_n^{ci})} \log p(y_n|f_n^{ci}) \Big]. \tag{9}$$

Since $q(f_n^{ci})$ are univariate Gaussians, the expectations in Eq. (9) can be efficiently approximated by quadrature. In addition, this decomposition is amenable to stochastic optimization, making it possible to process only a random subset of the negative classes $\mathcal{S}_n \subseteq \{1, \ldots, K\}\backslash c$, where c is the target class as in Eq. (10). This enables sparse updates

$$\mathrm{ELBO}(\boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{\eta}) \approx \sum_{n=1}^{N} \frac{K-1}{|\mathcal{S}_n|} \sum_{i \in \mathcal{S}_n} \Big(- \frac{1}{N} \mathrm{KL} (q(\mathbf{u}^i)||p(\mathbf{u}^i)) + \mathrm{E}_{q(f_n^{ci})} \log p(y_n|f_n^{ci}) \Big) \tag{10}$$

with constant computational complexity $\mathcal{O}(1)$ wrt. K. We choose $|\mathcal{S}_n| \ll K$, so that at each optimisation step, we make fewer updates to parameters reducing number of operations and memory footprint.

Next we describe the two different approximations for the likelihood: the One-vs-Each (OVE) approximation, and the Augment and Reduce (AR) approximation.

3.1. One-vs-Each (OVE)

The OVE approximation is done by replacing the exact probability by a lower bound based on pairwise probabilities corresponding to the event $y_n = c$ conditioned on the event that y_n takes one of the two labels $y_n \in \{c, k\}$ [4]. The joint log-likelihood function for the OVE approximation for the n'th observation is given by (see [4] for more details)

$$\log P(y_n = c|\mathbf{f}_n) = \log \frac{1}{1 + \sum_{i \neq c} e^{f_i - f_c}} \geq \log \prod_{i \neq c} \frac{1}{1 + e^{f_i - f_c}} = \sum_{i \neq c} \log \sigma(f_n^{ci}),$$

where the inequality follows from the fact that $(1 + \sum_i p_i) \leq \prod_i (1 + p_i)$ for $0 \leq p_i \leq 1$. Combining this bound with simple random sampling of the negative classes and substituting it into Eq. (10) yields the following approximate lower bound

$$\mathcal{L}_{\text{ove-sgd}} = \sum_{n=1}^{N} \frac{K-1}{|\mathcal{S}_n|} \sum_{i \in \mathcal{S}_n} \Big[- \frac{1}{N} \mathrm{KL}(q(\mathbf{u}^i)||p(\mathbf{u}^i)) + \tag{11}$$

$$\mathrm{E}_{q(f_n^{ci})} \log \sigma(f_n^{ci}) \Big]. \tag{12}$$

The stochastic OVE bound is an an unbiased estimate of the full OVE bound, but it is biased with respect to the original objective in Eq. (1) [3].

3.2. Augment and Reduce (AR)

Ruiz et al. [5] introduced a family of variational bounds for categorical likelihoods under the name of *augment and reduce* (A&R). The likelihood $p(y_n = c|\mathbf{f}_n)$ is augmented with a set of auxiliary variables $\boldsymbol{\epsilon}_n = [\epsilon_n^1, \ldots, \epsilon_n^K]$ such that

$$p(y_n = c|\mathbf{f}_n) = \int_{-\infty}^{\infty} \phi(\epsilon_n^c) \prod_{i \neq c} \Phi(f_n^c - f_n^i + \epsilon_n^i) d\epsilon_n^i, \tag{13}$$

where $\phi(\cdot), \Phi(\cdot)$ are the PDF and CDF of the auxiliary variables, respectively. The integral is intractable in general, but can be approximated with the following variational bound with respect to a variational distribution $q(\epsilon_n)$

$$\log p(y_n|\mathbf{f}_n) \geq \mathrm{E}_{q(\epsilon_n)} \Big[\log \frac{p(\epsilon_n)}{q(\epsilon_n)} + \frac{(K-1)}{|\mathcal{S}_n|} \sum_{i \in \mathcal{S}_n} \log \Phi(\epsilon_n + f_n^c - f_n^i) \Big]. \tag{14}$$

Thus, having a tractable CDF is a requirement for this approximation. The choices of the distributions for $\phi(\epsilon_n)$ and $q(\epsilon_n)$ determine the form of the likelihood. In this paper, we explore the following two specific choices: the logit and the softmax bounds [5].

3.2.1. AR Logit Bound

Choosing $\phi(\epsilon_n)$ to be the standard logistic distribution leads to the so called AR-logit bound on Eq. (13)

$$\log p(y_n|\mathbf{f}_n) \geq \mathrm{E}_{q(\epsilon)} \Big[\log \frac{\sigma(\epsilon)\sigma(-\epsilon)}{q(\epsilon)} + \frac{(K-1)}{|\mathcal{S}_n|} \sum_{i \in \mathcal{S}_n} \log \sigma(\epsilon + f_n^c - f_n^i) \Big]. \tag{15}$$

While the second term in the bound is intractable, we can use the reparameterization trick to approximate the expectation. Substituting this bound into Eq. (10) yields a lower bound that decomposes over classes. We will refer to this lower bound as $\mathcal{L}_{\text{arlogit}}$. The essence of the AR bound is that the K GPs, which are independent a priori, become coupled by the auxiliary variable for each data point. Assuming a Dirac delta distribution for ϵ centered at zero, the AR-logit bound collapses to the OVE bound plus a constant in Eq. (11). This generalises the OVE bound.

3.2.2. AR Softmax Bound

The equivalent AR bound for the softmax can be derived by substituting a standard Gumbel distribution for $\phi(\epsilon_n)$ in Eq. (14). By also choosing a Gumbel for the variational distribution $q(\epsilon_n)$, the general form of the bound given in Eq. (13)

simplifies to Eq. (16), since the expectation has an analytical solution

$$\log p(y_n|\mathbf{f}_n) \geq 1 - \log(\alpha) - \frac{1}{\alpha}\Big(1 + \frac{(K-1)}{|\mathcal{S}_n|}\sum_{k\in\mathcal{S}_n}\exp(f_n^k - f_n^c)\Big). \qquad (16)$$

Optimizing the variational parameter $\alpha \in [1, \infty)$ will provide a tighter bound to the softmax likelihood Eq. (1) than the OVE and OVE-SGD bounds. Unlike in the previous bounds, the expectation of Eq. (16) with respect to the marginals $q(f_n^{ci})$ given in Eq. (6) can be computed in closed form

$$\mathrm{E}_{q(\boldsymbol{f_n})}\left[\log p(y_n|\mathbf{f}_n)\right] \geq 1 - \log(\alpha_n) - \frac{1}{\alpha_n}\Big(1 + \frac{(K-1)}{|\mathcal{S}_n|}\sum_{i\in\mathcal{S}_n}\exp(-m_n^{ci} + \frac{(\sigma_n^{ci})^2}{2})\Big), \qquad (17)$$

where $m_n^{ci} = \mathrm{E}_{q(f_n^{ci})}\left[f_n^{ci}\right]$ and $\sigma_n^{ci} = \mathrm{Var}_{q(f_n^{ci})}\left[f_n^{ci}\right]$. Thus, this method does not require one-dimensional quadratures like in the ARlogit and OVE bound described above, hence removing the bias introduced by them [15].

3.3. OPTIMIZATION AND AMORTIZED INFERENCE

We optimize all the bounds introduced in section 3 with respect to both the variational parameters $\boldsymbol{\lambda}$ and the kernel parameters $\boldsymbol{\theta}$ using the ADAM optimizer with mini-batching. The OVE aproximation Eq. (11) is parameter free, but both AR approximations (Eq. (15) and (16)) introduce additional parameters in the ELBO due to the presence of the local variational distributions. This increases the dimensionality of the optimization problem, increasing the chance that the optimizer will get trapped in a local minima or a saddle point. To solve this problem, [5] proposes a nested loop approach in which they update the local variational parameters of a batch in a local/inner loop, re-estimate the ELBO quantity for this batch and then update the kernel parameters and $q(U; \lambda)$ parameters. The approach still needs to store the $\mathcal{O}(N)$ variational parameters. We refer to this scheme as the Inner-Loop-method (IL).

In contrast, we propose an amortized scheme (AMO) that reduces the memory footprint by embedding the constraint that similar data points which lie close to each other in the input space are likely to have similar auxiliary variables, and by extension similar variational parameters. We model ϵ_n as

$$\epsilon_n \sim q(\epsilon_n|\eta_n), \quad \eta_n = u(\boldsymbol{x}_n; \tilde{\boldsymbol{\lambda}}),$$

where η is the augmented variable parameterised by μ, β in the ARLOGIT bound and α in the ARSOFT bound. The map u can be any non-linear map from the input space to the variational parameters. In this work, we use a neural network with two hidden layers. The strength of the similarity constraint is controlled by the complexity and size of the network. Since the parameters are tied through by sharing of network weights, the optimisation problem is simplified.

4. EXPERIMENTS AND RESULTS

We evaluate the different methods empirically based on several benchmark datasets. For all datasets, we standardize by subtracting mean and dividing by standard deviations. BibTeX [1], Mediamill, Delicious [16] are all multilabel datasets which means that each datapoint may have more than one label assigned to it. We pick the first label for each datapoint as done in [5, 4]. This lowers the final number of classes for the last three datasets as given in Table 1. The mean values of $q(\mathbf{U})$ for each class are initialized randomly from $\mathcal{N}(0.1, 0.5)$ and the covariance matrix was initialised as an identity matrix.

4.1. Performance Metrics

We quantify the performance of the proposed methods with the classification accuracy and the *coverage*, motivated by the extreme learning community [17]. When the distribution of class labels are severely imbalanced, the classification performance for the infrequent classes will not be clearly reflected in the accuracy metric. It is given as the percentage of classes in test-set which have a non-zero number of true positives,

$$\text{Coverage} = K^{\text{TP}}/K^* \qquad (18)$$

where K^* represents the number of classes in the test set and K^{TP} is the number of classes with at least one true positive.

4.2. Baselines methods

Since most extreme classification methods, such as DISMEC [17] and PPD-Sparse [18], are based on linear models, we include linear models for both the OVE and AR-soft likelihood as baselines. We also compare our methods against two multi-class GP methods from the literature: the Robust-Max (GP-RM) likelihood [19], which was introduced for making models more robust to outliers, and Villa/Hernandez-Lobato likelihood (GP-HL), which can be derived in two ways by either taking the limit of noise parameter to zero in GP-RM, or by replacing the sigmoid function with a Gaussian CDF in the $\mathcal{L}_{\text{ove}}$ approximation. The computations are carried out using the GPFlow implementation [10].

4.3. Results

Table 1 compares the performance of the baseline methods with the proposed methods. The proposed GP methods perform better than linear models for all datasets except for the Delicious and Mediamill dataset, where the performance is similar to linear model. The ARSOFT approximation performs better than the rest on the first three datasets.

The experiments show that the AR methods generally perform better than both the non-stochastic and stochastic OVE methods when the number of negative class samples is fixed. The difference is more pronounced when K is large and $|\mathcal{S}|$ is

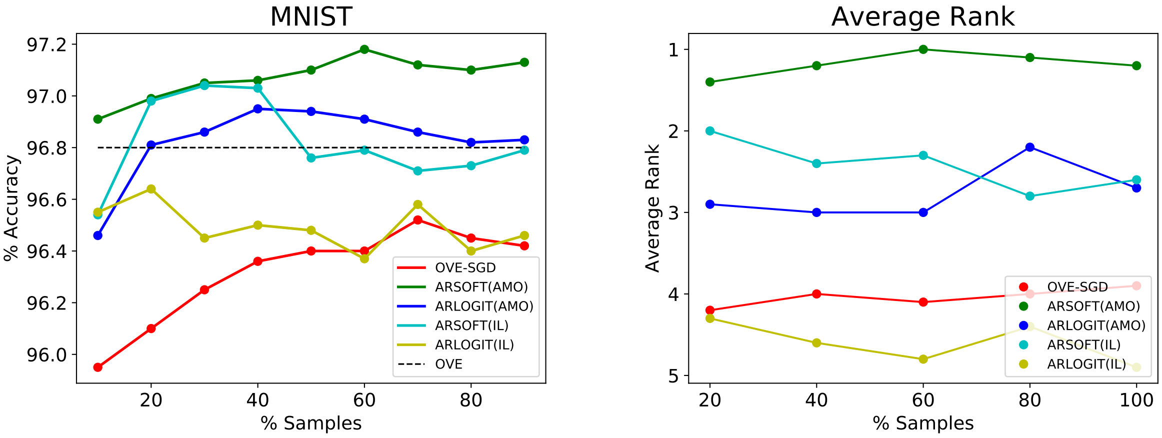


Fig. 1: The plot on the left shows the test set classification accuracy (higher is better) for the MNIST dataset as a function the sample size for the negative classes. The optimisation scheme is mentioned in parantheses. The plot on the right is a ranking plot from 1 to 5 (1 being best, 5 being lowest) for the different likelihood approximations and optimisation schemes for all datasets considered.

Name	N	K	Linear		GP-RM, GP-HL	OVE	$\|\mathcal{S}\|$	OVE-SGD	ARSOFT	ARLOGIT
			OVE	ARSOFT					AMO	AMO
MNIST	60000	10	91.9	92.4	95.4, 95.8	96.8	1	95.9	**96.9**	96.1
Fashion	60000	10	84.0	84.2	84.8, 86.3	87.8	2	86.5	**87.4**	86.6
BibTeX	4880	147	35.2	36.1	23.3, 34.2	35.9	30	36.4	**39.4**	36.8
Mediamill	30993	50	31.5	31.3	37.8, **38.9**	35.5	20	35.9	36.0	35.3
Delicious	12920	355	17.7	**18.3**	15.9, 17.5	16.4	30	16.0	16.4	16.2

Table 1: The third column gives accuracies obtained by a linear model combined with OVE and the best AR likelihood Ruiz2018. RM and HL refer to GP model with Robust max likelihood and Hernandez-Lobato likelihood, respectively. $|\mathcal{S}|$ is the subsample size. The baseline for GP was obtained using GPFlow, while for the linear models we used code provided by [5].

Name	Linear-OVE		Linear-ARSOFT		GP-RM		GP-HL		OVE-SGD		ARSOFT(AMO)		ARLOGIT(AMO)	
	A	C	A	C	A	C	A	C	A	C	A	C	A	C
M	31.5	7.0	31.3	7.2	37.8	12.5	**38.9**	20.9	35.9	35.0	36.0	**42.0**	35.3	22.0
M-10D	29.6	4.1	29.7	4.1	**34.8**	12.5	30.1	5.0	32.2	12.5	33.6	16.2	32.9	**16.5**
M-1000N	29.7	8.8	29.5	7.4	**32.3**	8.3	26.0	15.5	29.8	18.7	31.0	**24.0**	29.9	11.0
M-WMF	20.1	7.3	20.7	7.3	26.0	16.7	23.2	14.0	23.5	17.0	**26.4**	**35.5**	24.9	21.5

Table 2: Performance of models on Mediamill with different slices. M is the original Mediamill dataset, M-10D is reduced to $D = 10$ dimensions, M-1000N only contains $N = 1000$ observations, and in M-WMF the most frequent classes have been removed. A and C denote Accuracy and Coverage, respectively.

relatively small. This is consistent with the behavior observed by Ruiz et al. [5].

The performance of the amortized AR methods is better or similar to their non-amortized counterparts (see Figure 1), while having the advantage of a lower memory footprint. The Inner-Loop method (IL) does not perform as well for bigger data sets like BibTex.

The left panel in Figure 1 shows the classification accuracy for all methods on MNIST dataset when the percentage of negative class samples is varied from 10% to 90%. As expected, the general tendency is that classification accuracy increases when the percentage of negative samples is increased. The right panel shows the average rank for each method across all datasets. It is seen that the amortized AR method with the softmax likelihood is uniformly superior for all sample percentages. From here onwards, we only show results for amortized inference since they were mostly superior or similar to the inner loop inference, and more robust. An explanation could be that the optimisation in the local step can be challenging, quite sensitive to variational parameter update schedule and can get stuck in local minima, when the number of classes is high.

Table 1 shows that for the full Mediamill dataset, the proposed methods perform slightly worse than the baseline GP-RM and GP-HL methods. To further analyze this, we tested the methods on several different slices of the original Mediamill dataset. In particular, we manipulated the dimensionality D, the number of observations N, and the class imbalance by removing the most frequent classes. This resulted in the following three new datasets: M-10D, M-1000N, M-WMF, respectively, shown in Table 2. Both the baseline and proposed GP have better accuracy and coverage than the linear models for all variations of Mediamill. The accuracy for all baseline methods drop substantially when the most frequent classes are removed from the training set. The proposed methods seem to have disadvantage in case of high-class imbalance, but the relative performance gets better when the class imbalance is reduced. The two proposed methods have better coverage than the baseline methods for all variations of the Mediamill dataset. The ARSOFT method produced significantly better coverage in three out four variations of the Mediamill dataset, while producing comparable performance to the ARLOGIT method for the M-10D variant.

For all the data sets used in the experiments, a sample size of about 20-30% worked well and was sufficient for optimisation to be stable. The performance then saturated for higher sample sizes.

5. CONCLUSION

We proposed a scalable framework for extreme classification using Gaussian processes. The core idea is to combine the approximate likelihood method called Augment and Reduce with an amortized variational inference scheme. We applied the proposed methods to several benchmark datasets and demonstrated that the proposed method is capable of performing on par or even better compared to state-of-the-art methods for GP multi-class classification.

6. REFERENCES

[1] Y Prabhu and M Verma, "FastXML:Fast, accuraste and stable tree-classifier for extreme multi-label learning," in *KDD*, 2014.

[2] K Bhatia, H Jain, P Kar, M Varma, and P Jain, "Sparse local embeddings for extreme multi-label classification," in *NIPS*, 2015.

[3] F Fagan and G Iyengar, "Unbiased scalable softmax optimization," *arXiv preprint arXiv:1803.08577*, 2018.

[4] MK Titsias, "One-vs-each approximation to softmax for scalable estimation of probabilities," in *NIPS*, 2016.

[5] FJR Ruiz, MK Titsias, AB Dieng, and DM Blei, "Augment and reduce: Stochastic inference for large categorical distributions," in *ICML 18*, 2018.

[6] CE Rasmussen and CKI Williams, *Gaussian Processes for Machine Learning*, MIT Press, 1 2006.

[7] MK Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *AISTATS 12*, 2009.

[8] K Krauth, E V Bonilla, K Cutajar, and M Filippone, "AutoGP: Exploring the capabilities and limitations of Gaussian process models," in *UAI'17*, 2017.

[9] J Riihimäki, P Jylänki, and A Vehtari, "Nested expectation propagation for Gaussian process classification," *JMLR*, vol. 14, pp. 75–109, 2013.

[10] C Villacampa-Calvo and D Hernández-Lobato, "Scalable multi-class Gaussian process classification using expectation propagation," in *ICML'17*, 2017.

[11] M Girolami and S Rogers, "Variational Bayesian multinomial probit regression with Gaussian process priors," in *Neural Computation 18*, pp. 790–1817. 2006.

[12] JH Albert and S Chib, "Bayesian analysis of binary and polychotomous response data," *JASA*, vol. 88, pp. 669–679, 1993.

[13] J Hensman, A Matthews, and Z Ghahramani, "Scalable Variational Gaussian Process Classification," in *AISTATS 15*, 2015, vol. 38 of *PMLR*, pp. 351–360.

[14] J Hensman, N Fusi, and N Lawrence, "Gaussian processes for big data," in *UAI 2013*.

[15] AD Saul, *Gaussian Process Based Approaches for Survival Analysis*, Ph.D. thesis, 2018.

[16] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels,," in *ECML/PKDD 2008 Workshop on Mining Multidimensional Data*, 2008.

[17] R Babbar and B Schölkopf, "Dismec: Distributed sparse machines for extreme multi-label classification," in *WSDM'17*, 2017, pp. 721–729.

[18] IEH Yen, X Huang, W Dai, P Ravikumar, I Dhillon, and E Xing, "Ppdsparse: A parallel primal-dual sparse method for extreme classification," in *SIGKDD'17*, 2017.

[19] D Hernández-Lobato, J Miguel Hernández-Lobato, and P Dupont, "Robust multi-class Gaussian process classification," in *NeurIPS*, 2011, pp. 280–288.

Publication II

Eero Siivola, Akash Kumar Dhaka, Michael Riis Andersen, Pablo Garcia Moreno, Javier Gonzalez, Aki Vehtari. Preferential Batch Bayesian Optimization. *The International Workshop on Machine Learning for Signal Processing MLSP* , Gold Coast, Australia, November 2021.

PREFERENTIAL BATCH BAYESIAN OPTIMIZATION

Eero Siivola[⋆] *Akash Kumar Dhaka*[⋆] *Michael Riis Andersen*[†] *Javier González*[‡]
Pablo García Moreno[§] *Aki Vehtari*[⋆]

[⋆] Aalto University [†] Technical University of Denmark [‡] Microsoft Research [§] Amazon.com

ABSTRACT

Most research in Bayesian optimization (BO) has focused on *direct feedback* scenarios, where one has access to exact values of some expensive-to-evaluate objective. This direction has been mainly driven by the use of BO in machine learning hyper-parameter configuration problems. However, in domains such as modelling human preferences, A/B tests, or recommender systems, there is a need for methods that can replace direct feedback with *preferential feedback*, obtained via rankings or pairwise comparisons. In this work, we present preferential batch Bayesian optimization (PBBO), a new framework that allows finding the optimum of a latent function of interest, given any type of parallel preferential feedback for a group of two or more points. We do so by using a Gaussian process model with a likelihood specially designed to enable parallel and efficient data collection mechanisms, which are key in modern machine learning. We show how the acquisitions developed under this framework generalize and augment previous approaches in Bayesian optimization, expanding the use of these techniques to a wider range of domains. An extensive simulation study shows the benefits of this approach, both with simulated functions and four real data sets.

Index Terms— Gaussian processes, Bayesian optimization

1. INTRODUCTION

Understanding and emulating the way intelligent agents make decisions is at the core of what machine learning and artificial intelligent aim to achieve. To fulfil this goal, behavioural features can be learned from demonstrations like when a robot arm is trained using human-generated examples [1]. In many cases, however, the optimality of the instances is questionable. Reinforcement learning, via the explicit definition of some reward, is another approach [2]. That can be, however, subject to biases. Imagine asking a user of a streaming service to score a movie between zero and ten. Implicitly, this question assumes that she/he has a sense of the scale in which the new movie is evaluated, which implies that an exploration of the *movie space* has been already carried out.

An alternative way to understanding an agent's decisions is to do it via preferences. In the movie example, any two movies can be compared without scale. Also, the best of ten movies can be selected or a group of movies can be ranked from the worst to the best. This feedback, which can be provided without a sense of scale, provides information about the user preferences. Indeed, in prospect theory, studies have demonstrated that humans are better at evaluating differences rather than absolute magnitudes [3].

In the Bayesian optimization literature (BO), these ideas have also been studied in cases where the goal is to learn the optimum of some latent preference function defined in some Euclidean space [4]. Available methods use pairwise comparisons to recover a latent preference function, which in turn is used to make decisions about new queries. Despite the *batch* setting being a natural scenario here, where more than two points in the space are compared simultaneously, it has not yet been carefully studied in the literature. One relevant example for batch feedback is product design, especially in the food industry, where one can only produce a relatively small batch of different products at a time. The quality of products, especially for foods, is usually highly dependent on the time since production. The whole batch is usually best to be evaluated at once and the next batch of products should be designed based on the feedback so far. In this work, we show that mechanisms to propose preferential batches *sequentially* are very useful in practice, but far from trivial to define.

1.1. Problem formulation

Let $f : \mathcal{X} \to \mathbb{R}$ be a well-behaved *black-box* function defined on a bounded subset $\mathcal{X} \subseteq \mathbb{R}^d$. We are interested in solving the global optimization problem of finding

$$\mathbf{x}_{\text{min}} = \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}). \tag{1}$$

We assume that f is not directly accessible and that (noisy) queries to f can only be done in batches $\mathcal{B} = \{\mathbf{x}_i \in \mathcal{X}\}_{i=1}^{q}$. Let f be evaluated at all the batch locations, $y(\mathbf{x}_i) = f(\mathbf{x}_i)+\epsilon$, $i = 1,\ldots,q$, where $\epsilon \sim \mathrm{N}\big(0,\sigma^2\big)$ is a random noise with variance σ^2. We assume that we can receive a set of pairwise preferences of the noisy evaluations on the batch. Here a

Work done prior to Javier González joining Microsoft Research.

978-1-7281-6338-3/21/$31.00 ©2021 IEEE

pairwise preference is defined by $\mathbf{x}_i \prec \mathbf{x}_j := y(\mathbf{x}_i) \leq y(\mathbf{x}_j)$. The goal is to find $\mathbf{x}_{min}$ by limiting the total number of batch queries to f, which are assumed to be expensive. This setup is different from the one typically used in BO where direct feedback from (noisy) evaluations of f is available [5].

In particular, we are interested in cases in which the preferential feedback is collected in a sequence of B batches $\mathcal{B}_b = \{\mathbf{x}_i \in \mathcal{X}\}_{i=1}^q$ for $b = 1, \ldots, B$. Within each batch, at iteration b, the feedback is assumed to be collected as a complete (or partial) ordering of the elements of the batch, $I \in \mathbb{N}^q$, s.t. $\mathbf{x}_{I_i}^b \prec \mathbf{x}_{I_j}^b$, $\forall i < j \leq q$ or by the selection of the preferred element $\mathbf{x}_i^b$ of batch $\mathbf{x}_i^b \prec \mathbf{x}_j^b \; \forall j \neq i$.

We concentrate mainly on the batch winner case in this work. We argue that the batch winner feedback is the most useful type of batch feedback. As the preferential feedback is collected from humans, the full ranking of the batch is laborious for large batches and it sometimes even is impossible (e.g. in A/B testing). The full ranking can also be reduced to the batch winner case. Besides, as we later demonstrate in the experiments, the added benefit of full ranking instead of the batch winner is smaller than the difference between different acquisition functions.

1.2. Related work and contributions

Pairwise comparisons are usually called *duels* in the BO and bandits literature. In the BO context, [6] introduced a likelihood for including preferential feedback into Gaussian processes (GPs). [7] recomputed the model for all possible duel outputs of a discrete dataset and used the expected entropy loss to select the next query. [8] used expected improvement (EI) [9] sequentially to select the next duel. Most recently, [4] introduced a new state of the art and non-heuristic method inspired by Thompson sampling to select the next duel.

In this work, we introduce a method called *preferential batch Bayesian optimization* (PBBO) that allows optimizing black-box functions with BO when one can query preferences in a batch of input locations. The main contributions are:

- We formulate the problem so that the model for latent inputs in preference feedback scales beyond a batch size of two.
- We present and compare two alternative inference methods for the intractable posterior that results from the proposed batch setting.
- We adapt two well known acquisition functions to the proposed setting.
- We compare all inference methods, acquisition functions, and batch sizes jointly in extensive experiments with simulated and real data.

The code for reproducing the results is available at `https://github.com/EmuKit/emukit/tree/master/emukit/examples/preferential_batch_bayesian_optimization`

The remainder of the paper is organized as follows. In Section 2, we introduce the theoretical background. In Section 3, we introduce batch input preferential Bayesian optimization and three acquisition functions. In Section 4 we show the benefits of our approach with simulated and real data. We conclude the paper in Section 5.

2. MODELING BATCH PREFERENTIAL FEEDBACK WITH GAUSSIAN PROCESSES

We assume that the latent black-box function f is a realization of a zero-mean Gaussian process (GP), $p(f) = \mathcal{GP}$ fully specified by some covariance function K which specifies the covariance of the latent function between any two points [10].

2.1. Likelihood for batches of preferences

We propose a new likelihood function to capture the comparisons that are collected in batches. Assuming the general case of batch $\mathcal{B}$ of q locations and m preferences in a list $\mathbf{C} \in \mathbb{N}^{m\times 2}$, such that $\mathbf{x}_{C_{i,1}} \prec \mathbf{x}_{C_{i,2}} \forall i \in [1, \ldots, m]$, the likelihood of the preferences is

$$p(\mathbf{C}|\mathbf{f}) = \int \cdots \int \left(\prod_{i=1}^{m} \mathbb{1}_{y_{C_{i,1}} \leq y_{C_{i,2}}} \right) \left(\prod_{k=1}^{q} N(y_k|f_k, \sigma^2) \right) \mathrm{d}y_1 \ldots \mathrm{d}y_q, \tag{2}$$

where f_k is latent function value at $\mathbf{x}_k$ and σ is the noise of the comparison. This likelihood takes jointly into account the uncertainty of the preferences. As a special case, for batch size of two, the likelihood reduces to the one introduced in [6]. If the provided feedback is only the batch winner $\mathbf{x}_j$, the likelihood can be further simplified to (see details on the supplementary material)

$$p(\mathbf{C}|\mathbf{f}) = \int N(y_j|f_j, \sigma^2) \prod_{i=1, i\neq j}^{q} \Phi\left(\frac{y_j - f_i}{\sigma}\right) \mathrm{d}y_j, \tag{3}$$

where $\Phi(z) = \int_{-\infty}^{z} \mathrm{N}(\gamma|0,1)\,\mathrm{d}\gamma$. As the likelihood is not Gaussian, the posterior distribution is intractable and some posterior approximation has to be used.

2.2. Posterior distributions

The posterior distribution and the posterior predictive distributions of the model outcome are needed for making reasoned decisions based on the existing data. Let us assume B preference outcome observations $\mathbf{C}^b \in \mathbb{N}^{m\times 2}$ at batches $\mathbf{X}^b \in \mathbb{R}^{q\times d}$ $(b = 1, \ldots, B)$. Let us assume that the unknown latent function values $\mathbf{f}^b \in \mathbb{R}^{q\times 1}$ $(b = 1, \ldots, B)$ have a GP prior and each batch of preferences is conditionally independent given the latent values $\mathbf{f}^b$ at $\mathbf{X}^b$. The joint posterior distribution of all the latent function values $\{\mathbf{f}_b^p\}_{b=1}^B$ and $\mathbf{f}^*$ (at unseen $\mathbf{X}^*$) is

$$p(\mathbf{f}^*, \{\mathbf{f}^b\}_{b=1}^B | \mathbf{X}^*, \{\mathbf{X}^b\}_{b=1}^B, \{\mathbf{C}^b\}_{b=1}^B) \propto \\ p(\mathbf{f}^*, \{\mathbf{f}^b\}_{b=1}^B | \mathbf{X}^*, \{\mathbf{X}^b\}_{b=1}^B) \prod_{b=1}^B p(\mathbf{C}^b | \mathbf{f}^b). \tag{4}$$

The posterior predictive distribution for $\mathbf{f}^*$ is obtained by integrating over $\{\mathbf{f}^b\}_{b=1}^B$.

2.3. Model selection and inference

Since the likelihood of the preferential observations is not Gaussian, the whole posterior distribution is intractable and some approximation has to be used. Next, we present expectation propagation (EP) and variational inference (VI) approximations. EP can be used for general batch feedback in Eq. (2). With VI we limit to the batch winner case in Eq. (3). See more details on both these methods in the supplementary material.

2.3.1. Expectation propagation using multivariate normal as an approximate distribution

EP [11] approximates some intractable likelihood by a distribution from the exponential family so that the Kullback–Leibler (KL) divergence from the posterior marginals to the approximative posterior marginals is minimized. In this paper, we use multivariate normal distributions for each batch so that in the posterior distribution in Eq. (4), $\prod_{b=1}^B p(\mathbf{C}^b|\mathbf{f}^b)$ is approximated by $\prod_{b=1}^B N(\mathbf{f}^b|\boldsymbol{\mu}^b, \boldsymbol{\Sigma}^b)$. In practice, for approximative distributions from the exponential family, this can be done in an iterative manner where the approximation of batch b is replaced by the original one and the approximation of batch b is updated by matching the moments the full approximative distribution and the replaced one. Since the moments for the distribution in Eq. (2) are not analytically available, we approximate them by sampling.

2.3.2. Variational Inference using stochastic gradient descent

The batch winner likelihood (Eq. (3)) has the same structure as the one-vs-each likelihood in the context of multiclass classification with linear models [12]. In this formulation, the product of pairwise comparisons is used as a lower bound for the log of the batch winner likelihood,

$$\log \int N(y_j|f_j, \sigma^2) \prod_{i=1, i\neq j}^q \Phi\left(\frac{y_j - f_i}{\sigma}\right) \mathrm{d}y_j \\ \geq \sum_{i=1, i\neq j}^q \log \Phi\left(\frac{f_j - f_i}{\sqrt{\sigma_j + \sigma_i}}\right). \tag{5}$$

We want to highlight that this is not an exact likelihood, but a lower bound since we do not integrate over the uncertainty of the batch winner and we thus ignore the dependency of the observations within a batch.

Let $\mathbf{K}$ be the prior covariance matrix at $\mathbf{X} = [\mathbf{X}^1, \ldots, \mathbf{X}^B]^\mathrm{T}$, let $\boldsymbol{\alpha}$ be a vector, and let $\boldsymbol{\beta}$ be another vector. Following [13], we posit a Gaussian approximation of the posterior,

$$q(\mathbf{f}) = N(\mathbf{f}|\mathbf{K}\boldsymbol{\alpha}, (\mathbf{K} + \mathbf{I}\boldsymbol{\beta})^{-1}). \tag{6}$$

The variational parameters are optimised in an inner loop with stochastic gradient descent after collecting derivatives and likelihood terms from the comparison. The benefit of this form compared to EP is that it gives us a single bound making the optimization easier.

3. SEQUENTIAL LEARNING FOR BATCH SETTINGS

In this section, we present two strategies for selecting the batch locations. Although this work mainly concentrates on the batch winner case, the presented acquisition functions are applicable for the general preferential feedback of Eq. (2).

3.1. Expected Improvement for preferential batches

Expected improvement is a well established exploitative acquisition function that computes the expected improvement over the minimum of the values observed so far, y_{min}. It also has an extension in the batch setting, batch EI (q-EI) [14]. In the context of preferential feedback, we do not observe the exact function values and do not know the minimum of the observed values. This adds one more source of uncertainty to the q-EI for batches of direct feedback (see details on the supplementary material). One way of avoiding the computational cost of having to integrate over the uncertainty of the minimum and not having to update the model posterior is to use the minimum of the mean of the latent posterior ($\mu_{min} = \min_i \mu(\mathbf{x}_i)$) of the training data as a proxy for y_{min}. In this case, the acquisition function equals the relatively fast q-EI [14]

$$\begin{aligned} \text{q-EI} &= \mathbb{E}_\mathbf{y}\left[\left(\max_{i\in[1,\ldots,q]} (\mu_{min} - y_i)\right)_+\right] \\ &= \sum_{i=1}^q \mathbb{E}_\mathbf{y}\left(\mu_{min} - y_i \,\middle|\, y_i \leq \mu_{min},\, y_i \leq y_j \,\forall j \neq i\right) \\ &\quad p(y_i \leq \mu_{min},\, y_i \leq y_j \,\forall j \neq i). \end{aligned} \tag{7}$$

3.2. Thompson sampling for batches

A purely exploratory approach does not exploit the information about the known good solutions. EI approaches are known to over-exploit and the proposed approach is very expensive to compute. Although Thompson sampling is heuristic, it is known to work well in practice and nicely balance between exploration and exploitation. We use the scalable batch BO approach of [15] to select the batch locations in our experiments.

Algorithm 1 Pseudo-code of the proposed PBBO method. The inputs are the batch size q, the *stopping criterion*, the *acquisition strategy* and the GP model.

```
while stopping criterion is False do
    Fit a GP to the available preferential observations $\{\mathbf{C}^i\}_{i=1}^N$ at $\{\mathbf{X}^i\}_{i=1}^N$.
    Find $q$ locations $\mathbf{X}^{N+1} = \{\mathbf{x}_i\}_{i=1}^q$ using the acquisition strategy.
    Query the preference $\mathbf{C}^{N+1}$ of $\mathbf{X}^{N+1}$.
    Augment $\{\mathbf{X}^i\}_{i=1}^N$ with $\mathbf{X}^{N+1}$ and $\{\mathbf{C}^i\}_{i=1}^N$ with $\mathbf{C}^{N+1}$.
end while
```

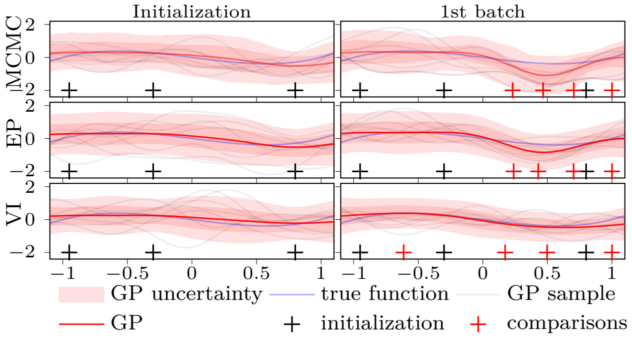


Fig. 1. Different rows visualize the true objective and the GP posterior for different inference methods (Markov chain Monte Carlo (MCMC), Expectation propagation (EP), and variational inference (VI)). The first column shows the GP posterior after observing preferential feedback for a batch of size three (x locations at black '+'-signs). The second column shows the first BO iteration (x locations at red '+'-signs) using q-EI and a batch size of four. The GP uncertainty is visualized as ±1 and ±2 standard deviations.

In practice, we sample q continuous draws from the posterior predictive distribution of the latent variable and select each batch location as a minimum of the corresponding sample.

3.3. Preferential batch Bayesian optimization

The pseudo-code for a general acquisition strategy is presented in Algorithm 1. The different parts of the algorithm scale as follows. Assuming N batches of size q, fitting the GP has the time complexity of $\mathcal{O}((Nq)^3)$. The inference method brings some overhead to this.

4. EXPERIMENTS

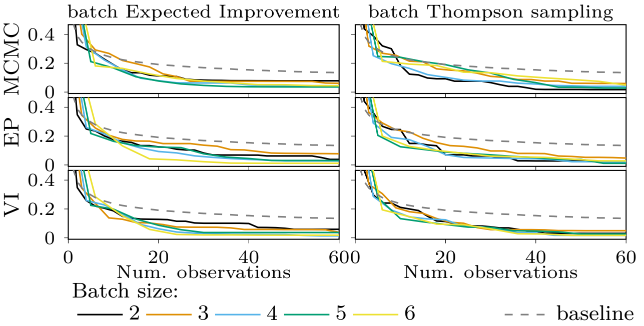


Fig. 2. Ursem Waves-function from the Sigopt library. Each line illustrates the smallest value of the objective function in the input locations visited so far as a function of the number of observations. The function is scaled between 0 and 1. Different colors in each plot are different batch sizes, rows are different inference methods and columns are different acquisition functions. Each line is a mean of 10 different runs. The dashed black line shows the average performance of the baseline, random search.

From hereafter, EP stands for the expectation propagation model and VI is the variational inference model. Markov chain Monte Carlo (MCMC) is used as a ground truth. As a baseline method, we show results if all acquisitions were selected completely at random. The acquisition strategies are abbreviated as follows. q-EI stands for the q-expected improvement, and TS stands for Thompson sampling. The methods are implemented using GPy [16] and the MCMC inference is implemented using Stan 2.18.0 [17]. In all experiments, the GP kernel is the squared exponential and the hyper-parameters are fixed to point values by optimizing a regular GP with 2500 noise free observations. The used MCMC algorithm is Hamiltonian Monte Carlo (HMC) with 9000 samples in total from 6 chains. The exact details and more results for all experiments are available in the supplementary material.

4.1. Effect of the inference method

The different inference methods approximate the uncertainty differently and this affects how the BO selects the next batch. Figure 1 visualizes the GP posterior for different inference methods with the same training data and then the posterior approximation after the first iteration of BO when using q-EI as the acquisition function. The black-box function is $f(x) = x^3 - x$. The Figure shows that EP results in very similar posteriors as the MCMC ground truth when observing only one batch of preferences that are relatively far away from each other (first column). When observing the second batch through BO, EP produces a wider posterior than MCMC, and VI produces a narrower posterior.

4.2. Synthetic functions from the Sigopt library

Sigopt library is a collection of benchmark functions developed to evaluate BO algorithms [18]. Ursem Waves is a function from the library with multiple local minima around the search

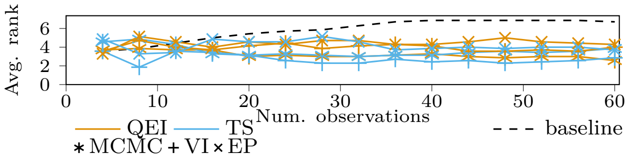


Fig. 3. All combinations of the inference methods and acquisition functions are ranked based on their average performance over 10 runs for the batch size of 4 as a function of the number of evaluations so far. Each combination is given a rank between 1–10 (lower is better) for each iteration. The figure shows the ranks averaged over 6 functions from the Sigopt library (Ursem Waves, Adjiman, Deceptive, MixtureOfGaussians02 and 3 and 4 dimensional Hartmann-functions). Performance of random search is shown as a baseline.

domain. Figure 2 shows the best absolute function value for the locations the function has been evaluated so far as a function of the number of function evaluations. The results are shown for batch sizes 2–6, three acquisition functions, and 3 inference methods. The shown lines are averaged over 10 random runs. The function evaluations are transformed to batch feedback by evaluating the function for the whole batch at once and returning the minimum as the batch winner. The results show that the both introduced acquisition functions perform better than the baseline. The results show no clear difference between batch sizes for any inference method or acquisition function. Figure 3 shows the average performance of the inference methods and all acquisition functions for the batch size of four. The results are averaged over six functions from the Sigopt library. The results show no clear difference in the performance between TS and q-EI.

4.3. Comparison of batch winner and full ranking

One argument against using the batch winner type feedback in BO is that it is less informative than full ranking. Although it is not always possible to get the full ranking and providing the batch winner is less work, it is interesting to compare these feedbacks. Figure 4 shows the average performance of complete ordering and batch winner type feedbacks for both acquisition functions with batch size four. The results are averaged over six functions from the Sigopt library. The results show that the variation between the acquisition functions is larger than it is between the feedback types.

4.4. Real life data case studies

To get insight into how the presented BO approach performs in real-life applications, we compare the methods also with real data. As the emphasis of this paper is on the batch size, we stick to low dimensional datasets ($d \leq 4$) and present results for batch sizes up to 6. We want to highlight that larger batch sizes and dimensions would remain feasible for TS. The

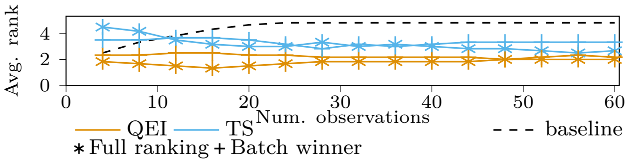


Fig. 4. Same as in Figure 3, but for comparing full ranking and batch winner feedbacks.

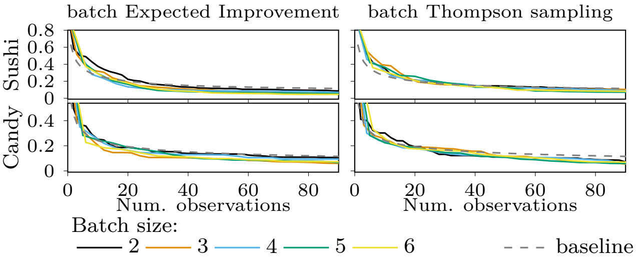


Fig. 5. Summary of the performance of the presented method on the real datasets. Each plot illustrates the minimum value seen so far as a function of the number of function evaluations. All data sets are scaled between 0 and 1. Different colors in plots are different batch sizes, rows are different data sets and columns are different acquisition functions. Each line is a mean of 10 different runs. The dashed black line shows the average performance of the baseline, random search. All lines use EP as an inference method.

presented datasets are as follows. Sushi dataset[1] has a complete ranking of 100 sushi items by humans and 4 continuous features describing each sushi item. In the Candy dataset, an online survey was used to collect pairwise preferences to 86 different candies[2] with two continuous features. We first recovered the full ranking (from best to worst) of the whole dataset and used that to provide feedback to any batch. Since real data is discrete, we use linear extrapolation to compute ranking for points that are not in the dataset.

Figure 5 shows the minimum function value seen so far as a function of the number of function evaluations. The results show batch sizes 2–6 and three acquisition functions for all 4 datasets when EP is used for inference. Each line shows the average over 10 random runs for each setting. The results are consistent with the results of the synthetic functions from the Sigopt library. One notable difference compared to the simulated results is that all methods beat the baseline only barely due to the low signal-to-noise ratio. When fitting a GP to the data sets that are scaled between 0–1, the noise standard deviation varies between 0.1–0.2. The continuous features alone seem not to be enough for performing the optimization.

[1] Available at: `http://kamishima.net/sushi/`

[2] Available at: `https://github.com/fivethirtyeight/data/tree/master/candy-power-ranking`

5. CONCLUSION

The results suggest that batch winner feedback is sufficient for optimization tasks as it is easier to collect than the full ranking of the batch and it performs almost as well in BO. The results suggest that if the dimensionality of the data is low and the chosen batch size is small, practitioners should use EP inference or MCMC sampling to approximate the posterior distribution. As an acquisition function, we recommend q-EI for low dimensions and small batch sizes and TS for high dimensions or large batch sizes due to its scalability.

Noise poses a problem to the presented approaches. Preferential observations are inherently less informative than direct observations. Noise reduces the information carried by preferential queries even more. Larger batch sizes do not reduce the problem due to the need to jointly account for the uncertainty of the whole batch. We see no apparent solution to this problem and it, unfortunately, reduces the usability of preferential feedback in noisy problems. With highly noisy data, random search with large batches might be a good option.

6. ACKNOWLEDGEMENTS

We acknowledge the computational resources provided by the Aalto Science-IT project and support by the Academy of Finland Flagship programme: Finnish Center for Artificial Intelligence, FCAI.

7. REFERENCES

[1] Jonathan Ho and Stefano Ermon, "Generative adversarial imitation learning," in *Proceedings of the 29th Conference on Neural Information Processing Systems*, pp. 4565–4573. 2016.

[2] Richard S. Sutton and Andrew G. Barto, *Introduction to Reinforcement Learning*, The MIT Press, 1st edition, 1998.

[3] Daniel Kahneman and Amos Tversky, "Prospect theory: An analysis of decision under risk," *Econometrica*, vol. 47, no. 2, pp. 263–91, 1979.

[4] Javier González, Zhenwen Dai, Andreas Damianou, and Neil D. Lawrence, "Preferential Bayesian optimization," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 1282–1291.

[5] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proceedings of the 25th Conference on Neural Information Processing Systems*, 2012, pp. 2951–2959.

[6] Wei Chu and Zoubin Ghahramani, "Preference learning with Gaussian processes," in *Proceedings of the 22nd International Conference on Machine Learning*, 2005, pp. 137–144.

[7] Wei Chu and Zoubin Ghahramani, "Extensions of Gaussian processes for ranking: semisupervised and active learning," *Proceedings of the Learning to Rank workshop at the 18th Conference on Neural Information Processing Systems*, pp. 29–34, 2005.

[8] Eric Brochu, *Interactive Bayesian optimization: learning user preferences for graphics and animation*, Ph.D. thesis, University of British Columbia, 2010.

[9] Jonas Močkus, "On Bayesian methods for seeking the extremum," in *Proceedings of the 28th Optimization Techniques IFIP Technical Conference*, 1975, pp. 400–404.

[10] Carl Edward Rasmussen and Christopher KI Williams, *Gaussian processes for machine learning*, The MIT Press, 2nd edition, 2006.

[11] Thomas P Minka, "Expectation propagation for approximate Bayesian inference," in *Proceedings of the 17th conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2001, pp. 362–369.

[12] Michalis K. Titsias, "One-vs-each approximation to softmax for scalable estimation of probabilities," in *Proceedings of the 29th Conference on Neural Information Processing Systems*, 2016, pp. 4161–4169.

[13] Manfred Opper and Cédric Archambeau, "The variational Gaussian approximation revisited," *Neural computation*, vol. 21, no. 3, pp. 786–792, 2009.

[14] Clément Chevalier and David Ginsbourger, "Fast computation of the multi-points expected improvement with applications in batch selection," in *Proceedings of the 26th International Conference on Learning and Intelligent Optimization*, 2013, pp. 59–69.

[15] José Miguel Hernández-Lobato, James Requeima, Edward O Pyzer-Knapp, and Alán Aspuru-Guzik, "Parallel and distributed thompson sampling for large-scale accelerated exploration of chemical space," *arXiv preprint arXiv:1706.01825*, 2017.

[16] GPy, "GPy: A Gaussian process framework in python," `http://github.com/SheffieldML/GPy`, since 2012.

[17] Stan Development Team, "The Stan Core Library," 2018, Version 2.18.0.

[18] Ian Dewancker, Michael McCourt, Scott Clark, Patrick Hayes, Alexandra Johnson, and George Ke, "A stratified analysis of Bayesian optimization methods," *arXiv preprint arXiv:1603.09441*, 2016.

SUPPLEMENTARY MATERIAL: PREFERENTIAL BATCH BAYESIAN OPTIMIZATION

Eero Siivola* *Akash Kumar Dhaka** *Michael Riis Andersen*† *Javier González*‡
Pablo García Moreno§ *Aki Vehtari**

* Aalto University † Technical University of Denmark ‡ Microsoft Research § Amazon.com

A. DERIVING THE LIKELIHOOD OF THE BATCH WINNER

If the provided batch type feedback is the batch winner $\mathbf{x}_j$,

$$\mathbf{C} = \begin{bmatrix} j & 1 \\ \vdots & \vdots \\ j & j-1 \\ j & j+1 \\ \vdots & \vdots \\ j & q \end{bmatrix}$$

the likelihood can be simplified as

$$\begin{aligned} p(\mathbf{C}|\mathbf{f}) = &\quad \int \ldots \int \left(\prod_{i=1,i\neq j}^{m} \mathbb{1}_{y_j \leq y_i}\right) \left(\prod_{k=1}^{q} N(y_k|f_k,\sigma^2)\right) \mathrm{d}y_1 \ldots \mathrm{d}y_q & (8) \\ = &\quad \int N(y_j|f_j,\sigma^2) \prod_{i=1,i\neq j}^{q} \left(\int_{y_j}^{\infty} N(y_i|f_i,\sigma^2)\mathrm{d}y_i\right) \mathrm{d}y_j & (9) \\ = &\quad \int N(y_j|f_j,\sigma^2) \prod_{i=1,i\neq j}^{q} \Phi\left(\frac{y_j - f_i}{\sigma}\right) \mathrm{d}y_j. & (10) \end{aligned}$$

B. DETAILS OF THE EXPECTATION PROPAGATION MODEL

In expectation propagation, the observation model is approximated by a product of distributions from the exponential family,

$$p(\{\mathbf{C}^b\}_{b=1}^B \,|\, \{\mathbf{f}^b\}_{b=1}^B) = \prod_{b=1}^{B} p(\mathbf{C}^b \,|\, \mathbf{f}^b) \approx \prod_{b=1}^{B} q_b(\mathbf{f}^b) = q(\{\mathbf{f}^b\}_{b=1}^B). \tag{11}$$

The approximative distributions, $q_b(\cdot)$, (parametrized by distributions from the exponential family) are called 'site approximations'. The prior distribution and the product of the site approximations can be used to approximate the posterior of the latent function values,

$$q(\mathbf{f}) = \frac{\mathrm{N}(\mathbf{f} \,|\, \mathbf{0}, k_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})) \left(\prod_{b=1}^{B} q_b(\mathbf{f}^b)\right)}{\int \cdots \int \mathrm{N}(\mathbf{f} \,|\, \mathbf{0}, k_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})) \left(\prod_{i=b}^{B} q_b(\mathbf{f}^b)\right) \mathbf{df}^1 \cdots \mathbf{df}^B}, \tag{12}$$

where $k_{\boldsymbol{\theta}}(\cdot,\cdot)$ is the covariance function parametrized by hyper-parameters θ, and $\mathbf{X} = [\mathbf{X}^1,\ldots,\mathbf{X}^B]^{\mathrm{T}}$. Before going into how to optimize the parameters of the site approximations, let us first define the 'cavity distribution',

$$q_{-j}(\mathbf{f}^j) \propto \int \cdots \int \mathrm{N}(\mathbf{f} \,|\, \mathbf{0}, k_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})) \left(\prod_{\substack{b=1 \\ b\neq j}}^{B} q_b(\mathbf{f}^b)\right) \mathbf{df}^1 \cdots \mathbf{df}^{j-1}\mathbf{df}^{j+1} \cdots \mathbf{df}^B. \tag{13}$$

Work done prior to Javier González joining Microsoft Research.

Also, let us define the 'tilted distribution', which is the cavity distribution multiplied with the exact likelihood of the missing observation

$$q_{\backslash j}(\{\mathbf{f}^j\}_{b=1}^B) = p(\mathbf{C}^j \,|\, \mathbf{f}^j) q_{-j}(\mathbf{f}^j). \tag{14}$$

In expectation propagation, we iteratively try to optimize the site approximations as a part of the global approximation. This is done by minimizing the following Kullback-Leibler (KL) -divergence

$$\mathrm{KL}\left(q_{\backslash j}(\mathbf{f}^j) \,||\, q_{-j}(\mathbf{f}^j q_j(f^j))\right) = \int q_{\backslash j}(\mathbf{f}^j) \log\left(\frac{q_{\backslash j}(\mathbf{f}^j)}{q_{-j}(\mathbf{f}^j) q_j(f^j)}\right) \mathrm{d}\mathbf{f}^j, \tag{15}$$

with respect to the parameters of the j:th approximative distribution. The solution to this equals to matching the zeroth, first and second moments of the global approximation and the tilted distribution[1]. Since in our case the moments of the tilted distribution are not computable in closed form, we approximate them by sampling.

C. DETAILS OF THE VARIATIONAL INFERENCE MODEL

The idea in variational inference is to minimize the Kullbak-Leibler -divergence between the approximative distribution and the intractable posterior,

$$\mathrm{KL}\left(q(\{f^b\}_{b=1}^B) \,||\, p(\{f^b\}_{b=1}^B \,|\, \{\mathbf{X}^b\}_{b=1}^B, \{\mathbf{C}^b\}_{b=1}^B)\right), \tag{16}$$

with respect to the parameters of the approximative distribution. In our case, the approximative distribution is defined either by Equation (7) or (8). However, since the posterior is intractable, the above equation must be approximated my maximizing the evidence lower bound,

$$\mathbb{E}_{q(\{f^b\}_{b=1}^B)}\left[\log p(\{\mathbf{C}^b\}_{b=1}^B \,|\, \{f^b\}_{b=1}^B)\right] - \mathrm{KL}\left(q(\{f^b\}_{b=1}^B) \,||\, p(\{f^b\}_{b=1}^B \,|\, \{\mathbf{X}^b\}_{b=1}^B)\right). \tag{17}$$

This requires some gradient based optimization scheme. The required derivatives can be found e.g. from Titsias (2016)[2].

D. EXPLAINING THE HIGH COMPUTATIONAL COST OF PQ-EI

As we further open Equation (8), it becomes:

$$\begin{aligned}\text{pq-EI} &= \mathbb{E}_{\mathbf{y},y_{min}}\left[\left(\max_{i\in[1,\dots,q]} (y_{min} - y_i)\right)_+\right] \\ &= \int_{-\infty}^{\infty} p(y_{min}) \sum_{i=1}^{q} \mathbb{E}_{\mathbf{y}}\left(y_{min} - y_i | y_i \le y_{min},\ y_i \le y_j\ \forall j \ne i,\ y_{min}\right) \times p(y_i \le y_{min},\ y_i \le y_j \forall j \ne i \mid y_{min}) \mathrm{d}y_{min},\end{aligned}$$

where $p(y_{min})$ is the distribution of minimum of $p \times q$ dimensional Normal distribution[3] (p is number of iterations before this and q is the batch size). More explicitly

$$p(y_{min}) = \sum_{i=1}^{p}\sum_{j=1}^{q} \mathrm{N}(-y_{min} \mid -\mu_{ij}, \Sigma_{ij\ ij}) \boldsymbol{\Phi}_{pq-1}\left(-y_{min}\mathbf{1} \mid -\boldsymbol{\mu}_{-ij}(y_{min}),\ \Sigma_{-ij-ij,ij}\right),$$

where $\mu_i j$ is the posterior mean of the latent function at i:th batch and j:th batch location, $\Sigma_{ij\ ij}$ is the posterior covariance of predictive output at the same location and

$$\boldsymbol{\mu}_{-ij}(y_{min}) = \boldsymbol{\mu}_{-ij} - (y_{min} - \mu_{ij})\boldsymbol{\Sigma}_{-ij\ ij}/\Sigma_{ij\ ij}, \text{ and } \Sigma_{-ij\ -ij,ij} = \Sigma_{-ij\ -ij} - \boldsymbol{\Sigma}_{-ij\ ij}\boldsymbol{\Sigma}^{\mathrm{T}}_{-ij\ ij}/\Sigma_{ij\ ij}.$$

Furthermore $\mathbb{E}_{\mathbf{y}}\left(y_{min} - y_i | y_i \le y_{min},\ y_i \le y_j\ \forall j \ne i,\ y_{min}\right)$ can be computed efficiently with Tallis formula for small batch sizes. However, even thought we need to only perform numerical integration over one dimension (as multidimensional cumulative normal distributions can efficiently be approximated), the computation of $p(y_{min})$ becomes computationally very demanding because of the need for computation of very high dimensional cumulative normal distribution functions ($pq - 1$ becomes very large after few iterations).

[1] Seeger, M. Expectation propagation for exponential families. Technical report, 2005.

[2] Titsias, M. K. One-vs-each approximation to softmax for scalable estimation of probabilities. In Proceedings of the 29th Conference on Neural Information Processing Systems, pp. 4161–4169, 2016.

[3] Arellano-Valle, R. B., and Genton, M. G. On the exact distribution of the maximum of absolutely continuous dependent random variables. Statistics & Probability Letters, 78(1), 27-35, 2008.

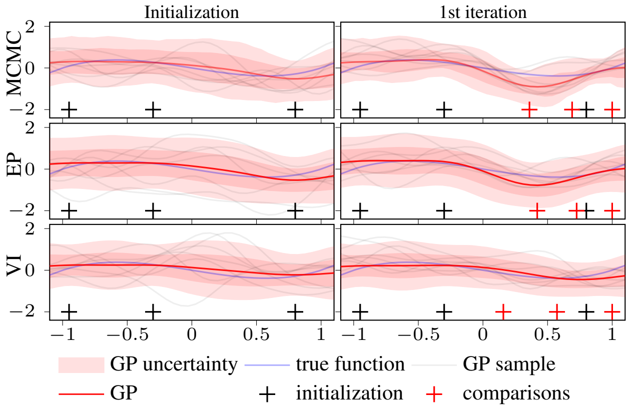


Fig. 6. Same as in Figure 1, but for batch size 3.

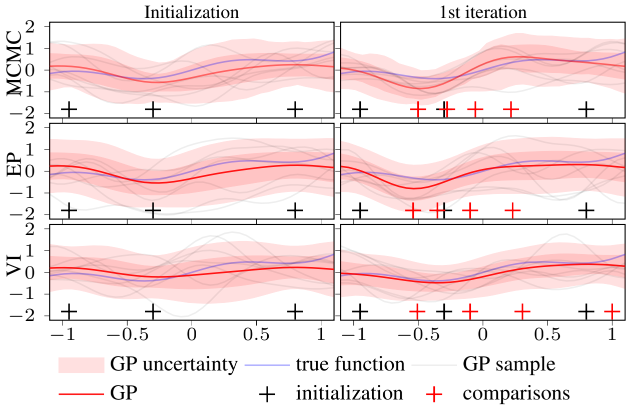


Fig. 7. Same as in Figure 1, but for different function.

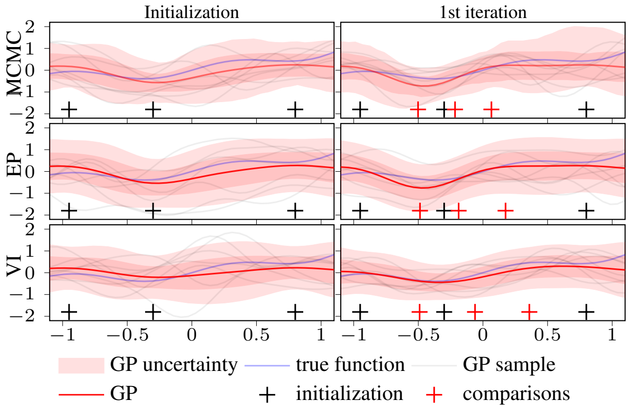


Fig. 8. Same as in Figure 1, but for batch size 3 and for different function.

E. DETAILS AND ADDITIONAL RESULTS FOR SECTION 4.1

Figure 6 presents same experiment as in Section 4.1, but for batch size of 6. Figures 7 and 8 have results for function $\frac{1}{4}\sin(5x) + \frac{1}{2}e^{x} - \frac{1}{2}$.

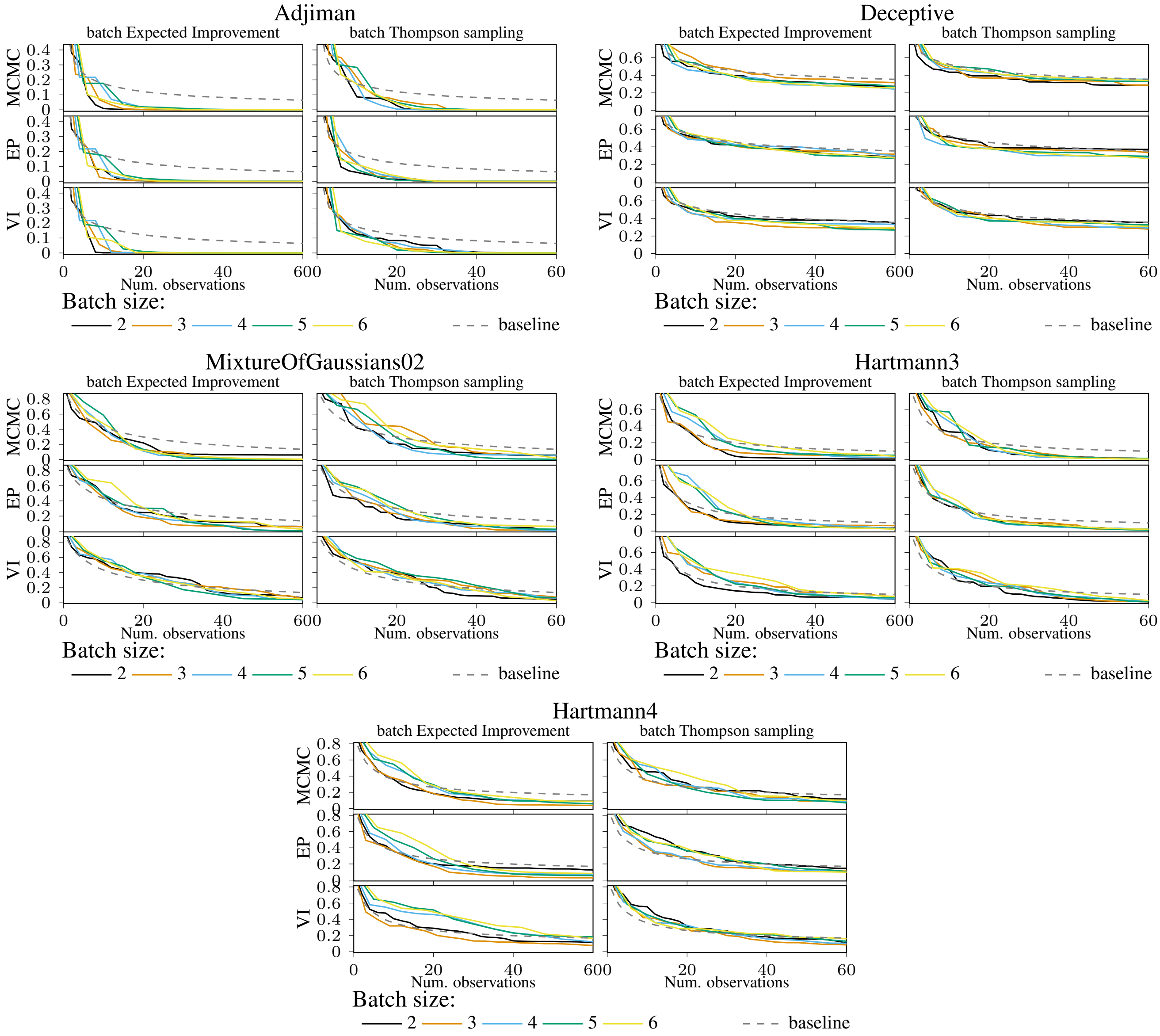


Fig. 9. Same as in Figure 2, but for Adjiman, Deceptive, MixtureOfGaussians02 and 3 and 4 dimensional Hartmann-functions from the Sigopt function library.

F. DETAILS AND ADDITIONAL RESULTS FOR SECTION 4.2

For all simulation runs, the function bounds and output was scaled between 0 and 1, for all dimensions. The batch feedbacks were computed from outputs which were corrupted with noise that has standard deviation of 0.05. q-EI was computed with 5000 posterior samples. All acquisition functions were optimized using limited memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm with box constraints. The optimization was restarted 30 times for q-EI. To increase the robustness of the EP, we do not allow distances between points to be less than 0.05 within a single batch. The numerical gradients of TS were computed with $\delta = 10^{-5}$ and only 100 closest samples were taken into account when conditioning on the evaluated samples. Optimization of EP was limited to at maximum 100 iterations. Optimization of VI was limited to 50 iterations and Adam was used for optimization.

Figure 9 presents similar results as in Section 4.2 for the 5 other functions from the Sigopt function library. All these functions are well known global optimization bench mark functions.

G. ADDITIONAL RESULTS FOR SECTION 4.3

Additional results for the experiments in Section 4.3. Figure 10 compares full ranking and batch winner feedbacks for six functions from the Sigopt library for q-EI and TS for batch sizes 3–6.

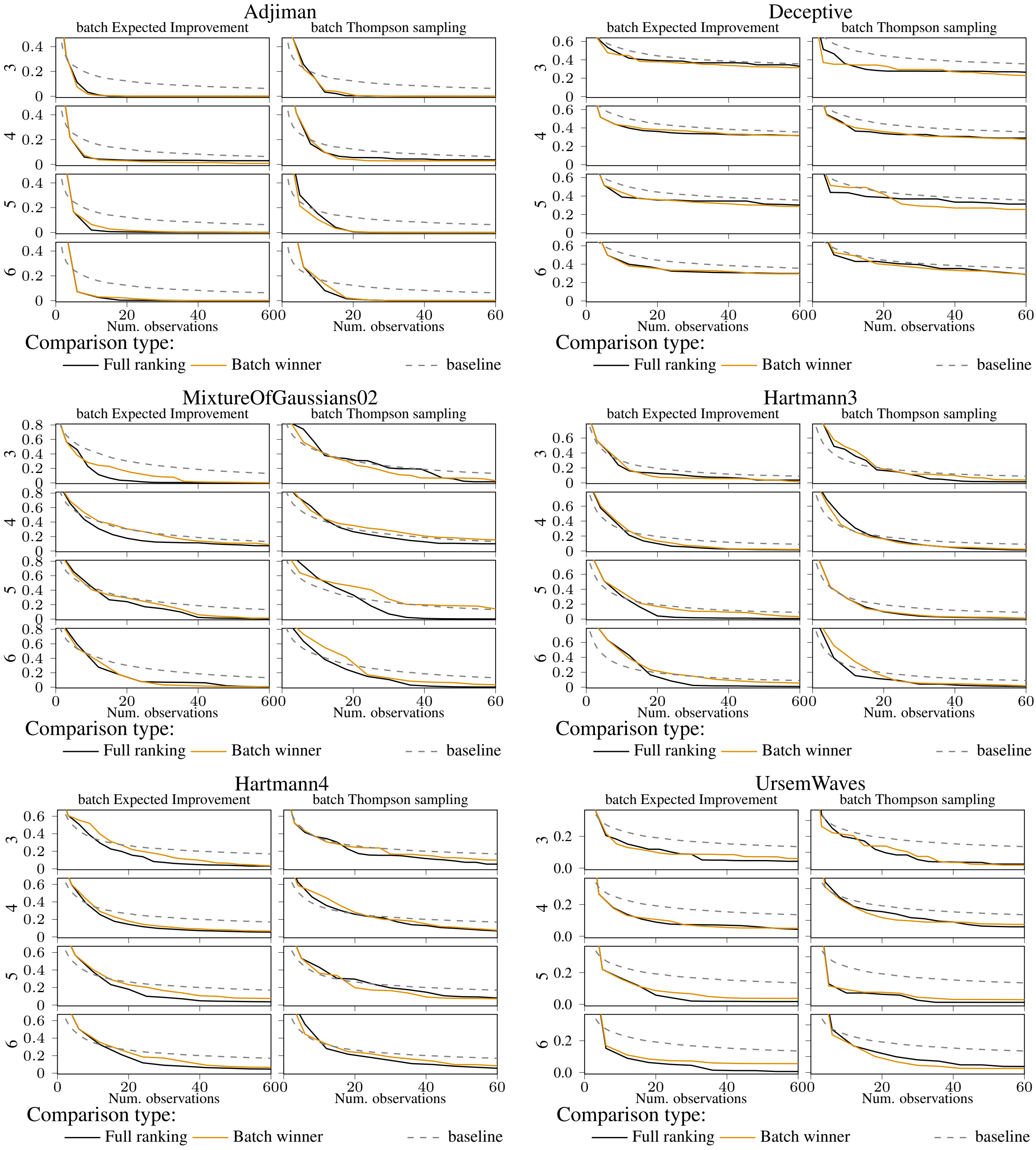


Fig. 10. Comparison of full ranking and batch winner feedbacks for six functions from the Sigopt function for the tree acquisition functions and batch sizes 3–6.

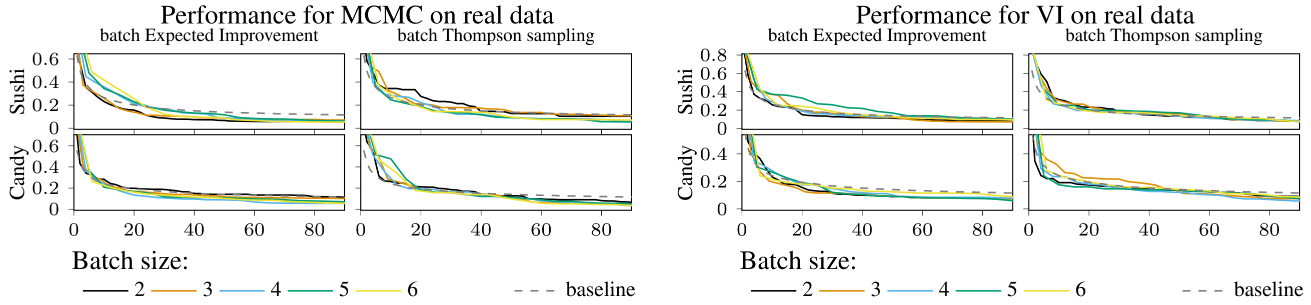


Fig. 11. Performances of MCMC and VI inferences on the ral data.

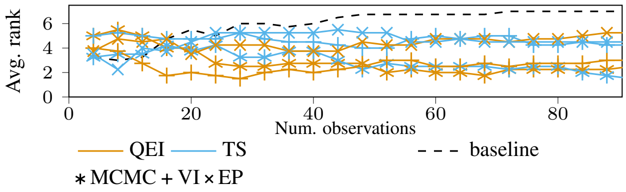


Fig. 12. Same as in Figure 3, but for Sushi and Candy datasets.

H. ADDITIONAL RESULTS FOR SECTION 4.4

The details of the experiments are the same as for the experiments in Section 4.4 with few exceptions. Since the functions are higher dimensional, acquisition optimization for q-EI was restarted 60 times. Also, no noise was added to real data.

Figure 11 presents similar results as in Section 4.4 for VI and MCMC. Figure 12 illustrates the ranks of different inference methods similarly as in Figure 3, but for Sushi and Candy datasets.

Publication III

Akash Kumar Dhaka, Alejandro Catalina, Michael Riis Andersen, Mans Magnusson, Jonathan Huggins and Aki Vehtari. Robust, Accurate Stochastic Optimization for Variational Inference. *Advances in Neural Information Processing Systems*, Volume 33, pages 10961–10973, 2020.

Robust, Accurate Stochastic Optimization for Variational Inference

Akash Kumar Dhaka
Aalto University
akash.dhaka@aalto.fi

Alejandro Catalina
Aalto University
alejandro.catalina@aalto.fi

Michael Riis Andersen
Technical University of Denmark
miri@dtu.dk

Måns Magnusson
Uppsala University
mans.magnusson@statistik.uu.se

Jonathan H. Huggins
Boston University
huggins@bu.edu

Aki Vehtari
Aalto University
aki.vehtari@aalto.fi

Abstract

We consider the problem of fitting variational posterior approximations using stochastic optimization methods. The performance of these approximations depends on (1) how well the variational family matches the true posterior distribution, (2) the choice of divergence, and (3) the optimization of the variational objective. We show that even in the best-case scenario when the exact posterior belongs to the assumed variational family, common stochastic optimization methods lead to poor variational approximations if the problem dimension is moderately large. We also demonstrate that these methods are not robust across diverse model types. Motivated by these findings, we develop a more robust and accurate stochastic optimization framework by viewing the underlying optimization algorithm as producing a Markov chain. Our approach is theoretically motivated and includes a diagnostic for convergence and a novel stopping rule, both of which are robust to noisy evaluations of the objective function. We show empirically that the proposed framework works well on a diverse set of models: it can automatically detect stochastic optimization failure or inaccurate variational approximation.

1 Introduction

Bayesian inference is a popular approach due to its flexibility and theoretical foundation in probabilistic reasoning [2, 46]. The central object in Bayesian inference is the posterior distribution of the parameter of interest given the data. However, using Bayesian methods in practice usually requires approximating the posterior distribution. Due to its computational efficiency, variational inference (VI) has become a commonly used approach for large-scale approximate inference in machine learning [26, 56]. Informally, VI methods find a simpler approximate posterior that minimizes a divergence measure $\mathcal{D}\left[q||p\right]$ from the approximate posterior q to the exact posterior distribution p – that is, they compute a optimal variational approximation $q^* = \arg\min_{q\in\mathcal{Q}} \mathcal{D}\left[q||p\right]$. The variational family is often parametrized by a vector $\boldsymbol{\lambda} \in \mathbb{R}^K$ so the parameter of q^* is given by

$$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}\in\mathbb{R}^K} \mathcal{D}\left[q_{\boldsymbol{\lambda}}||p\right]. \tag{1}$$

Variational approximations in machine learning is typically used for prediction, but recent work has shown that these approximations possess good statistical properties as point estimators and as

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

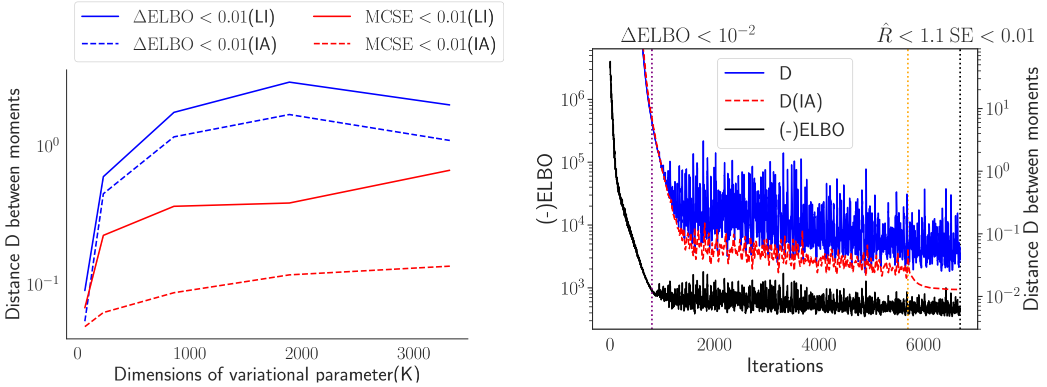


Figure 1: **(left)** The distance between the variational and ground truth moments for a full rank VI approximation on linear regression models of varying dimensions of posterior (see Section 4 for a precise definition of the distance). ΔELBO denotes the standard stopping rule, MCSE denotes our proposed stopping rule, and IA indicates that our iterate averaging approach was used while LI means the last iterate was used. IA and our proposed stopping rule both improve accuracy, particularly in higher dimensions. **(right)** The negative evidence lower bound (-ELBO) and the distances between the variational and ground truth moments based on the current iterate and using IA. The stopping point based on ΔELBO is shown by the dotted red line and occurs prematurely. Using our proposed algorithm, the starting and stopping points for IA are shown by the dotted orange and black lines, respectively.

posterior approximations [7, 39, 57, 58]. Variational inference is therefore becoming an attractive statistical method since variational approximations can often be computed more efficiently than either the maximum likelihood estimate or more precise posterior estimates – particularly when there are local latent variables that need to be integrated out. Therefore, there is a need to develop variational methods that are appropriate for statistical inference: where the model parameters are themselves the object of interest, and thus the accuracy of the approximate posterior compared to the true posterior is important. In addition, we would ideally like to refine a variational approximation further using importance sampling [23, 60] – as in the adaptive importance sampling literature [38].

Meanwhile, two developments have greatly increased the scope of the applicability of VI methods. The first is stochastic variational inference (SVI), where Eq. (1) is solved using stochastic optimization with mini-batching [21]. The increased computational efficiency of mini-batching allows SVI to scale to datasets with tens of millions of observations. The second is black box variational inference methods, which have extended variational inference to a wide range of models in probabilistic programming context by removing the need for model-specific derivations [28, 44, 51]. This flexibility is obtained by approximating local expectations and their auto-differentiated gradients using Monte Carlo approximations. While using stochastic optimization to solve Eq. (1) makes variational inference scalable as well as flexible, there is a drawback: it becomes increasingly difficult to solve the optimization problem with sufficiently high accuracy, particularly as the dimensionality of the variational parameter $\boldsymbol{\lambda}$ increases. Figure 1(left, solid lines) demonstrates this phenomenon on a simple linear regression problem where the exact posterior belongs to the variational family. Since $q^* = p$, all of the error is due to the stochastic optimization.

Because in machine learning the quality of a posterior approximation is usually evaluated by out-of-sample predictive performance, the additional error from the stochastic optimization is not necessarily problematic. Therefore, there has been less attention paid to developing stochastic optimization schemes that provide very accurate variational parameter estimates and, ideally, have good importance sampling properties too. And, as seen in Fig. 1(left, solid blue line), standard VI optimization schemes remain insufficient for statistical inference because they do not provide accurate variational parameter estimates – particularly in higher dimensions.

Moreover, existing optimizers are fragile, in that they require the choice of many hyperparameters and can fail badly. For example, the common stopping rule ΔELBO [28] is based on the change in the variational objective function value (the negative ELBO). But, as illustrated in Fig. 1(right), using ΔELBO results in termination before the optimizer converges, resulting in an inaccurate variational approximation (intersection of blue line and purple vertical line). Using a smaller cutoff for ΔELBO to ensure convergence resulted in the criterion never being met because the stochastic estimates of the negative ELBO were too noisy. To remedy this problem a combination of a smaller step size (resulting in slower convergence) and a more accurate Monte Carlo gradient estimates (resulting is greater per-iteration computation) must be used. Thus, the standard optimization algorithm is fragile due to a non-trivial interplay between its many hyperparameters, which requires the user to carefully tune all of them jointly.

In this paper, we address the shortcomings of current stochastic optimizers for VI by viewing the underlying optimization algorithm as producing a Markov chain. While such a perspective has been pursued in theoretical contexts [12, 43] and in the deep neural network literature [15, 22, 24, 35], the potential innovative algorithmic consequences of such a perspective, particularly in the VI context, have not been explored. Our Markov chain perspective allows us create more accurate variational parameter estimates by using iterate averaging, which is particularly effective in high dimensions (see red dotted lines in Fig. 1). But, even when using iterate averaging, the problems of fragility remain. In particular, we need to decide (A) when to start averaging (or when the optimizer has failed) and (B) when to terminate the optimization. For (A), we use the $\widehat{R}$ diagnostic [16, 54], a well-established method from the MCMC literature. For (B), we use Monte Carlo standard error estimates based on the chain's effective sample size (ESS) and the ESS itself [54] to ensure convergence of the parameter estimate (again drawing on a rich MCMC literature [13, 14]). We also use the $\hat{k}$ diagnostic from the importance sampling literature to check on the quality of the variational approximation and determine whether it can be used as an importance distribution [55, 60]. By combining all of these ideas, we develop an optimization framework that is robust to the selection of optimization hyperparameters such as step size and mini-batch size while also producing substantially more accurate posterior approximations. We empirically validate our proposed framework on a wide variety of models and datasets.

2 Background: Variational Inference

Let $p(\boldsymbol{y}, \boldsymbol{\theta})$ denote the joint density for a model of interest, where $\boldsymbol{y} \in \mathcal{Y}^N$ is a vector of N observations and $\boldsymbol{\theta} \in \mathbb{R}^P$ is a vector of model parameters. In this work, we assume that the observations are conditionally independent given $\boldsymbol{\theta}$; that is, the joint density factorizes as[1] $p(\boldsymbol{y}, \boldsymbol{\theta}) = \prod_{i=1}^N p(y_i|\boldsymbol{\theta})p_0(\boldsymbol{\theta})$. The goal is to approximate the resulting posterior distribution, $p(\boldsymbol{\theta}) \equiv p(\boldsymbol{\theta}|\boldsymbol{y})$, by finding the best approximating distribution $q \in \mathcal{Q}$ in the variational family $\mathcal{Q}$ as measured by a divergence measure. We focus on two commonly used variational families – the *mean-field* and the *full-rank* Gaussian families – and the standard Kullback–Leibler (KL) divergence objective, but our approach generalizes to other variational families and divergences as well. It can be shown that minimizing the KL divergence is equivalent to maximizing the functional known as the evidence lower bound (ELBO) $\mathcal{L} : \mathbb{R}^K \to \mathbb{R}$ given by [3]

$$\mathcal{L}(\boldsymbol{\lambda}) \equiv \mathbb{E}_q[\ln p(\boldsymbol{y}, \boldsymbol{\theta})] - \mathbb{E}_q[\ln q(\boldsymbol{\theta})] = \sum_{i=1}^N \left(\mathbb{E}_q[\ln p(y_i|\boldsymbol{\theta})] - \frac{1}{N}\mathrm{KL}[q||p_0]\right) = \sum_{i=1}^N \mathcal{L}_i(\boldsymbol{\lambda}),$$

where q is parametrized by $\boldsymbol{\lambda} \in \mathbb{R}^K$ and $\mathcal{L}_i(\boldsymbol{\lambda}) \equiv \mathbb{E}_q[\ln p(y_i|\boldsymbol{\theta})] - \frac{1}{N}\mathrm{KL}[q||p_0]$. The optimal approximation is $q_{\boldsymbol{\lambda}^*}$ for $\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})$.

2.1 Stochastic Optimization for VI

We will consider approximately finding $\boldsymbol{\lambda}^*$ using the stochastic optimization scheme

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \eta\gamma_t\hat{\boldsymbol{g}}_t\,, \tag{2}$$

where $\hat{\boldsymbol{g}}_t$ is an unbiased, stochastic estimator of the gradient $\mathcal{L}$ at $\boldsymbol{\lambda}_t$ (i.e., $\mathbb{E}[\hat{\boldsymbol{g}}_t] = \nabla\mathcal{L}(\boldsymbol{\lambda_t})$), η is a base step size, and $\gamma_t > 0$ is the learning rate at iteration t, which may depend on current and past iterates and gradients. The noise in the gradients is a consequence of using mini-batching, or approximating the local expectations $\mathcal{L}_i(\boldsymbol{\lambda})$ using Monte Carlo estimators, or both [21, 37, 44]. For

[1] In addition, we may have that $p(y_i|\boldsymbol{\theta}) = \int p(y_i|\boldsymbol{\theta}, z_i)p(z_i|\boldsymbol{\theta})dz_i$. But, for simplicity, we do not write the explicit dependence on the local latent variable z_i.

standard stochastic gradient descent (SGD), γ_t is a deterministic function of t only and converges asymptotically if γ_t satisfies the Robbins–Monro conditions $\sum_{t=1}^{\infty} \gamma_t = \infty$ and $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$ [45]. SGD is very sensitive to the choice of step size since too large of a step size will result in the algorithm diverging while too small of a step size will lead to very slow convergence. The shortcomings of SGD have led to the development of more robust, adaptive stochastic optimization schemes such as Adagrad [11], Adam [27, 52], and RMSProp [20], which modify the step size schedule according to the norm of current and past gradient estimates.

Even when using adaptive stochastic optimization schemes, however, it remains non-trivial to check for convergence because we only have access to unbiased estimates of the value and gradient of the optimization objective $\mathcal{L}$. Practitioners often run the optimization for a pre-defined number of iterations or use simple moving window statistics of $\mathcal{L}$ such as the running median or the running mean to test for convergence. We refer to the approach based on looking at the change in $\mathcal{L}$ as the ΔELBO stopping rule. This stopping rule can be problematic as the scale of the ELBO makes it non-trivial to specify a universal convergence tolerance ϵ. For example, Kucukelbir et al. [28] used $\epsilon = 10^{-2}$, but Yao et al. [60] demonstrate that $\epsilon < 10^{-4}$ might be needed for good accuracy. More generally, sometimes the objective estimates are too noisy relative to the chosen step size η, learning rate γ_t, threshold ϵ, and the scale of $\mathcal{L}$, which results in the stopping rule never triggering because the step size is too large relative to the threshold. The stopping rule can also trigger too early if ϵ is too large relative to η and the scale of $\mathcal{L}$. In either case, the user might have to adjust any or all of η, γ_t, and ϵ; run the optimiser again; and then hope for the best.

2.2 Refining a Variational Approximation

Another challenge with variational inference is assessing how close the variational approximation $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$ is to the true posterior distribution p. Recently, the $\hat{k}$ diagnostic has been suggested as a diagnostic for variational approximations [60]. Let $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_S \sim q_{\boldsymbol{\lambda}}$ denote draws from the variational posterior. Using (self-normalized) importance sampling we can then estimate an expectation under the true posterior as $\mathbb{E}\left[f(\boldsymbol{\theta}))\right] \approx \sum_{s=1}^{S} f(\boldsymbol{\theta}_{\boldsymbol{s}}) w(\boldsymbol{\theta}_s) / \sum_{s=1}^{S} w(\boldsymbol{\theta}_s)$, where $w(\boldsymbol{\theta}_s) \equiv p(\boldsymbol{\theta}_s|y)/q(\boldsymbol{\theta}_s)$. If the proposal distribution is far from the true posterior, the weights $w(\boldsymbol{\theta}_s)$ will have high or infinite variance. The number of finite moments of a distribution can be estimated using the shape parameter k in the generalized Pareto distribution (GPD) [55]. If $k > 0.5$, then variance of the importance sampling estimate of $\mathbb{E}\left[f(\boldsymbol{\theta})\right]$ is infinite. Theoretical and empirical results show that values below 0.7 indicate that the approximation is close enough to be used for importance sampling, while values above 1 indicate that the approximation is very poor [55].

Recent work [18] suggests that SGD iterates can converge towards a heavy tailed stationary distribution with infinite variance for even simple models (i.e. linear regression). Furthermore, even in cases that don't show infinite variance, the heavy tailed distribution may not be consistent for the mean, i.e. the mean of the stationary distribution might not coincide with the mode of the objective. In this work we again rely on $\hat{k}$ to provide an estimate of the tail index of the iterates (at convergence) and warn the user when the empirical tail index indicates a very poor approximation. We leave a more thorough study of this phenomenon for future work.

3 Stochastic Optimization as a Markov Chain

Figure 1 (left) shows that as the dimensionality of the variational parameter increases, the quality of the variational approximation degrades. To understand the source of the problem, we can view a stochastic optimization procedure as producing a discrete-time stochastic process $(\boldsymbol{\lambda}_t)_{t \geq 1}$ [5, 8, 32, 36, 59]. Under Robbins–Monro-type conditions, many stochastic optimization procedures converge asymptotically to the exact solution $\boldsymbol{\lambda}^*$ [33, 45], but any iterate $\boldsymbol{\lambda}_t$ obtained after a finite number of iterations will be a realization of a diffuse probability distribution π_t (i.e., $\boldsymbol{\lambda}_t \sim \pi_t(\boldsymbol{\lambda}_t)$) that depends on the objective function, the optimization scheme, and the number of iterations t.

We can gain further insight into the behavior of $(\boldsymbol{\lambda}_t)_{t \geq 1}$ by considering SGD with constant learning rate (that is, with $\gamma_t = 1$). Under regularity assumptions, SGD admits a stationary distribution π_∞ (that is, $\lim \pi_t = \pi_\infty$). Moreover, π_∞ will have covariance $\boldsymbol{\Sigma}_\infty$ and mean $\boldsymbol{\lambda}_\infty$ such that $\|\boldsymbol{\lambda}_\infty - \boldsymbol{\lambda}^*\| = O(\eta)$ [8]. Thus, for some sufficiently large t_0, once $t \geq t_0$ the SGD will reach approximate stationarity: $\pi_t \approx \pi_\infty$. This implies that $\mathbb{E}[\boldsymbol{\lambda}_t]$ is within $O(\eta)$ of $\boldsymbol{\lambda}^*$. However, the variance $\mathbb{V}[\boldsymbol{\lambda}_t] \approx \boldsymbol{\Sigma}$ could be large. Indeed, we expect that as the number of model parameters increase – and hence the number of variational parameters K increases – the expected squared distance from $\boldsymbol{\lambda}$ to the optimal parameter $\boldsymbol{\lambda}^*$ will increase. For example, assuming for simplicity that the stationary distribution is isotropic with $\boldsymbol{\Sigma} = \alpha^2 \boldsymbol{I}_K$ (where $\boldsymbol{I}_K$ denotes the $K \times K$ identity matrix), the expected squared distance from $\boldsymbol{\lambda}$ to the optimal value is given by $\mathbb{E}[\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^*\|^2] = \alpha^2 K + O(\eta^2)$. Therefore, we should expect distance between $\boldsymbol{\lambda}_t$ and $\boldsymbol{\lambda}^*$ to be $O(\sqrt{K})$, which implies that the variational parameter estimates output by SGD become increasingly inaccurate as the dimensionality of the variational parameter increases. As demonstrated in Fig. 1(left), one should be particularly careful when fitting a full-rank variational family since the number of parameters is $K = P(P+1)/2$.

Although the preceding discussion only applies directly to SGD, it is reasonable to expect that robust stochastic optimization schemes such as Adagrad, Adam, and RMSprop will have similar behavior as long as γ_t and $\hat{\boldsymbol{g}}_t$ depend at most very weakly on iterates far in the past.

3.1 Improving Optimization Accuracy with Iterate Averaging

While we have shown that we should not expect a single iteration $\boldsymbol{\lambda}_t$ to be close to $\boldsymbol{\lambda}^*$ in high-dimensional settings, the expected value of $\boldsymbol{\lambda}_t$ *is* equal to (or, more realistically, close to) $\boldsymbol{\lambda}^*$. Therefore, we can use *iterate averaging* (IA) to construct a more accurate estimate of $\boldsymbol{\lambda}^*$ given by

$$\bar{\boldsymbol{\lambda}} \equiv \tfrac{1}{T} \textstyle\sum_{i=1}^{T} \boldsymbol{\lambda}_{t+i}, \tag{3}$$

where we should aim to choose $t \geq t_0$. In the fixed step-size setting described above, the estimator $\bar{\boldsymbol{\lambda}}$ has bias of order η and covariance $\mathbb{V}[\bar{\boldsymbol{\lambda}}] \approx \boldsymbol{\Sigma}/T + 2\sum_{1 \leq i < j \leq T} \operatorname{cov}[\boldsymbol{\lambda}_{t+i}, \boldsymbol{\lambda}_{t+j}]/T^2$. Hence, as long as the iterates $\boldsymbol{\lambda}_t$ are not too strongly correlated, we can reduce the variance and alleviate the effect of dimensionality by using iterative averaging.

Iterate averaging has been previously considered in a number of scenarios. Ruppert [50] proposes to use a moving average of SGD iterates to improve SGD algorithms in the context of linear one-dimensional models. Polyak and Juditsky [42] extend the moving average approach to multi-dimensional and nonlinear models, and showed that it improved the rate of convergence in several important scenarios; thus, it is often referred to as Polyak–Ruppert averaging. In related work, Bach and Moulines [1] show that an averaged stochastic gradient scheme with constant step size can achieve optimal convergence for linear models even for (non-strongly) convex optimization objectives. Recent work demonstrates that averaging iterates can help improve generalization in deep neural networks [15, 22, 24, 35]; note, however, that our application of IA aims not just to improve predictive accuracy but also the accuracy of the posterior approximation.

3.2 Making Iterate Averaging Robust

In order to make iterate averaging robust in practice, we must (1) ensure that the distributions of the iterates have finite variance, and (2) determine effective, automatic ways to set the two (implicit) free parameters of $\bar{\boldsymbol{\lambda}}$: t (when to start averaging) and T (how many iterates to average). #1 is crucial since otherwise even computing a Monte Carlo estimate $\bar{\boldsymbol{\lambda}}$ is questionable. We use an approach based on the $\hat{k}$ statistic (see Line 9 of Algorithm 1); since in our experiments we did not find any cases of infinite-variance iterates, we defer further discussion of our approach to the Supplementary Material. This use of $\hat{k}$ over the process' iterates is not to be confused with our application of $\hat{k}$ to determine the quality of the variational approximation that we compute after the optimization. For #2, recall that our Markov chain perspective suggests that we should start averaging at $t > t_0$, where t_0 denotes the iteration after which the distribution of $\boldsymbol{\lambda}_t$ has approximately reached stationarity and therefore is near the optimum [25, 47]. We must then select T large enough that $\bar{\boldsymbol{\lambda}}$ is sufficiently close to $\boldsymbol{\lambda}^*$. We address how to robustly choose t and T in turn.

Determining when to start averaging Previous approaches to selecting t rely on the so-called Pflug criterion [6, 41, 48], which is based on evaluating the sum of the inner product of successive gradients. Unfortunately this approach is not robust and can be slow to detect convergence [40]. To develop an alternative, robust approach to selecting t we turned to the Markov chain Monte Carlo literature. In MCMC, the $\widehat{R}$ statistic is a canonical way to determine if a Markov chain have reached stationarity [16, 17, 54]. The standard approaches to computing $\widehat{R}$ is to use multiple Markov chains. If we have J chains and N iterates in each chain, $\boldsymbol{\lambda}_i^{(j)}$, such that $i = 1, \ldots, N; j = 1, \ldots, J$, then

$\widehat{R} \equiv (\hat{\mathbb{V}}/\hat{\mathbb{W}})^{1/2}$, where $\hat{\mathbb{V}}$ and $\hat{\mathbb{W}}$ are estimates of, respectively, the between-chain and within-chain variances. We use the split-$\widehat{R}$ version, where all chains are split into two before carrying out the computation above, which helps with detecting non-stationarity [17, 54] and allows us to use it even when $J = 1$.

In order to utilize $\widehat{R}$, we run J optimization runs ("chains") in parallel and consider the iterates at stationarity when $\widehat{R} < \tau$, where $\tau > 1$ is a user-chosen cutoff. We select a moving window and only use the most recent $a \times t$ samples for computing $\widehat{R}$ (where $0 < a \leq 1$ and t is the current iterations counter), since we do not expect iterates before the (unknown) t_0 to be close to the stationary distribution. There is a trade-off between making a large, which leads to more accurate and potentially smaller estimates for $\widehat{R}$, and making a small, which leads to more quickly determining when the iterates are near stationarity, but more noisy estimate. In practice we found $a = 0.5$ to be a good choice, although somewhat larger or smaller values would work as well. $a = 0.5$ is also the most commonly used window size in MCMC literature. Concerning the choice of the cutoff τ, in the MCMC literature $\widehat{R}$ is required to be very precise since the stationary distribution is the true posterior, so $\tau = 1.01$ or even smaller is recommended [53, 54]. In our case, since we are less concerned about the quality of the stationary distribution, we use $\tau = 1.2$. The algorithm is robust for values even upto 1.4. $\widehat{R}$ is computed after every W iteration.

Determining when to stop averaging Once $t > t_0$ is found using $\widehat{R}$, we must determine how many iterates to average. Since all J optimizations are guaranteed to reach the same optimum (if there are no local optima) due to our use of $\widehat{R}$, we can combine the iterates into a single variational parameter estimate $\bar{\boldsymbol{\lambda}} = \sum_{j=1}^{J} \sum_{i=1}^{T} \boldsymbol{\lambda}_{t+i}^{(j)}/(JT)$, where $\boldsymbol{\lambda}_{s}^{(j)}$ the sth iterate of the jth chain.

Due to the non-robustness of the ΔELBO stopping rule, we propose an alternative stopping criterion that is robust to the (unknown) scale of the objective and which accounts for the fact that the variational parameter is the quantity of interest, not the value of the objective function. Again turning to the MCMC literature and taking advantage of our iterative averaging approach, we propose to use the Monte Carlo standard error (MCSE) [14, 19, 54], which is given as $\text{MCSE}(\lambda_i) \equiv \{\mathbb{V}(\lambda_i)/\text{ESS}(\lambda_i)\}^{1/2}$, where $\mathbb{V}(\lambda_i)$ is the variance of the ith component of the iterates, $\text{ESS} \equiv JN/(1 + \sum_{t=1}^{\infty} 2\rho_t)$ is the effective sample size (ESS), N is the number of iterations after $\widehat{R}$ convergence (used to compute the variance), and ρ_t is the autocorrelation at lag t. The ESS accounts for the dependency between iterates and in general we expect it to be smaller than the total number of iterates JN. We compute the ESS using the method described in Vehtari et al. [54]. In addition to checking that the median value of the $\text{MCSE}(\lambda_i)$ is below some tolerance ϵ, to ensure the MCSE estimates are actually reliable, we also require that all of the effective sample sizes are above a threshold e.

We note that a benefit of our approach is that the MCSE also provides an estimate of how many significant figures in the parameter estimate $\bar{\boldsymbol{\lambda}}$ are reliable. Such reliability estimates are particularly important in high dimensions since, as we will see (Section 4 and Table 1), even small perturbations to the location or scale parameters can result in a very bad approximation to the posterior distribution.

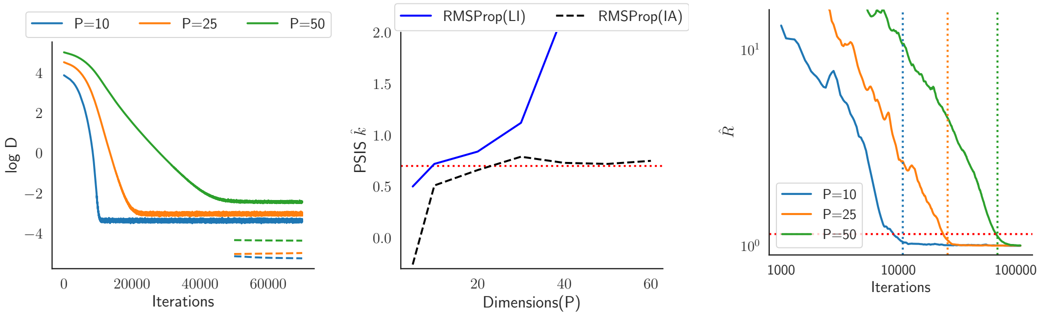
Diagnosing convergence problems with autocorrelation values The autocorrelation values ρ_t that are computed when estimating ESS can also used as a diagnostic if $\widehat{R}$ is not falling below τ or the MCSE is not decreasing when more iterations are averaged. Large autocorrelations before $\widehat{R} < \tau$ may indicate that the window a needs to be increased in order to estimate $\widehat{R}$ effectively. Large autocorrelations after averaging has started suggests iterate averaging may not be reliable.

4 Experiments

We now turn to validating our robust stochastic optimization algorithm for variational inference (summarized in Algorithm 1) through experiments on both simulated and real-world data. In our experiments we used $\eta = 0.01, W = 100, a = 0.5, \tau = 1.2$, and $e = 20$. To ensure a fair comparison to the ΔELBO stopping rule, we used $J = 1$ in all of our experiments; the exception is that Fig. 2 used $J = 4$ since it does not involve a comparison to ΔELBO. We also put ΔELBO at an advantage by doing some tuning of the threshold ϵ, while keeping $\epsilon = 0.02$ when using

Algorithm 1 Robust Stochastic Optimization for Variational Inference

```
Input: learning rate $\eta$, # of optimization runs $J$, window size $a$, evaluation window $W$, $\widehat{R}$ cutoff
    $\tau$, MCSE cutoff $\epsilon$, ESS cutoff $e$, iterate initalizations $\boldsymbol{\lambda}_0^{(j)}$ for $j = 1, \ldots, J$
for $t \leftarrow 1$ to $T_{\text{max}}$ do
    Compute $\boldsymbol{\lambda}_t^{(j)}$ via Eq. (2), $j = 1, \ldots, J$
    if $t \bmod W = 0$ then
        Compute $\widehat{R}_i$, the $\widehat{R}$ value for the $i$th component of $\boldsymbol{\lambda}$        ▷ using last $at$ iterates
        if $\max_i \widehat{R}_i < \tau$ then
            $T_0 \leftarrow t$
            break
if $\max_i \widehat{R}_i < \tau$ or $\hat{k}$ of iterates $> 1.0$ then
    Warn user that optimization may not have converged
    return $\bar{\boldsymbol{\lambda}}$ computed from the last $W$ iterates
else
    for $t \leftarrow T_0$ to $T_{\text{max}}$ do
        Compute $\boldsymbol{\lambda}_t^{(j)}$ via Eq. (2), $j = 1, \ldots, J$
        if $t - T_0 \bmod W = 0$ and MCSE $< \epsilon$ and ESS $> e$ then        ▷ using last $t - T_0$ iterates
            break
    return $\bar{\boldsymbol{\lambda}}$ computed from the last $t - T_0$ iterates
```



(a) (b) (c)

Figure 2: For the linear regression model with posterior correlation 0.9, the evolution of **(a)** moment distance D, **(b)** $\hat{k}$ statistic, and **(c)** $\widehat{R}$ statistic during optimization. For D and $\hat{k}$ (of the variational approximation) we show the values for the last iterate (solid lines) and averaged iterates (dashed lines).

our MCSE criterion. We show the results based on using RMSprop, but we found that AdaGrad performed similarly (see Supplementary Material). For the variational approximation family we used multivariate Gaussians $q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{m}_q = \boldsymbol{\mu}, \boldsymbol{\Sigma}_q = \boldsymbol{L}\boldsymbol{L}^T)$ where $\boldsymbol{L}$ is the Cholesky decomposition of the covariance matrix. We used `viabel` [23] for inference, TensorFlow Probability [9] and Stan [4] for model-construction, and `arviz` [29] for tail-index estimation.

The linear regression experiments with synthetic data mentioned in Section 1 (and described in detail in the Supplementary Material) provide a useful case study of stochastic variational inference where the true posterior distribution belongs to the variational family, meaning that any inaccuracy in the variational approximation was due to the stochastic optimization procedure. We also investigated a variety of models and datasets using black box variational inference: logistic regression [61] on three UCI datasets (Boston, Wine, and Concrete [10]); a high-dimensional hierarchical Gaussian model (Radon [34]), the 8-school hierarchical model [49], and a Bayesian neural network model with 10 hidden units and 2 layers [30] to classify 100 handwritten digits from the MNIST dataset [31] (MNIST100). The 8-school model has a significantly non-Gaussian posterior and has served as a test case in a number of recent variational inference papers [23, 60]. We considered both the centered parameterization (CP) and non-centered one (NCP) because the NCP version of 8-school is easier to approximate with variational methods [23, 60], and therefore experiments on both provide insight into the robustness of a variational algorithm. We also experiment with a four layer normalising flow (NF) model to fit the 8-school posterior, which gave the best estimate for posterior mean in all experiments with 8-school, with iterate averaging. For all real-data experiments we estimated the ground-truth posterior moments (i.e., the mean μ and covariance matrix Σ) using the dynamic Hamiltonian Monte Carlo algorithm in Stan [4]. We used these to compute the normalized moment distance $D \equiv (D_{\boldsymbol{\mu}}^2 + D_{\boldsymbol{\Sigma}}^2)^{1/2}$, where $D_{\boldsymbol{\mu}} \equiv \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2$, $D_{\boldsymbol{\Sigma}} \equiv \|\boldsymbol{\Sigma} - \hat{\boldsymbol{\Sigma}}\|^{1/2}$ and $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ denote, respectively, the variational estimates of the posterior mean and covariance.

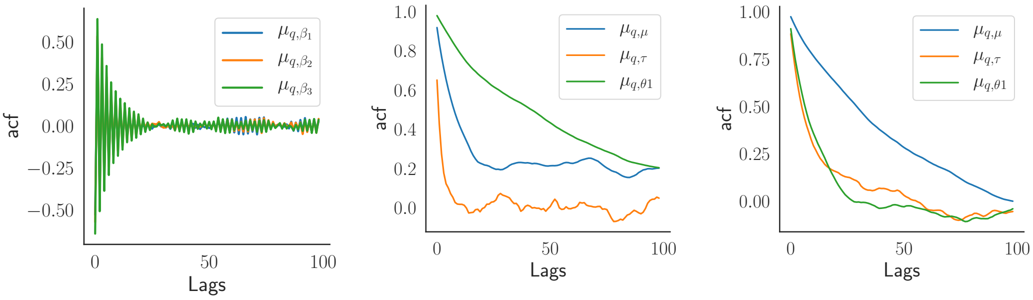
Iterate averaging improves variational parameter estimates First we investigated the benefits of using iterate averaging rather than the final iterate. For the linear regression model, Fig. 1 shows the benefits of IA when using either ΔELBO or MCSE as a stopping criteria, with a larger gain coming from its use with MCSE (and $\widehat{R}$) since in that case the iterates were closer to the optimum. Figure 1(right) shows the improved accuracy of iterate averaging compared to using the last iterate in detail for the case when the dimension of the linear regression model was $P = 70$. Figures 2a and 2b provides a further example of the benefits of iterate averaging for linear regression in the more challenging case of strong posterior correlation. IA provides an approximately two orders of magnitude improvement in accuracy. The improvement in importance sampling performance is also dramatic: while the $\hat{k}$ statistic for the variational approximation after the last iterate is above the 0.7 reliability threshold even when with data of dimension $P = 10$, the $\hat{k}$ statistic of IA remains below or near the 0.7 when $P = 60$.

Table 1 shows that in our real-data experiments, IA almost universally outperforms the last iterate when using Algorithm 1, both in terms of moment estimates and approximation's $\hat{k}$; however, because the ΔELBO stopping rule sometimes resulted in premature termination of the optimizer, IA did not always provide a benefit with ΔELBO, which lends further support for using our more comprehensive robust optimization framework. The only exception was the (multimodal) MNIST100 posterior, where for MCSE the $\hat{k}$ statistic for the last iterate was superior to that for IA – although both were very large.

MCSE stopping criteria improves robustness and accuracy Recall that Fig. 1 (left) provides an case where the ΔELBO stopping rule results in premature termination of the optimizer. For the real-data examples, in Table 1 we see that due to substantially earlier termination (small T), using ΔELBO consistently results is less accurate posterior approximations in terms of moment estimates and $\hat{k}$. The only exception is the Radon model, which never reaches convergence according to the ΔELBO criterion and, as a result, produces better posterior mean accuracy and a smaller $\hat{k}$ statistic

Table 1: Real-data results comparing the ΔELBO stopping rule to our proposed MCSE stopping rule (which implements all of Algorithm 1). K = number of variational parameters,and T = total number of iterations before termination. $\star$ denotes that convergence was not reached after $T_{\max}$ iterations. Rule=Stopping Rule, 8-s.=eight school, E=ELPD

Model	K	Rule	T	$D_{\boldsymbol{\mu}}$	$D_{\boldsymbol{\mu}}$ (IA)	$D_{\boldsymbol{\Sigma}}$	$D_{\boldsymbol{\Sigma}}$ (IA)	$\hat{k}$	$\hat{k}$ (IA)	E	E(IA)
Boston	104	ΔELBO	2100	0.02	0.008	0.06	0.38	0.90	11	−95	−120
		MCSE	5900	0.003	**0.001**	0.008	**0.004**	0.55	**0.06**	−79	**-78**
Wine	77	ΔELBO	2400	0.005	0.004	0.017	0.11	0.78	15	−435	**-410**
		MCSE	5300	0.002	**0.001**	0.0006	**0.00003**	0.70	**0.07**	−424	−425
Concrete	44	ΔELBO	1800	0.02	0.04	0.018	0.51	2.7	15	−158	−170
		MCSE	3900	0.015	**0.001**	0.02	**0.004**	0.74	**0.09**	−152	**-151**
8-s. (CP)	65	ΔELBO	1100	1.9	4.5	**3.5**	5.8	0.98	0.85		
		MCSE	6200	2.1	**1.8**	**3.5**	3.7	0.88	**0.78**		
8-s. (NCP)	65	ΔELBO	1700	0.12	**0.09**	1.02	1.02	0.60	0.60		
		MCSE	2400	0.14	0.13	1.05	**0.98**	**0.58**	0.63		
8-s. (NF)	84	ΔELBO	800	0.17	0.18	1.89	2.01	0.70	0.72		
		MCSE	7500	0.17	**0.06**	1.48	**1.27**	0.67	0.64		
Radon	4094	ΔELBO	*15000	5.8	**5.7**	0.80	**0.40**	1.2	**0.34**		
		MCSE	9500	6.0	5.9	1.2	1.1	1.3	0.40		
MNIST100	7951	ΔELBO	1200	82.7	83.7	34.1	34.1	33	32		
		MCSE	*10000	**33.6**	51.0	**34**	**34**	**7.0**	11		



(a) Linear Regression

(b) 8-school (CP)

(c) 8-school (NCP)

Figure 3: Autocorrelation plots for **(a)** the location parameters for weights:β_1, β_2, and β_3 for linear regression using a mean-field variational family and **(b,c)** the location parameters of μ, τ and θ_1 for 8-schools centered and non-centered parameterisations. The plots serve as a diagnostic tool for assessing the efficiency of averaging.

than using MCSE. On the other hand, MCSE runs for approximately half as many iterations, still has a $\hat{k}$ statistic less than 0.5, and produces a more accurate posterior mean estimate. The threshold $\epsilon = 0.02$ was kept the same for all the datasets in case of MCSE, roughly of the same order as the step size, and we found it to be quite robust compared to ΔELBO. We also report Expected Log Predictive Density for the UCI datasets, our algorithm obtains a better ELPD on two of the datasets.

Autocorrelation and $\hat{k}$ detect problematic variational approximations Figure 3 provides an example where, for linear regression, the oscillation in the autocorrelation plot indicates super-efficiency in the averaging due to negative correlation in odd lags [54]. Supplementary Figures 1b and 1c provide examples where, for the 8-school models (both CP and NCP), the iterates are heavily correlated and thus averaging is less efficient, which is reflected in the less dramatic benefits of using IA (Table 1). The $\hat{k}$ statistics (Table 1) provide good guidance of approximation accuracy.

$\widehat{R}$ detects optimization failure Figures 1 and 2c and Table 1 provide examples where $\widehat{R}$ successfully detects convergence of the optimization. Just as importantly, $\widehat{R}$ can also diagnose optimization problems such as multi-modality. For example, if the variational objective has multiple (local) optima, different optimizations can end up in different optima due to by random initialization; but this would be indicated by a large $\widehat{R}$. For example, when we used Algorithm 1 with $J = 4$ for the multimodal MNIST100 model, the maximum $\widehat{R}$ was 4.8. This result also provides support for using $J > 1$ parallel optimizations, since such multimodality cannot be detected when $J = 1$. A direction for future work would be to approximate a multimodal posterior by extending our approach to analyze the convergence in each mode and then combine results of different modes (e.g., by stacking weights [60]).

Acknowledgements

We would like to thank Ben Bales for useful discussions about the $\widehat{R}$ statistic and the anonymous reviewers for their helpful suggestions. We thank Academy of Finland (grants 298742, and 313122) and Finnish Center for Artificial Intelligence for partial support of the research. We also acknowledge the computational resources provided by the Aalto Science-IT project.

Broader impact

There are sometimes misconceptions about how fast or accurate variational inference can be for Bayesian inference. In this paper, we show potential pitfalls of current practices that may lead to incorrect conclusions, especially when the interest of the user is more focused on inference than prediction. More robust and reliable inference makes data analysis for decision-making by scientists and organizations (e.g., corporations, governments, and foundations) more reliable and reproducible.

Whether such improvements in decision-making quality lead to better outcomes for society will depend upon the goals of the organization or person. On net, however, we expect more reliable data analysis to be for the good.

References

[1] F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate o(1/n). In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 773–781. Curran Associates, Inc., 2013.

[2] J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. Wiley, New York, 2000.

[3] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[4] B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddel. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76:1 –32, 2017.

[5] J. Chee and P. Toulis. Convergence diagnostics for stochastic gradient descent with constant learning rate. In *International Conference on Artificial Intelligence and Statistics*, 2018.

[6] J. Chee and P. Toulis. Convergence diagnostics for stochastic gradient descent with constant learning rate. In A. Storkey and F. Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1476–1485, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

[7] B.-E. Chérief-Abdellatif and P. Alquier. Consistency of variational Bayes inference for estimation and model selection in mixtures. *Electronic Journal of Statistics*, 12(2):2995–3035, 2018.

[8] A. Dieuleveut, A. Durmus, and F. Bach. Bridging the Gap between Constant Step Size Stochastic Gradient Descent and Markov Chains. *The Annals of Statistics*, 48(3):1348–1382, 2020.

[9] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. D. Hoffman, and R. A. Saurous. Tensorflow distributions. *CoRR*, abs/1711.10604, 2017, 1711.10604.

[10] D. Dua and C. Graff. UCI machine learning repository, 2017.

[11] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July 2011. ISSN 1532-4435.

[12] M. A. Erdogdu, L. Mackey, and O. Shamir. Global Non-convex Optimization with Discretized Diffusions. In *Advances in Neural Information Processing Systems*, 2018.

[13] J. M. Flegal. Monte Carlo standard errors for Markov chain Monte Carlo. *PhD Thesis*, 2008.

[14] J. M. Flegal, M. Haran, and G. L. Jones. Markov chain Monte Carlo: Can we trust the third significant figure? *Statistical Science*, 23(2):250–260, 2008. ISSN 08834237.

[15] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8789–8798. Curran Associates, Inc., 2018.

[16] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statist. Sci.*, 7(4):457–472, 11 1992.

[17] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis, third edition*. CRC Press, 2013.

[18] M. Gurbuzbalaban, U. Simsekli, and L. Zhu. The heavy-tail phenomenon in sgd, 2020.

[19] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[20] G. Hinton, N. Srivastava, and K. Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent, 2012.

[21] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

[22] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. Hopcroft, and K. . Weinberger. Snapshot ensembles: Train 1, get m for free. In *International Conference on Learning Representations*, 2017.

[23] J. H. Huggins, M. Kasprzak, T. Campbell, and T. Broderick. Validated Variational Inference via Practical Posterior Error Bounds. In *AISTATS*, Oct. 2019.

[24] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. Wilson. Averaging weights leads to wider optima and better generalization. *Uncertainty in Artificial Intelligence - Proceedings, UAI 2018*, 2018.

[25] P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford. Parallelizing stochastic gradient descent for least squares regression: Mini-batching, averaging, and model misspecification. *Journal of Machine Learning Research*, 18(223):1–42, 2018.

[26] M. I. Jordan, Z. Ghahramani, and et al. An introduction to variational methods for graphical models. In *Machine Learning*, pages 183–233. MIT Press, 1999.

[27] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[28] A. Kucukelbir, R. Ranganath, A. Gelman, and D. Blei. Automatic variational inference in stan. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 568–576. Curran Associates, Inc., 2015.

[29] R. Kumar, C. Colin, A. Hartikainen, and O. A. Martin. ArviZ a unified library for exploratory analysis of Bayesian models in Python. *The Journal of Open Source Software*, 2019.

[30] J. Lampinen and A. Vehtari. Bayesian approach for neural networks—review and case studies. *Neural networks*, 14(3):257–274, 2001.

[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[32] X. Li and F. Orabona. On the Convergence of Stochastic Gradient Descent with Adaptive Stepsizes. In *International Conference on Artificial Intelligence and Statistics*, 2019.

[33] X. Li and F. Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In K. Chaudhuri and M. Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 983–992. PMLR, 16–18 Apr 2019.

[34] C. Lin, A. Gelman, P. Price, and D. Krantz. Analysis of local decisions using hierarchical modeling, applied to home radon measurement and remediation. *Statistical Science*, 14, 08 1999.

[35] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pages 13132–13143, 2019.

[36] S. Mandt, M. D. Hoffman, and D. M. Blei. Stochastic gradient descent as approximate bayesian inference. *J. Mach. Learn. Res.*, 18(1):4873–4907, Jan. 2017. ISSN 1532-4435.

[37] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih. Monte carlo gradient estimation in machine learning. *arXiv preprint arXiv:1906.10652*, 2019.

[38] M.-S. Oh and J. O. Berger. Adaptive Importance Sampling in Monte Carlo Integration. *Journal of Statistical Computation and Simulation*, 41:143–168, 1992.

[39] D. Pati, A. Bhattacharya, and Y. Yang. On Statistical Optimality of Variational Bayes. In *International Conference on Artificial Intelligence and Statistics*, 2018.

[40] S. Pesme, A. Dieuleveut, and N. Flammarion. On convergence-diagnostic based step sizes for stochastic gradient descent, 2020.

[41] G. C. Pflug. Non-asymptotic confidence bounds for stochastic approximation algorithms with constant step size. *Monatshefte für Mathematik*, 110(3-4):297–314, 1990.

[42] B. T. Polyak and A. B. Juditsky. Acceleration of stoachastic approximation by averaging. *SIAM Journal on Control and Optimization.*, 1992.

[43] M. Raginsky, A. Rakhlin, and M. Telgarsky. Non-convex learning via Stochastic Gradient Langevin Dynamics - a nonasymptotic analysis. In *Conference on Learning Theory*, 2017.

[44] R. Ranganath, S. Gerrish, and D. Blei. Black Box Variational Inference. In S. Kaski and J. Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 814–822, Reykjavik, Iceland, 22–25 Apr 2014. PMLR.

[45] H. Robbins and S. Monro. A stochastic approximation method. In *The Annals of Mathematical Statistics.*, 1951.

[46] C. P. Robert. *The Bayesian Choice*. Springer, New York, NY, 2nd edition edition, 2007.

[47] N. L. Roux. Anytime tail averaging. *arXiv preprint arXiv:1902.05083*, 2019.

[48] N. L. Roux, M. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, pages 2663–2671, USA, 2012. Curran Associates Inc.

[49] D. B. Rubin. Estimation in parallel randomized experiments. *Journal of Educational Statistics*, 6(4):377–401, 1981. ISSN 03629791.

[50] D. Ruppert. Efficient estimations from a slowly convergent robbins-monro process. *Technical report, Cornell University Operations Research and Industrial Engineering*, 1988.

[51] M. Titsias and M. Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1971–1979, Bejing, China, 22–24 Jun 2014. PMLR.

[52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[53] D. Vats and C. Knudson. Revisiting the Gelman-Rubin Diagnostic. *arXiv.org*, Dec. 2018, 1812.09384.

[54] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P.-C. Bürkner. Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC. *arXiv preprint arXiv:1903.08008*, 2019.

[55] A. Vehtari, D. Simpson, A. Gelman, Y. Yuling, and J. Gabry. Pareto smoothed importance sampling. *arXiv preprint arXiv:1507.02646*, 2019.

[56] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1–2):1–305, Jan. 2008. ISSN 1935-8237.

[57] Y. Wang and D. M. Blei. Frequentist Consistency of Variational Bayes. *Journal of the American Statistical Association*, 17(239):1–86, June 2018.

[58] Y. Wang and D. M. Blei. Variational Bayes under Model Misspecification. In *Advances in Neural Information Processing Systems*, 2019.

[59] S. Yaida. Fluctuation-dissipation relations for stochastic gradient descent. In *International Conference on Learning Representations*, 2019.

[60] Y. Yao, A. Vehtari, D. Simpson, and A. Gelman. Yes, but did it work?: Evaluating variational inference. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5581–5590, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[61] I.-C. Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808, 1998.

Publication IV

Akash Kumar Dhaka, Alejandro Catalina, Michael Riis Andersen, Manushi Welandawe, Jonathan Huggins, Aki Vehtari. Challenges and Opportunities in High-dimensional Variational Inference. *Advances in Neural Information Processing Systems*, Volume 34, 2021.

Challenges and Opportunities in High-dimensional Variational Inference

Akash Kumar Dhaka*
Aalto University, Silo AI
akash.dhaka@aalto.fi

Alejandro Catalina*
Aalto University
alejandro.catalina@aalto.fi

Manushi Welandawe
Boston University
manushi.welandawe@bu.edu

Michael Riis Andersen
Technical University of Denmark
miri@dtu.dk

Jonathan H. Huggins
Boston University
huggins@bu.edu

Aki Vehtari
Aalto University
aki.vehtari@aalto.fi

Abstract

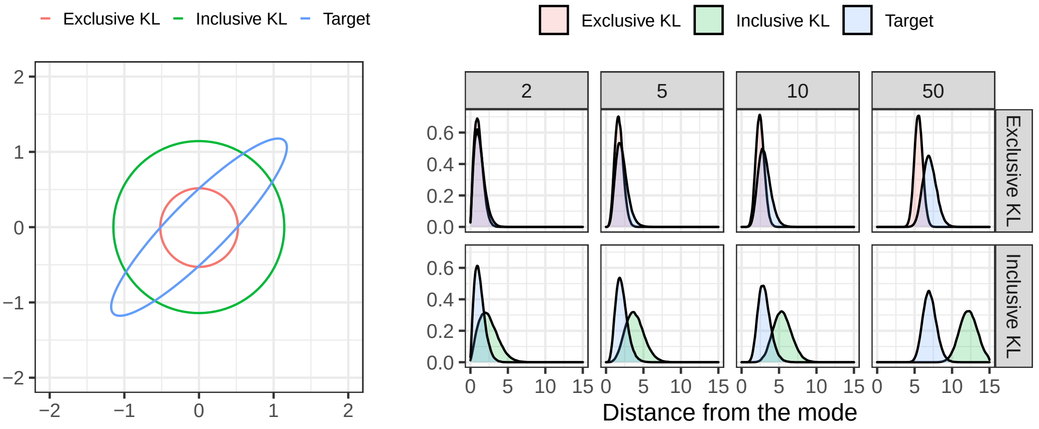
Current black-box variational inference (BBVI) methods require the user to make numerous design choices—such as the selection of variational objective and approximating family—yet there is little principled guidance on how to do so. We develop a conceptual framework and set of experimental tools to understand the effects of these choices, which we leverage to propose best practices for maximizing posterior approximation accuracy. Our approach is based on studying the pre-asymptotic tail behavior of the density ratios between the joint distribution and the variational approximation, then exploiting insights and tools from the importance sampling literature. Our framework and supporting experiments help to distinguish between the behavior of BBVI methods for approximating low-dimensional versus moderate-to-high-dimensional posteriors. In the latter case, we show that mass-covering variational objectives are difficult to optimize and do not improve accuracy, but flexible variational families can improve accuracy and the effectiveness of importance sampling—at the cost of additional optimization challenges. Therefore, for moderate-to-high-dimensional posteriors we recommend using the (mode-seeking) exclusive KL divergence since it is the easiest to optimize, and improving the variational family or using model parameter transformations to make the posterior and optimal variational approximation more similar. On the other hand, in low-dimensional settings, we show that heavy-tailed variational families and mass-covering divergences are effective and can increase the chances that the approximation can be improved by importance sampling.

1 Introduction

A great deal of progress has been made in black-box variational inference (BBVI) methods for Bayesian posterior approximation, but the interplay between the approximating family, divergence measure, gradient estimators and stochastic optimizer is non-trivial – and even more so for high-dimensional posteriors [1, 10, 29, 31]. While the main focus in the machine learning literature has

[*] Equal contribution.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).



(a) (b)

Figure 1: Illustration of a mean-field approximation with exclusive (mode-seeking) and inclusive (mass-covering) divergences. **(a)** The typical 2D illustration (correlation 0.9) gives the impression that the inclusive divergence would provide a better approximation. **(b)** For correlated Gaussian targets in dimensions $D = 2, 5, 10, 50$, the marginal distributions of the distance from the mode for samples drawn from the approximation (red) and the target (blue). The intuition from the low-dimensional examples does not carry over to higher dimensions: although the importance ratios are still bounded, even for a lower correlation level, the overlap in typical sets of the target and the approximations gets worse both for exclusive and inclusive divergences.

been on improving predictive accuracy, the choice of method components becomes even more critical when the goal is to obtain accurate summaries of the posterior itself.

In this paper, we show that, while the choice of approximating family and divergence is often motivated by low-dimensional illustrations, the intuition from these examples do not necessarily carry over to higher-dimensional settings. By drawing a connection between importance sampling and the estimation of common divergences used in BBVI, we are able to develop a comprehensive framework for understanding the reliability of BBVI in terms of the *pre-asymptotic* behavior of the density ratio between the target and the approximate distribution. When this density ratio is heavy-tailed, even unbiased estimators exhibit a large bias with high probability, in addition to high variance. Such heavy tails occur when there is a mismatch between the typical sets of the approximating and target distributions. In higher dimensions, even over-dispersed distributions miss the typical set of the target [18, 27]. Thus, as illustrated in Fig. 1, the benefits of heavy-tailed approximate families and divergences favoring mass-covering diminish as dimensionality of the target distribution increases. Building on these insights, we make the following main contributions:

1. We develop a conceptual and experimental framework for predicting and empirically evaluating the reliability of BBVI based on the choice of variational objective, approximating family, and target distribution. Our framework also incorporates the Pareto k diagnostic [27] as a simple and practical approach for obtaining empirical and conceptual insights into the pre-asymptotic convergence rates of estimators of common divergences and their gradients.

2. We validate our framework through an extensive empirical study using simulated data and many commonly used real datasets with both Gaussian and non-Gaussian target distributions. We consider the exclusive and inclusive Kullback-Leibler (KL) divergences [4, 21, 24], tail-adaptive f-divergence [29], χ^2 divergence [7], and α-divergences [12], and the resulting variational approximation for isotropic Gaussian and Student-t and normalising flow families.

3. Based on our framework and numerical results, we provide justified recommendations on design choices for different scenarios, including low- to moderate-dimensional and high-dimensional posteriors.

2 Preliminaries and Background

Let $p(\theta, Y)$ be a joint distribution of a probabilistic model, where $\theta \in \mathbb{R}^D$ is a vector of model parameters and Y is the observed data. In Bayesian analysis, the posterior $p(\theta) := p(\theta \mid Y) = p(\theta, Y)/p(Y)$ (where $p(Y) := \int p(\theta, Y) d\theta$) is typically the object of interest, but most posterior summaries of interest are not accessible because the normalizing integral, in general, is intractable. Variational inference approximates the exact posterior $p(\theta \mid Y)$ using a distribution $q \in \mathcal{Q}$ from a family of tractable distributions $\mathcal{Q}$. The best approximation is determined by minimizing a divergence $D(p \parallel q)$, which measures the discrepancy between p and q:

$$q_{\lambda^*} = \arg\min_{q_\lambda \in \mathcal{Q}} D(p \parallel q), \tag{1}$$

where $\lambda \in \mathbb{R}^K$ is a vector parameterizing the variational family $\mathcal{Q}$. Thus, the properties of the resulting approximation q are determined by the choice of variational family $\mathcal{Q}$ as well as the choice of divergence D.

The family $\mathcal{Q}$ is often chosen such that quantities of interest (e.g., moments of q) can be computed efficiently. For example, q can be used to compute Monte Carlo or importance sampling estimates of the quantities of interest. Let $w(\theta) := p(\theta, Y)/q(\theta)$ denote the density ratio between the joint and approximate distributions. For a function $\phi : \mathbb{R}^D \to \mathbb{R}$, the biased *self-normalized importance sampling estimator* for the posterior expectation $\mathbb{E}_{\theta \sim p}[\phi(\theta)]$ is given by

$$\hat{I}(\phi) := \sum_{s=1}^{S} \frac{w(\theta_s)}{\sum_{s'=1}^{S} w(\theta_{s'})} \phi(\theta_s),$$

where $\theta_1, \ldots, \theta_S \sim q$ are independent. Using importance sampling can allow for computation of more accurate posterior summaries and to go beyond the limitations of the variational family. For example, it is possible to estimate the posterior covariance even when using a mean-field variational family.

Pareto Smoothed Importance Sampling. Since importance sampling estimates can have very high variance, Pareto smoothed importance sampling (PSIS) can be used to substantially reduce the variance with small additional bias [27]. This procedure modifies and stabilises extreme importance ratios using a generalized Pareto distribution fit to the upper tail of the distribution of the ratios.

Variational families. Let $q_\lambda(\theta)$ be an approximating family parameterised by a K-dimensional vector $\lambda \in \mathbb{R}^K$ for D-dimensional inputs $\theta \in \mathbb{R}^D$. Typical choices of q include mean-field Gaussian and Student's t families [3, 14], full and low rank Gaussians [15, 22], mixtures of exponential families [17, 19], and normalising flows [25]. We focus on the most popular mean-field and normalizing flow families. Mean-field families assume independence across the D dimensions: $q(\theta) = \prod_{i=1}^{D} q_i(\theta_i)$, where each q_i typically belongs to some exponential family or other simple class of distributions. Normalising flows [1] provide more flexible families that can capture correlation and non-linear dependencies. A normalizing flow is defined via the transformation of a probability density through a sequence of invertible mappings. By composing several maps, a simple distribution such as a mean-field Gaussian can be transformed into a more complex distribution [25].

f-divergences. The most commonly used divergences are examples of f-divergences [28]. For a convex function f satisfying $f(1) = 0$, the f-divergence is given by

$$D_f(p \parallel q) := \mathbb{E}_{\theta \sim q}\left[f\left(\frac{p(\theta \mid Y)}{q(\theta)}\right)\right].$$

The exclusive Kullback-Leibler (KL) divergence corresponds to $f(w) = -\log(w)$, the inclusive KL divergence corresponds to $f(w) = w \log(w)$, the χ^2 divergence corresponds to $f(w) = (w-1)^2$, and the general α-divergences correspond to $(w^\alpha - w)/\{\alpha(\alpha - 1)\}$. We also consider the *adaptive* f-divergence proposed by Wang et al. [29].

Loss estimation and stochastic optimization. In all the cases we consider, minimizing the f-divergence is equivalent to minimizing the loss function $\mathcal{L}_f(p \parallel q) := \mathbb{E}_{\theta \sim q}[f(w(\theta))]$ (although, see Wan et al. [28] for a different approach). Let $L(\lambda) := \mathcal{L}_f(p \parallel q_\lambda)$ denote the loss as a function of the variational parameters λ. The loss and its gradient $G(\lambda) := \nabla_\lambda L(\lambda)$ can both be approximated using, respectively, the Monte Carlo estimates

$$\widehat{L}(\lambda) = \tfrac{1}{S} \textstyle\sum_{s=1}^{S} f(w(\theta_s)) \quad \text{and} \quad \widehat{G}(\lambda) = \tfrac{1}{S} \textstyle\sum_{s=1}^{S} g(\theta_s), \tag{2}$$

where $\theta_1, \dots, \theta_S$ are independent draws from q_λ and $g : \mathbb{R}^K \to \mathbb{R}^K$ is an appropriate gradient-like function that depends on f and w. The two most popular gradient estimators in the literature are the score function and the reparameterization gradient estimator [20, 30]. The *score function gradient* corresponds to $g(\theta) = \{f(w(\theta)) - w(\theta)f'(w(\theta))\}\nabla_\lambda \log q_\lambda(\theta)$. It is a general-purpose estimator that applies to both discrete and continuous distributions q, but it is known to suffer from high variance. When this estimator is used for the *mass-covering* divergences such as the inclusive KL and general α-divergences with $\alpha > 1$, the importance weights are usually replaced with self-normalized importance weights $w(\theta_s)/\sum_{i=1}^{S} w(\theta_i)$. The *reparameterization gradient* [20] requires expressing the distribution q_λ as a deterministic transformation of a simpler base distribution r such that $T_\lambda(z) \sim q_\lambda$ with $z \sim r$. This allows writing an expectation with respect to q_λ as an expectation over the simpler distribution r. The reparameterization estimator corresponds to using $g(z_s) = \nabla_\lambda f(w(T_\lambda(z_s)))$ (for $z_s \sim r$) in place of $g(\theta)$, where w implicitly depends on λ as well. In the case of the adaptive f-divergence, the importance weights $w(\theta_1), \dots, w(\theta_S)$ are sorted, and the gradients corresponding to each sample are then weighed by the empirical rank. The gradient estimates can be used in a stochastic gradient optimization scheme such that

$$\lambda^{t+1} \leftarrow \lambda^t + \eta_t \widehat{G}(\lambda^t), \tag{3}$$

where η_t is the step size. In practice, more stable adaptive stochastic gradient optimisation methods such as RMSProp or Adam [9, 13], which smooth or normalize the noisy gradients, are often used.

Numerous prior work have studied some of the challenges tied to optimizing these divergence measures under the presence of noisy gradient estimates [1, 10, 29, 31]. Particularly, when dealing with mass-covering divergences, the gradient estimates can become so noisy that convergence is not possible in practice, as we will illustrate later on.

3 Assessing the Reliability of Black-box Variational Inference

3.1 Conceptual framework

How can we determine – both conceptually and experimentally – what is required to obtain reliable estimates of the variational divergence and optimal variational approximation? As we have seen, the most common variational divergences and their Monte Carlo gradient estimators can be expressed in terms of the density ratio $w(\theta)$. Reliable black-box variational inference ultimately depends on the behavior of $w(\theta)$ since (1) accurate optimization requires low-variance and (nearly) unbiased gradient estimates $\widehat{G}(\lambda)$, and (2) determining convergence and validating the quality of variational approximations can require accurate estimates $\widehat{L}(\lambda)$ of variational divergences [14, 15]. While *asymptotically* (in the number of iterations and Monte Carlo sample size S) there may be no issues with stochastic optimization or divergence estimation, in practice black-box variational inference operates in the *pre-asymptotic* regime. **Therefore, the reliability of black-box variational inference depends on the pre-asymptotic behavior of the $w(\theta)$, and how it interacts with the choice of variational objective and gradient estimator.**

Before accounting for the effects of the objective and gradient estimator, first consider the behavior of the density ratio $w(\theta)$, which can also be interpreted as an importance sampling weight with $q_\lambda(\theta)$ as the proposal distribution [cf. 2, 16, 29]. Pickands [23] proved, under commonly satisfied conditions, that for u tending to infinity, the distribution of $w(\theta) \mid w(\theta) > u$ is well-approximated by the three-parameter generalized Pareto distribution $\mathsf{GPD}(u, \sigma, k)$, which for $k > 0$ has density $p(w \mid u, \sigma, k) = \sigma^{-1}\{1 + k(w-u)/\sigma\}^{-1-1/k}$ where w is restricted to (u, ∞). Since $w(\theta) > 0$, this implies its distribution is heavily skewed to the right with a power-law tail. Consider the idealized scenario of estimating the mean of $w(\theta) \sim \mathsf{GPD}(u, \sigma, k)$. We assume the mean is finite, which is equivalent to assuming $k < 1$ since $\lfloor 1/k \rfloor$ determines the number of finite moments. Because of the heavy right skew, most of the mass of $w(\theta)$ is below its mean. Therefore, even after averaging a large number of samples, most empirical estimates $\sum_{s=1}^{S} w(\theta_s)$ will be smaller than the true mean. Figure 3a illustrates this behavior for different values of k: even with 1 million samples, the empirical mean is far below the true mean when $k > 0.7$. The highly variable sizes of the confidence intervals based on 10,000 replications further highlight the instability of the estimator. **So, even though the empirical mean is an unbiased estimator, in the pre-asymptotic regime (before the generalized central limit theorem is applicable [5]), in practice the estimates are heavily biased downward with high probability.** If $w(\theta)$ is not a generalized Pareto distribution, we can instead treat k as

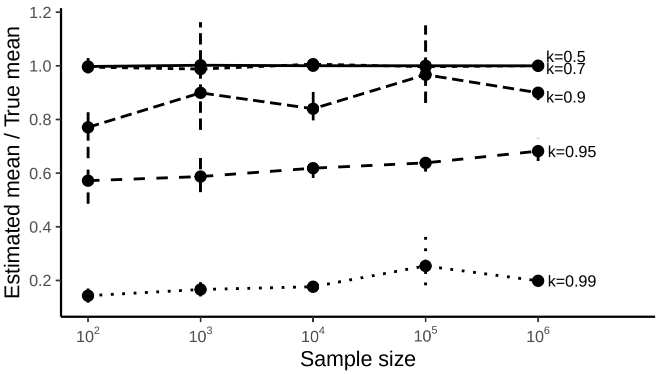


Figure 2: The ratio of estimated mean and true mean for different values of k shape parameter of a generalized Pareto distribution and confidence intervals in a finite sample size simulation.

the *tail index* $k := \inf\{\ell > 0 : \mathbb{E}_{\theta\sim q}\{w(\theta)^{1/\ell}\} < \infty\}$, which encodes the same tail behavior as $\mathsf{GPD}(u, \sigma, k)$. Crucially, we should expect k to be much larger than 0 when there is a significant mismatch between the target distribution and the variational family. **Since selecting a variational family that can match the typical set tends to be more difficult in higher dimensions, we should expect k to be larger for higher-dimensional posteriors.**

We can generalize our observations about pre-asymptotic estimation bias to the estimators $\widehat{L}(\lambda)$ and $\widehat{G}(\lambda)$. For the loss estimator, we replace $w(\theta_s)$ with $f(w(\theta_s))$, where $f(w)$ is polynomial in w and $\log w$ for the class of losses we consider. If the dominant term of $f(w)$ is of order w^α, the tail behavior will be similar to a generalized Pareto with $k_\alpha = \alpha k$. Thus, $\widehat{L}(\lambda)$ will have larger pre-asymptotic bias as α increases. For example, estimation of the mass-covering inclusive KL (where $\alpha = 1$) – and, more generally, mass-covering α-divergences with $\alpha > 0$ – will suffer from a large pre-asymptotic bias. On the other hand, for the mode-seeking exclusive KL, $f(w) = \log(w)$, so we can expect all moments to be finite and therefore a much smaller pre-asymptotic bias.

Similar considerations apply to the gradient estimator, with the details depending on the specific estimator used. However, when using self-normalized weights for α-divergences, we can expect a large pre-asymptotic bias whenever $w(\theta)$ has such bias since self-normalization involves estimating the mean of $w(\theta)$. This bias will affect the accuracy of the solution found using stochastic optimization. Thus, the quality of the solutions found can only partially be improved by using a smaller step size since smaller step sizes will only reduce the effects of a large estimator variance, but not the effects from a large bias. We provide more details on the behavior of the score function and reparameterized gradients for each of the divergences in **????**, following Geffner and Domke [11].

In summary, our framework makes two key predictions:

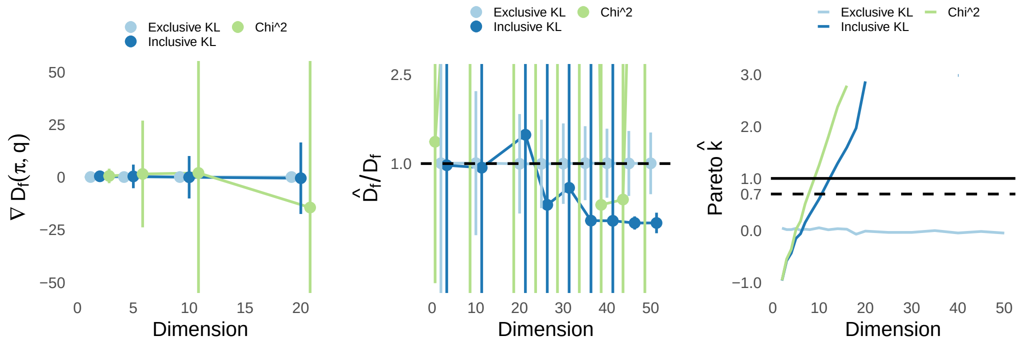
(P1) Estimates and gradients of mode-seeking divergences (in particular exclusive KL divergence with log dependence on w) have lower variance and are less biased than those of mass-covering divergences (in particular α-divergences with $\alpha > 0$, with polynomial dependence on w).

(P2) The degree of polynomial dependence on w determines how rapidly the bias and variance will increase as approximation accuracy degrades – in particular, in high dimensions.

Because the adaptive f-divergence depends directly on the (ordered) weights, we expect it to behave similarly to the mass-covering divergences.

3.2 Experimental framework

In the light of potentially large non-asymptotic bias arising from the heavy right tail of $w(\theta)$, it is important to verify the pre-asymptotic behavior of the Monte Carlo estimators used in variational inference. We follow the approach developed by Vehtari et al. [27] for importance sampling and compute an empirical estimate $\hat{k}$ of the tail index k by fitting a generalized Pareto distribution to the observed tail draws. In the importance sampling setting, Vehtari et al. [27] show that the minimal sample size to have a small error with high probability scales as $S = \mathcal{O}(\exp\{k/(1-k)^2\})$. Vehtari



(a) (b) (c)

Figure 3: Results for correlated Gaussian targets of dimension $D = 1, \dots, 50$ using either the exclusive or inclusive KL divergence as the variational objective. Each line in the plots corresponds to fitting and evaluating the same divergence measure as indicated in the legend. Each result is the average of 50 independent simulations. Quantiles are computed from simulating $100,000$ draws. **(a)** Bias and variance of the gradients of the optimised f-divergence for one parameter θ_d for increasing dimensions at the end of the optimisation for correlated Gaussian targets of dimension $D = 1, \dots, 20$ and mean field Gaussian as variational approximation. **(b)** The ratio of the f-divergence estimate to the true value. **(c)** The $\hat{k}$ values for the variational approximations.

et al. [27] also demonstrate that $\hat{k}$ provides a practical *pre-asymptotic convergence rate estimate* even when the variance is infinite and a generalized central limit theorem holds. While estimating $\hat{k}$ in general requires larger sample size than is commonly used to estimate the stochastic gradients, we can still use it to diagnose and identify the challenges with different divergences. If $\hat{k} > 0.7$, the minimal sample size to obtain a reliable Monte Carlo estimate is so large that it is usually infeasible in practice. This cutoff is in agreement with our findings shown in Fig. 3a. Thus, together with our conceptual framework, we have a third key prediction:

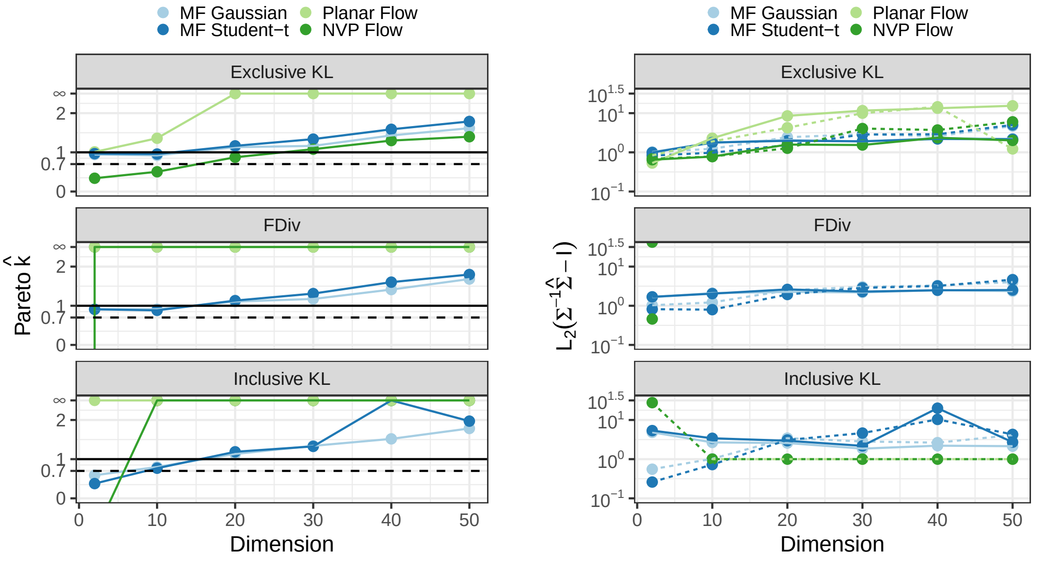
(P3) The $\hat{k}$ value can be used to diagnose pre-asymptotic reliability of variational objectives. In particular, the α-divergence with $\alpha > 0$ will become unreliable when $\max(1, \alpha) \times \hat{k} > 0.7$, even if w is bounded (by a very large constant).

3.3 Verification of Pre-asymptotic (Un)reliability

We first verify our three key predictions in a simple setting where we can compute most of the relevant quantities such as the loss function in closed form. Specifically, we fit a mean-field Gaussian to a Gaussian with constant 0.5 correlation factor using the inclusive KL, exclusive KL, χ^2, and $1/2$-divergences. We vary the dimensionality D from 1 to 50, which is a surrogate for the degree of mismatch between the optimal variational approximation and the target distribution. To find the optimal divergence-based approximation, we optimize the closed-form expression for the divergences between two Gaussians. Hence, we can consider on the *best-case scenario* and ignore the complexities and uncertainty due to the stochastic optimization. Due to space limitations, we focus on representative cases of the approximations from optimising the mode-seeking exclusive KL divergence and the mass-covering inclusive KL divergence. Results for the other divergences are included in the appendix.

(P1) Mode-seeking divergences are more stable and reliable than mass-covering ones. Figure 3b shows that as the approximation–target mismatch increases with dimension, the bias in and variance of the divergence estimates increases substantially for the inclusive KL and χ^2 but only moderately for the exclusive KL. Similarly, Fig. 2 shows that gradient bias and variance increases with dimension for inclusive KL and χ^2 but not exclusive KL.

(P2) Degree of polynomial dependence on w determines sensitivity to approximation–target mismatch. Figure 3b shows that divergence estimates resulting from optimising higher polynomials of w become more and more unstable as dimensions increases.



(a)

(b)

Figure 4: Results for increasing dimensions of the robust regression model. **(a)** Pareto $\hat{k}$ values for BBVI approximations. **(b)** Relative error of covariance estimates for BBVI (solid lines) and after PSIS correction (dashed lines).

(P3) $\hat{k}$ **diagnoses pre-asymptotic reliability.** Figure 3c shows that the $\hat{k}$ values grow rapidly for the inclusive KL-based approximation, particularly for higher-degree dependence on w, which agrees with predicted behavior and large bias and variance of the inclusive KL and χ^2. In contrast, the $\hat{k}$ values remain fairly stable for the exclusive KL-based approximation, again in agreement with predicted and observed bias and variance behavior.

4 Experiments

In this section, we describe a series of experiments to study how our pre-asymptotic framework can be used for assessing the reliability of black-box variational approximations for practical applications and developing best-practices. For all posteriors, we fit mean-field Gaussian and Student-t families, a planar flow [25] with 6 layers and a non-volume preserving (NVP) flow [8] with 6 stacked neural networks with 2 hidden layers of 10 neurons each for both the translation and scaling operations with a standard Gaussian distribution for the latent variables. We use Stan [26] for model construction. For stochastic optimization we use RMSProp with initial step size of 10^{-3} run for either $T_{\max}$ iterations or until convergence was detected using a modified version of the algorithm by Dhaka et al. [6]. For the exclusive KL we use 10 draws for gradient estimation per iteration, while for the other divergences we use 200 draws, and a warm start at the solution of the exclusive KL. In practice, we found the optimisation for χ^2 divergence extremely challenging, with the solution failing to converge even for moderate dimensions $D \approx 10$. Therefore, we only include results for the KL divergences and the adaptive f-divergence. We compare the accuracy of approximated posterior moments to ground-truth computed either analytically or using the dynamic Hamiltonian Monte Carlo algorithm in Stan [26]. Specifically, we consider the estimates $\hat{\mu}$ and $\hat{\Sigma}$ for, respectively, the posterior mean μ and covariance matrix Σ. We also consider the mean and covariance estimates produced by PSIS and compute $\hat{k}$. The experiments were carried on a laptop and an internal cluster with only CPU capability. The code for the experiments will be made available after acceptance using MIT license.

4.1 Heavy-tailed posteriors

First, we study the toy robust regression model previously used by Huggins et al. [14] given by

$$\beta_d \sim \mathrm{N}(0, 10), \qquad\qquad y_n \mid x_n, \beta \sim t_{10}(\beta^\top x_n, 1),$$

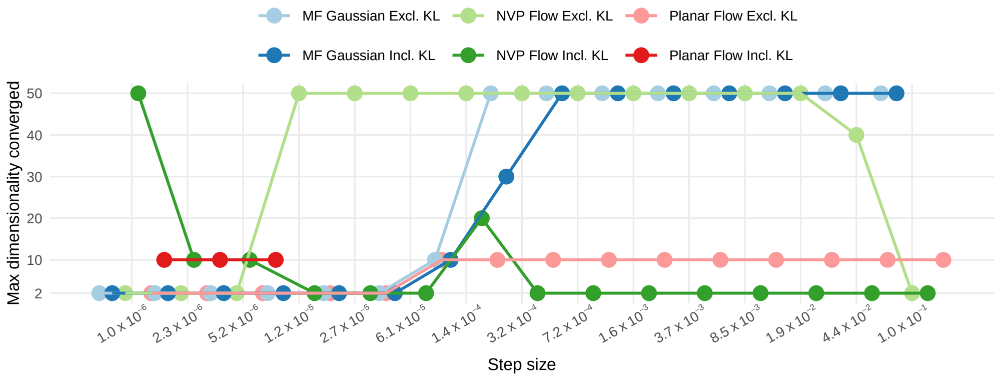


Figure 5: Maximum dimensionality converged per step size for the robust regression model.

where $y_n \in \mathbb{R}, x_n \in \mathbb{R}^D$ are the target and predictors respectively, β denotes the unknown coefficients, and D is varied from 2 to 50. We generated data from the same model with covariates generated from a zero-mean Gaussian with constant correlation of 0.4. The Student's t leads to the posterior having heavy tails, making it a more challenging target distribution. We use $T_{\max} = 10{,}000$.

Mode-seeking divergences are easier to optimize. Figure 4a shows that the estimated tail index $\hat{k}$ generally increases with the dimension as expected. In particular, the $\hat{k}$ values when using normalizing flows, which are more challenging to optimize, is low for $D < 20$ when using exclusive KL, but infinite when using either the inclusive KL or f-divergence. From Fig. 4b we can see that exclusive KL provides also more accurate and reliable posterior approximations than the inclusive KL and adaptive f-divergence, particularly for the normalizing flows. This observation is consistent with the prediction (P3) of the proposed framework. The better performance for normalizing flows corroborates the relative ease of stochastic optimization with the exclusive KL divergence compared to the inclusive KL or the adaptive f-divergence – despite the fact that we used 20 times as many Monte Carlo samples to estimate the gradients for the inclusive KL and the f-divergence compared to the exclusive KL. To further illustrate the relative difficulty of optimizing the inclusive KL divergence, Fig. 5 shows the largest dimension for which the stochastic optimization converged as a function of the step-size. For most step-sizes, the combination of normalizing flows and the inclusive KL divergence only converged for $D = 2$, whereas convergence is possible in higher dimensions for simpler variational families. These observations are consistent with predictions (P1)-(P2) of the proposed framework.

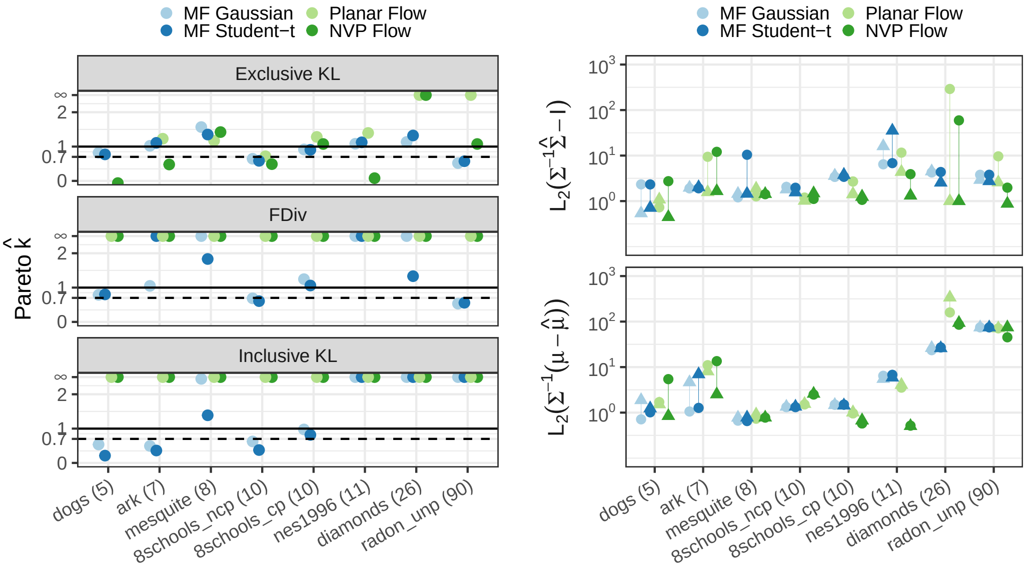
Adaptive f-divergence interpolates between the exclusive and inclusive KL divergence, but is difficult to optimize. In low dimensions, the adaptive f-divergence behaves somewhere between the two KL divergences as seen in **??** – as it was designed to [29]. As confirmed by Fig. 4, For higher-dimensional posteriors, we expect it to behave more like the exclusive KL, but it is less stable due to its functional dependence on the importance weights.

Normalizing flows can be effective but are challenging to optimize. Fig. 4 also shows that normalizing flows can be quite effective when used with exclusive KL to ensure stable optimization. However, as can be seen in **??**, when using out-of-the-box optimization with no problem-specific tuning (as we have done for a fair comparison), the normalizing flows approximations can have pathological features – even in low dimensions.

4.2 Realistic models and datasets

We now study how the choice of divergence and approximating family compare across a diverse range of benchmark posteriors. We compare variational approximations for models and datasets from `posteriordb`[*] in terms of accuracy of the estimated moments and predictive likelihood. We used an 80/20 training/test split on all datasets to compute the predictive likelihoods. We use $T_{\max} = 15{,}000$.

[*] `https://github.com/stan-dev/posteriordb`



(a)

(b)

Figure 6: Results for `posteriordb` experiments. Dimensionality of each dataset is given in parentheses. **(a)** Pareto $\hat{k}$ values for BBVI approximations. **(b)** Relative error of mean and covariance estimates for BBVI using exclusive KL (circles) and after PSIS correction (triangles).

Table 1: Predictive likelihood results on `posteriordb` datasets using a Mean Field Gaussian approximation. The results denote the likelihood with the variational approximation solution obtained and after PSIS correction to the solution. **Bold** (underline) indicates best-performing method(s) (variational method(s))

Name	HMC	Excl. KL	Excl. KL+PSIS	Incl. KL	Incl. KL+PSIS
dogs	-71.1±1.2	-71.2 ±1.3	-71.7±1.5	-110±3.5	**-70.5**±4.1
arK	**-32.4**±0.6	**-34.3**±0.7	**-34.4**±0.7	-35.2±0.8	-34.9±0.8
mesquite	**-1681**±127	-2512±140	-5418±186	-∞	-∞
nes1996	**-412.9**±1.7	**-412.8**±1.7	-427.9±1.8	-2140.5±59.7	-499.3±45.6
diamonds	**22.1**±3.1	-2.6±1.3	1.5±1.2	-3196.6±57.9	-3149±55.7
radon	**-234.4**±1.1	-353.0±19.3	-325.0±19.5	-377.4±1.9	-370.5±2.2

Exclusive KL remains the most reliable for realistic posteriors. The results are summarized in Fig. 6, where the same pattern is seen: the exclusive KL is superior for higher-dimensional posteriors (e.g., $D > 10$) or when combined with normalizing flows, while inclusive KL is better for lower-dimensional posteriors. Despite the superior performance of the exclusive KL divergence, the large values for $\hat{k}$ indicate that fitting approximations based on normalizing flows remains a challenge in high dimensions. The performance for the adaptive f-divergence is comparable to the inclusive KL divergence. Table 1 shows that the exclusive KL divergence consistently outperforms the inclusive KL divergence in terms of predictive accuracy, but can be significantly worse than HMC.

Importance sampling can substantially improve accuracy. Focusing on exclusive KL, Fig. 6b shows the relative errors of the first two moments for the variational approximation (dots) and after correcting the estimates using PSIS (triangles). In some cases, the PSIS correction dramatically improved the accuracy of the normalizing flows.

Reparameterization is an important tool for improving accuracy. The 8-schools model is low-dimensional ($D = 10$), but the funnel-shaped posterior makes inference challenging for variational approximations [14, 31]. As has been noted previously in the literature, and is clear from Figs. 6a and 6b, reparameterizing the model so that the posterior better matches the variational family can be an effective way to improve the accuracy of the approximation. See **??** for an illustration.

5 Discussion

Our conceptual framework based on the pre-asymptotic behavior of the density ratios / importance weights w along with our comprehensive experiments lead to a number of important takeaways for practitioners looking to obtain reasonably accurate posterior approximations using black-box variational inference:

- The instability of mass-covering divergences like inclusive KL and χ^2 means that, given currently available methodology, users are better off using the exclusive KL divergence except for easy low-dimensional posteriors. The reliance of the adaptive f-divergence on importance weights leads to similar instability.
- Importance sampling appears to almost always be beneficial for improving accuracy, even when the $\hat{k}$ diagnostic is large. However, a large $\hat{k}$ does suggest the user should not expect even the PSIS-corrected estimates to be particularly accurate.
- Using normalizing flows – particularly NVP flows – together with exclusive KL and PSIS provides the best and most consistent performance across posteriors of varying dimensionality and difficulty. We therefore suggest this combination as a good default choice.

Our results suggest an important direction for future work is improving the stability of optimization with normalizing flows, which still tend to have some pathological behaviors unless they are very carefully tuned since such tuning significantly detracts from the benefits of using BBVI.

6 Limitations

While our experiments included a range of common statistical model types, our findings may not generalize to all types of posteriors or to other variational families. For example, we did not explore semi-implicit methods or applications to neural networks. We also did not investigate alternative divergences such as those used in importance-weighted autoencoders.

Acknowledgments

We thank Sihan Liu for pointing out and fixing errors in the code. We thank Ari Hartikainen and PyStan developers for help with posteriordb, Academy of Finland (grants 298742, and 313122) and Finnish Center for Artificial Intelligence for partial support of the research. We also acknowledge the computational resources provided by the Aalto Science-IT project. AKD would like to thank Tomas Geffner and Eero Siivola for useful discussions.

References

[1] Abhinav Agrawal, Daniel R. Sheldon, and Justin Domke. Advances in black-box VI: normalizing flows, importance weighting, and optimization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[2] Robert Bamler, Cheng Zhang, Manfred Opper, and Stephan Mandt. Perturbative black box variational inference. In *Advances in Neural Information Processing Systems*, volume 30, pages 5079–5088, 2017.

[3] D. M. Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[4] Jörg Bornschein and Yoshua Bengio. Reweighted wake-sleep. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[5] Louis H Y Chen and Qi-Man Shao. Normal approximation under local dependence. *The Annals of Probability*, 32(3):1985–2028, 2004.

[6] Akash Kumar Dhaka, Alejandro Catalina, Michael R Andersen, Måns Magnusson, Jonathan Huggins, and Aki Vehtari. Robust, accurate stochastic optimization for variational inference. In *Advances in Neural Information Processing Systems*, volume 33, pages 10961–10973, 2020.

[7] Adji Bousso Dieng, Dustin Tran, Rajesh Ranganath, John Paisley, and David Blei. Variational inference via \chi upper bound minimization. In *Advances in Neural Information Processing Systems 30*, pages 2732–2741. 2017.

[8] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

[9] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[10] Tomas Geffner and Justin Domke. On the difficulty of unbiased alpha divergence minimization, 2020.

[11] Tomas Geffner and Justin Domke. Empirical evaluation of biased methods for alpha divergence minimization. In *Symposium on Advances in Approximate Bayesian Inference, AABI 2020*, 2020.

[12] Jose Hernandez-Lobato, Yingzhen Li, Mark Rowland, Thang Bui, Daniel Hernandez-Lobato, and Richard Turner. Black-box alpha divergence minimization. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1511–1520. PMLR, 2016.

[13] G. E. Hinton and Tijmen Tieleman. Lecture 6.5 – Rmsprop: Divide the gradient by a running average of its recent magnitude. In *Coursera: Neural networks for machine learning*, 2012.

[14] Jonathan H Huggins, Mikolaj Kasprzak, Trevor Campbell, and T. Broderick. Validated Variational Inference via Practical Posterior Error Bounds. In *AISTATS*, 2020.

[15] Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, and D. M. Blei. Automatic Variational Inference in Stan. In *Advances in Neural Information Processing Systems*, Advances in Neural Information Processing Systems, 2015.

[16] Yingzhen Li and Richard E Turner. Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, volume 29, pages 1073–1081, 2016.

[17] Wu Lin, Mohammad Emtiyaz Khan, and Mark Schmidt. Fast and Simple Natural-Gradient Variational Inference with Mixture of Exponential-family Approximations. In *International Conference on Machine Learning*, 2019.

[18] David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[19] Andrew C. Miller, Nicholas J. Foti, and Ryan P. Adams. Variational boosting: Iteratively refining posterior approximations. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2420–2429. PMLR, 2017.

[20] Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo gradient estimation in machine learning. *arXiv preprint arXiv:1906.10652*, 2019.

[21] Christian A. Naesseth, Fredrik Lindsten, and David M. Blei. Markovian score climbing: Variational inference with KL(p||q). *CoRR*, abs/2003.10374, 2020.

[22] Victor M H Ong, David J Nott, and Michael S Smith. Gaussian Variational Approximation With a Factor Covariance Structure. *Journal of Computational and Graphical Statistics*, 27(3): 465–478, 2018. doi: 10.1080/10618600.2017.1390472.

[23] James Pickands. Statistical inference using extreme order statistics. *Annals of Statistics*, 3: 119–131, 1975.

[24] Dennis Prangle. Distilling importance sampling, 2021.

[25] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538. PMLR, 2015.

[26] Stan Development Team. Stan modeling language users guide and reference manual. 2.26, 2020. URL `https://mc-stan.org`.

[27] Aki Vehtari, Daniel Simpson, Andrew Gelman, Yao Yuling, and Jonah Gabry. Pareto smoothed importance sampling. *arXiv preprint arXiv:1507.02646*, 2019.

[28] Neng Wan, Dapeng Li, and Naira Hovakimyan. f-Divergence Variational Inference. In *Advances in Neural Information Processing Systems*, 2020.

[29] Dilin Wang, Hao Liu, and Qiang Liu. Variational inference with tail-adaptive f-divergence. In *Advances in Neural Information Processing Systems*, volume 31, pages 5737–5747, 2018.

[30] Ming Xu, Matias Quiroz, Robert Kohn, and Scott A. Sisson. Variance reduction properties of the reparameterization trick. In *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 2711–2720. PMLR, 2019.

[31] Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Yes, but did it work?: Evaluating variational inference. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5581–5590. PMLR, 2018.