# Topics in Approximate Inference

Yingzhen Li

current version: Nov 2017

**Abstract**

This monograph provides a brief introduction on approximate inference and goes in depth for a selected list of related topics. Many of the presented materials are adapted from my published papers, my research notes during PhD, and my (draft) PhD thesis. I expect to keep updating this list of topics to include new techniques.

## Contents

# 0 Front Matter

Approximate inference is key to modern probabilistic modelling, and since the start of my PhD there has been considerable progress on this subject. The literature becomes very diverse so that a new comer to the subject might find it difficult to learn the key techniques and identify important papers. Therefore in this document I share my reading and research notes on approximate inference, and I hope this would help people understand the general idea of this subject.

## 0.1 How to use this document

I assume you (as the reader) know some basic concepts in probability and machine learning. If you are completely new to approximate inference, then I would encourage you to start from § 0.3.

The notes are organised into three categories:

- algorithms for fitting approximations;
- architecture designs of the approximation;
- applications to machine learning tasks.

I should note that the topics included in the list are not "mutually independent", for example, a specific design of the approximate distribution might require special care of the fitting algorithm.

We will build from basics to advances for each topic in each category, so if you find sections x.1 and x.2 very confusing then please contact me. At the end of each topic section I will include a short list of (what I think are) "must read" papers, and all the citations can be found in the reference list.[1] If you want to suggest papers you are also welcome to contact me. Comments and extra examples are included as "remark" paragraphs: they often contain advanced stuff so it can be skipped if appropriate.

## 0.2 Math preparation

In this section we establish the mathematical notations and concepts that will be repeatedly used in the rest of the note.

- Observation variables $\boldsymbol{x}$, and in supervised learning case we also use $\mathbf{y}$ as the labels.
- Latent variable $\boldsymbol{z}$, it is unobserved and needs to be integrated out.
- Model parameter $\boldsymbol{\theta}$: for example it can be the set of neural network weights. Bayesian methods also consider it as a random variable.

[1] I am not an expert of sampling-based methods so I usually cite review papers and books for them, and I definitely have missed some important papers.

- Variational parameter $\boldsymbol{\phi}$: the parameters associate to the approximation.

- Probability density function:
  Denote the measurable space as $(\Theta, \Sigma)$, where $\Theta$ is the sample space of the random variable $\boldsymbol{\theta}$ of interest, and $\Sigma$ is a pre-defined $\sigma$-algebra on $\Theta$. A *probability distribution* $P$ is a measure defined on $\Sigma$ such that $P(\Theta) = 1$. Also we assume there exists a dominating measure (also called reference measure) $\mu$ on $\Sigma$ such that, for any probability distribution $P$ defined on $\Sigma$, we can define its *probability density function* $p$ by $dP = pd\mu$.[2] For simplicity in the rest of the note we will work with the sample space $\Theta = \mathbb{R}^D$, the $\sigma$-algebra $\Sigma = \{S : S \subset \mathbb{R}^D\}$, and the dominating measure $d\mu = d\boldsymbol{\theta}$. Finally we write $\mathcal{P}$ the space of PDFs such that any probability distribution $P$ defined on $\Sigma$ has its PDF $p \in \mathcal{P}$.

- Divergence:
  Given a set of probability density functions $\mathcal{P}$ for a random variable $\boldsymbol{\theta}$, a divergence on $\mathcal{P}$ is defined as a function $\mathrm{D}[\cdot||\cdot] : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ such that $\mathcal{D}[p||q] \geq 0$ for all $p, q \in \mathcal{P}$, and $\mathcal{D}[p||q] = 0$ iff. $p = q$.

  The definition of divergence is much weaker than that for a *distance* such as the $l_2$-norm, since it does not need to satisfy either symmetry in arguments or the triangle inequality. Hence there exist many available divergences to use, where some of them are heavily used in approximate inference.

## 0.3 Inference, integration and optimisation

Probabilistic modelling starts by defining a distribution of data. For instance, in discriminative supervised learning, one would define a conditional distribution $p(\mathbf{y}|\boldsymbol{x}, \boldsymbol{\theta})$, which is also called the *likelihood function* of $\boldsymbol{\theta}$. A concrete example for this would interpret $p(\mathbf{y}|\boldsymbol{x}, \boldsymbol{\theta})$ as outputting the probability of a configuration of $\mathbf{y}$ (e.g. a label or a real value) by transforming the input $\boldsymbol{x}$ (an image, a sentence, etc.) through a neural network parameterised by $\boldsymbol{\theta}$. Before observing any real-world data, the parameters $\boldsymbol{\theta}$ are unknown, but we have a prior belief $p_0(\boldsymbol{\theta})$ about what value they might take, e.g. they should have small $\ell_2$ norm if using a Gaussian prior centred at zero. Then we receive the observations $\mathcal{D} = \{(\boldsymbol{x}_n, \mathbf{y}_n)\}_{n=1}^N$, and based on data we want to answer questions on the unknown parameters $\boldsymbol{\theta}$, for example: given $\mathcal{D}$, what is the most probable value of $\boldsymbol{\theta}$, and how likely is $\boldsymbol{\theta}$ to be set to a given value? Answering these questions is precisely the procedure of *inference*: a procedure of deducing unknown properties (in our example the neural network weights) given the observed, or known information.

### 0.3.1 Exact Bayesian inference as integration

Bayesian statisticians are particularly interested in answering the latter question, by computing the *posterior distribution*, or the *posterior belief* of $\boldsymbol{\theta}$ given $\mathcal{D}$, using

[2] We can also define divergences without assuming a common reference measure, which is out of the scope of this note. In this case one should work with equalities up to zero measure.

Bayes' rule [Bayes and Price, 1763, Laplace, 1820]:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})}{p(\mathcal{D})}, \tag{1}$$

with $p(\mathcal{D}|\boldsymbol{\theta}) = \prod_n p(\mathbf{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta})$ following the i.i.d. assumption. The elegance of Bayes' rule is that *it separates inference from modelling*. The model – the prior distribution and the likelihood – completely determines the posterior distribution, and the only thing left is to *compute* the inference.

A closer look at Bayes' rule reveals that the core computation of Bayesian inference is *integration*. Using the sum rule of probability distributions we have the marginal distribution computed as[3]

$$p(\mathcal{D}) = \int p(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})d\boldsymbol{\theta},$$

and if this integral is tractable, then the posterior distribution can be easily computed by (1). Moreover, to predict the label $\mathbf{y}^*$ on unseen datum $\boldsymbol{x}^*$ a Bayesian would compute the *predictive* distribution

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}) = \int p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}, \tag{2}$$

which again requires solving an integration problem. Even more, since it is hard to visualise the posterior distribution in high dimensions, one would instead look at the statistics of the posterior, for example

$$\text{posterior mean } \mu = \int \boldsymbol{\theta}p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}, \quad \text{posterior variance } \Sigma = \int (\boldsymbol{\theta}-\mu)(\boldsymbol{\theta}-\mu)^{\mathrm{T}}p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta},$$

both are integration tasks as well. In summary, many tasks in Bayesian computation can be framed as computing an integral of some function $F(\boldsymbol{\theta})$ against the posterior distribution:

$$\int F(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}, \tag{3}$$

and the goal of this document is to discuss how to perform this integration *pragmatically and efficiently*.

### 0.3.2 Approximate Bayesian inference as optimisation

Having an integration task at hand, the first action I would take is to check my college calculus book with the hope of finding an analytical solution. Unfortunately, for a vast number of integrands and distributions, the integral (3) does not exhibit an analytical form (or at least people have yet to discover it). This is particularly the case for neural networks: except for some limited special cases,[4]

[3] In discrete variable case the integral is calculated w.r.t. discrete measure, i.e. summation, which will also be referred as integration in the rest of the manuscript.

[4] e.g. the prior is Gaussian and the neural network only has one hidden layer with ReLU activation.

in general the marginal probability is intractable, let alone the posterior and the predictive distribution.

Instead of finding tractable forms of the integral, many mathematicians have their research careers dedicated to an alternative method: *numerical integration*. Because in a continuous space one could never compute $F(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})$ at *all* locations then sum them up, instead methods such as discretisation and Monte Carlo are employed. The Monte Carlo idea is particularly interesting in our context: since the integral is computed against a probability distribution, a naive approach would first sample from the posterior $\boldsymbol{\theta}_k \sim p(\boldsymbol{\theta}_k|\mathcal{D})$ then calculate the integral as

$$\int F(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D}) \approx \frac{1}{K}\sum_{k=1}^{K} F(\boldsymbol{\theta}_k). \tag{4}$$

However this simple Monte Carlo approach assumes that the posterior distribution is easy to draw samples from, which is again intractable in most scenarios. Statisticians have applied advanced sampling schemes to (approximately) draw samples from the posterior, including importance sampling, rejection sampling and Markov chain Monte Carlo [Gelman et al., 2014]. Unfortunately, in high dimensions these methods require a considerable number of samples, and the simulation time for MCMC can be prohibitively long.

Now comes the brilliant idea of (optimisation-based, or indirect) *approximate inference*: can we find another distribution $q(\boldsymbol{\theta})$ that makes the integral $\int F(\boldsymbol{\theta})q(\boldsymbol{\theta})d\boldsymbol{\theta}$ comparably easier, and at the same time has minimal *approximation error* to the exact integral we want? Concretely, using the knowledge of the functional form $F$ one can come up with a class of candidate distributions $\mathcal{Q}$, in which integrating $F$ w.r.t. any $q \in \mathcal{Q}$ has analytical form or can be evaluated quickly with numerical methods. Then the only task here is to obtain the *optimal* $q$ distribution in $\mathcal{Q}$ such that the $q$ integral is the most accurate approximation to the exact one. So in short, *approximate inference* converts the integration problem of (Bayesian) inference into an *optimisation* task. For example, an indirect[5] approach for fitting the $q$ distribution would minimise a distance/divergence/discrepancy measure from the approximation to the exact posterior

$$q^*(\boldsymbol{\theta}) = \arg\min_{q\in\mathcal{Q}} \mathrm{D}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D})]. \tag{5}$$

Note here the measure $\mathrm{D}[\cdot||\cdot]$ might not be symmetric. A popular choice for the divergence measure is the *Kullback-Leibler divergence* [Kullback and Leibler, 1951, Kullback, 1959] which leads to the widely used *variational inference* algorithm [Jordan et al., 1999, Beal, 2003]. In general an optimisation objective function $\mathcal{F}$ is designed to allow an accurate approximation to be obtained:

$$q^*(\boldsymbol{\theta}) = \arg\min_{q\in\mathcal{Q}} \mathcal{F}(q(\boldsymbol{\theta}); p(\boldsymbol{\theta}|\mathcal{D})), \tag{6}$$

which might not reflect a specific choice of divergence/discrepancy. Often this objective function $\mathcal{F}$ is crafted such that at the optimum, $\mathcal{F}^*$ can serve as an accurate

[5] a direct method would consider minimising error$(\mathbb{E}_q[F], \mathbb{E}_p[F])$, however that involves the exact integral and is mostly intractable.

approximation to the (log) marginal distribution, or *model evidence* $\log p(\mathcal{D})$ as well. A prevalent approach in this category considers the *Bethe free energy* [Bethe, 1935] that was first studied in statistical physics, which has also been shown as the underlying objective of another popular approach called *belief propagation* [Pearl, 1982]. All these methods are thoroughly discussed in later sections, and once $q$ is obtained, at prediction time the Bayesian predictive distribution (2) is approximated by

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}) \approx \int p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta}) q(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{7}$$

**Remark** (a comparison to Sampling methods)**.** Many Bayesian statisticians prefer sampling methods – and in fact it is the emergence of sampling methods such as importance sampling (IS), sequential Monte Carlo (SMC) [Doucet et al., 2001] and Markov chain Monte Carlo (MCMC) that contributes to the rapid development of Bayesian statistics. They have very nice theoretical guarantees, for example, IS and SMC provide unbiased estimates of the integral and are asymptotically exact when the number of samples $K \to +\infty$. MCMC has similar asymptotic exactness guarantee but it also requires the number of transitions $T \to +\infty$. However, I view all these sampling methods as approximate inference algorithms, simply due to the fact that in practice one can never obtain an infinite number of samples, nor simulating the MCMC dynamics for an infinite amount of time. The "effective $q$ distribution" in use is $q(\boldsymbol{\theta}) = \frac{1}{K}\sum_{k=1}^K \delta(\boldsymbol{\theta} = \boldsymbol{\theta}^k)$ in MC(MC) and $q(\boldsymbol{\theta}) = \sum_{k=1}^K \hat{w}_k \delta(\boldsymbol{\theta} = \boldsymbol{\theta}^k)$ in (re-sampled) IS/SMC. These methods are very carefully designed to achieve guarantees of unbiasedness and consistency.

**Remark** (a comparison to Bayesian quadrature)**.** Another important technique for approximating integrals is Bayesian quadrature [O'Hagan, 1991, Kennedy and O'Hagan, 1996, Ghahramani and Rasmussen, 2003], which has attracted a lot of attention as well and has been expanded to form part of an emerging research field called probabilistic numerics.[a] Here we note that, Bayesian quadrature and the approximate inference methods discussed above, address different intractability issues in integration tasks. Typically, Bayesian quadrature assumes the analytical form of the function $F$ is unknown or very expensive to evaluate, and builds a probabilistic model (e.g. Gaussian process) for $F$ given samples from the target distribution $p$. Approximate inference, on the other hand, constructs approximate distributions to the intractable distribution $p$, and considers tractable functions $F$ instead. In short, both approaches can be categorised as *model-based approximate integration*, with the only difference that they fit approximations to different components of the integrand. Readers are also referred to e.g. approximate Bayesian computation [Beaumont et al., 2002] for those integrands without tractable $F$ and $p$, and in this document we only discuss approximate inference methods and assume $F$ is analytic and cheap to compute for a given configuration.

[a] http://www.probabilistic-numerics.org/

# Part I
# Algorithms for fitting approximate distributions

## 1 Variational inference

Many approximate inference algorithms measure the approximation quality by considering the "closeness" between the target and the approximation. Then an approximate distribution can be obtained by minimising the selected "closeness" measure, and for *variational inference* (VI) this concept of "closeness" is established as the *Kullback-Leibler divergence*.

### 1.1 Kullback-Leibler (KL) divergence

*Kullback-Leibler divergence* [Kullback and Leibler, 1951, Kullback, 1959], or *KL divergence*, is arguably one of the most widely used divergence measures, not only in approximate inference but also in machine learning, statistics, and information theory.

**Definition 1.** *(Kullback-Leibler Divergence) The Kullback-Leibler (KL) divergence on $\mathcal{P}$ is defined as a function* $\mathrm{KL}[\cdot||\cdot] : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ *with the following form*

$$\mathrm{KL}[p||q] = \int p(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta}, \quad p, q \in \mathcal{P}, \tag{8}$$

*where* log *is the natural logarithm (to base e).*

One can easily check that indeed the above definition is a valid divergence. By Jensen's inequality[6] we have (8) always non-negative, and it reaches zero iff. $p = q$. Also it is clear that the KL divergence is asymmetric, i.e. $\mathrm{KL}[p||q] \neq \mathrm{KL}[q||p]$. Historically, especially when used in approximate inference context, these two cases have been referred as the *inclusive* KL divergence for $\mathrm{KL}[p||q]$, and the *exclusive* KL divergence for $\mathrm{KL}[q||p]$. These names originate from the observation that fitting $q$ to $p$ by minimising these two KL divergences returns results of different behaviour, detailed as follows:

- Fitting $q$ to $p$ by minimising $\mathrm{KL}[q||p]$:
  This KL divergence would emphasise assignment of *low* probability mass of $q$ to the location where $p$ is very small, thus the name "exclusive" KL. Consider a region $S \in \Theta$ that has $q(\boldsymbol{\theta}) > 0$ but $p(\boldsymbol{\theta}) = 0$ for $\boldsymbol{\theta} \in S$, then this would make the integrand in (8) infinity, thus the KL divergence assigns an extremely high cost to $q$ here. On the other hand, if $p(\boldsymbol{\theta}) > 0$ but $q(\boldsymbol{\theta}) = 0$, then the integrand restricted to the subset $S$ is zero, meaning that the cost

[6] Jensen's inequality: for any convex function $f$ and distribution $p$, $\mathbb{E}_p[f(x)] \geq f(\mathbb{E}_p[x])$.

for missing a region with positive $p$ mass is much lower. We also refer this property as "zero-forcing", or "mode-seeking" when $q$ is restricted to be uni-modal.

- Fitting $q$ to $p$ by minimising $\mathrm{KL}[p||q]$:
  Conversely, this KL divergence would emphasise assignment of *high* probability mass of $q$ to the location where $p$ has positive mass, thus the name "inclusive" KL. Consider the case that $q(\boldsymbol{\theta}) > 0$ but $p(\boldsymbol{\theta}) = 0$, then this would make the integrand in (8) zero. In contrast, if $p(\boldsymbol{\theta}) > 0$ but $q(\boldsymbol{\theta}) = 0$, then the integrand is infinity, meaning that the cost for missing a region with positive $p$ mass is extremely high. We also refer this property as "mass-covering".

**Remark** (maximum likelihood estimation and KL divergences)**.** We will show that maximum likelihood estimation (MLE) is equivalent to minimising a KL divergence. For a given dataset $\mathcal{D} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$, define the empirical distribution as $\hat{p}_{\mathcal{D}}(\boldsymbol{x}) = \frac{1}{N}\sum_{n=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{x}_n)$ where $\delta(\cdot)$ denotes the Dirac delta function. Then we want to fit the data with a parametric probabilistic model $p(\boldsymbol{x}|\boldsymbol{\theta})$ using MLE:

$$\hat{\boldsymbol{\theta}}^{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\boldsymbol{\theta}). \tag{9}$$

Simple calculation reveals that maximising the log-likelihood of $\boldsymbol{\theta}$ is equivalent to minimising the KL divergence

$$\hat{\boldsymbol{\theta}}^{\mathrm{ML}} = \arg\min_{\boldsymbol{\theta}\in\Theta} \mathrm{KL}[\hat{p}_{\mathcal{D}}(\boldsymbol{x})||p(\boldsymbol{x}|\boldsymbol{\theta})] = \arg\min_{\boldsymbol{\theta}\in\Theta} -\frac{1}{N}\sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\boldsymbol{\theta}) + \text{const.}$$

MLE is widely used in all types of machine learning tasks, e.g. learning generative models. We will come back to this topic in later sections.

## 1.2 Variational free-energy

Unfortunately, direct divergence minimisation is still intractable, since that involves evaluating the target distribution itself. For example, consider minimising the exclusive KL divergence $\mathrm{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D})]$ to obtain the approximate posterior. But we still need to compute $p(\boldsymbol{\theta}|\mathcal{D})$, and in particular the marginal likelihood $p(\mathcal{D})$ which is intractable. In this section we discuss variational inference (VI) – a widely used approximate inference algorithm – which incorporates divergence minimisation in a smart way. To emphasise that the algorithm is applicable to more general cases beyond posterior approximation, we now write the *target distribution* as

$$p(\boldsymbol{\theta}) = \frac{1}{Z}p^*(\boldsymbol{\theta}),$$

where $p^*(\theta)$ is the *unnormalised* target distribution and $Z = \int p^*(\boldsymbol{\theta})d\boldsymbol{\theta}$ is the *normalising constant* or *partition function*. In the posterior approximation context

$p^*(\boldsymbol{\theta}) = p(\boldsymbol{\theta}, \mathcal{D})$ and $Z = p(\mathcal{D})$.

As already discussed, the exclusive KL divergence minimisation problem is intractable. Fortunately the minimiser of the exclusive KL can also be obtained by an equivalent minimisation problem of the so called *variational free-energy* (VFE):

$$\begin{aligned} \min_q \mathcal{F}_{\text{VFE}}(q; p) &:= \min_q \text{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta})] - \log Z \\ &= \min_q \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p^*(\boldsymbol{\theta})} d\boldsymbol{\theta}. \end{aligned} \tag{10}$$

This is because the normalising constant $Z$ is independent with the approximation $q$, thus can be dropped in the exclusive KL. Historically the negative of the variational free-energy is also frequently discussed, which is named *variational lower-bound* or *evidence lower-bound (ELBO)* in the context of posterior approximation

$$\mathcal{L}_{\text{VI}}(q; p) := -\mathcal{F}_{\text{VFE}}(q; p) = \int q(\boldsymbol{\theta}) \log \frac{p^*(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta}. \tag{11}$$

The lower-bound property comes from the fact that $\log Z \geq \mathcal{L}_{\text{VI}}(q; p)$, because of the non-negativity of KL divergence. Equivalently, this property can also be derived as follows:

$$\begin{aligned} \log Z &= \log \int p^*(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \log \int q(\boldsymbol{\theta}) \frac{p^*(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} \\ &\geq \int q(\boldsymbol{\theta}) \log \frac{p^*(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta}. \qquad \text{\# Jensen's inequality} \end{aligned}$$

Here Jensen's inequality is applied to the logarithm which is concave. When posterior approximation is considered we also denote the two quantities as $\mathcal{F}_{\text{VFE}}(q; \mathcal{D})$ and $\mathcal{L}_{\text{VI}}(q; \mathcal{D})$, respectively. In summary, variational inference finds an approximation to the posterior through an *optimisation* process, which is drastically different from sampling approaches that construct *empirical point mass* distributions to describe the posterior.

### 1.2.1 A brief history of variational inference

Variational inference can be viewed as an application of *variational methods* that mathematicians and physicists have studied for centuries. Historically, physicists mainly focused on mean-field theories for complex systems [Parisi, 1988], whereas Dempster et al. [1977] as statisticians proposed the famous *expectation maximisation* (EM) algorithm that also has a VI interpretation [Neal and Hinton, 1998]. Interestingly the pioneers of deep learning had also applied variational inference (though under other names) for Bayesian neural networks [Peterson and Anderson, 1987, Hinton and Van Camp, 1993] that will be surveyed later. Especially since the development of [Peterson and Anderson, 1987], mean-field approximations started to be an attractive alternative to sampling methods for probabilistic inference in graphical models [Ghahramani, 1995].

However it was until Saul et al. [1996] which introduced the generic form of the variational lower-bound to explain the mean-field approximation. The first papers that I can find which coined the term "variational inference" are Lawrence et al. [1998] and Jordan et al. [1999], where Jordan et al. [1999] provided a detailed summary of the previous work coming from the same group. Later on, researchers started to extend the variational principle to cases beyond graphical models, e.g. the *variational Bayes* (VB) algorithm [Attias, 1999, 2000, Sato, 2001, Beal, 2003] that is used to perform posterior approximations of the model parameters and even model selection.

## 1.3 A mean-field approximation example

As an example for the variational inference algorithm, here we present the variational mean-field approximation [Parisi, 1988] for Bayesian linear regression. Readers are also refer to [Bishop, 2006] for more details and here we would briefly cover the derivations presented there. Mean-field approximation, also known as the factorised approximation, assumes the approximate posterior to be the form of

$$q(\boldsymbol{\theta}) := \prod_{i=1}^{D} q_i(\theta_i). \tag{12}$$

In general one can partition the elements of $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_D)$ into disjoint groups and apply factorisations over groups. This general case is usually called *structured* mean-field approximation [Saul and Jordan, 1996], and for simplicity in the following example we only consider the fully factorised case (12). Also we emphasise that there's no further assumption/restriction that is made on the functional form of $q_i(\theta_i)$. As we shall see, the variational free-energy is still convex in $q_i(\theta_i)$ and thus the solution provided by the following is the global optimum.

To derive the best approximation in the mean-field distribution family, we first substitute (12) into (10) (and use $\boldsymbol{\theta}_{\neq j}$ to denote all the $\theta_i$ variables except $\theta_j$):

$$\begin{aligned}
\mathcal{F}_{\text{VFE}}(q; p) &= \int \prod_i q_i(\theta_i) \left( \sum_i \log q_i(\theta_i) - \log p^*(\boldsymbol{\theta}) \right) d\boldsymbol{\theta} \\
&= \int q_j \log q_j(\theta_j) d\theta_j - \int q_j(\theta_j) \left( \int \prod_{i \neq j} q_i(\theta_i) \log p^*(\boldsymbol{\theta}) d\boldsymbol{\theta}_{\neq j} \right) d\theta_j + \text{const} \\
&:= \int q_j(\theta_j) \log q_j(\theta_j) d\theta_j - \int q_j(\theta_j) \log \tilde{p}(\theta_j) d\theta_j + \text{const},
\end{aligned}$$

where $\tilde{p}(\theta_j)$ denote the "marginal" distribution satisfying

$$\log \tilde{p}(\theta_j) = \int \prod_{i \neq j} q_i(\theta_i) \log p^*(\boldsymbol{\theta}) d\boldsymbol{\theta}_{\neq j} + \text{const}.$$

This means, by fixing the functional form of $q_i$ for all $i \neq j$, VFE is reduced to the KL-divergence $\text{KL}[q_j(\theta_j) || \tilde{p}(\theta_j)]$ plus a constant that is independent to $q_j$.

Thus the free-energy is still convex in $q_j$, in which the unique global optimum is obtained by setting $q_j(\theta_j) = \tilde{p}(\theta_j)$. To be precise, we explicitly write down the optimal mean-field approximation as

$$q(\theta_j) = \frac{\exp\left[\int \prod_{i\neq j} q_i(\theta_i) \log p^*(\boldsymbol{\theta}) d\boldsymbol{\theta}_{\neq j}\right]}{\int \exp\left[\int \prod_{i\neq j} q_i(\theta_i) \log p^*(\boldsymbol{\theta}) d\boldsymbol{\theta}_{\neq j}\right] d\theta_j}. \tag{13}$$

Now as an example consider Bayesian linear regression with 2-D inputs $\boldsymbol{x}$ and 1-D output $y$:

$$\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0^{-1}), \quad y|\boldsymbol{x} \sim \mathcal{N}(y; \boldsymbol{\theta}^T \boldsymbol{x}, \sigma^2).$$

Given the observations $\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, the posterior distribution of $\boldsymbol{\theta}$ can be computed analytically as $p(\boldsymbol{\theta}|\mathcal{D}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$ with $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_0 + \frac{1}{\sigma^2}\sum_n \boldsymbol{x}_n \boldsymbol{x}_n^T$ and $\boldsymbol{\Lambda}\boldsymbol{\mu} = \boldsymbol{\Lambda}_0\boldsymbol{\mu}_0 + \frac{1}{\sigma^2}\sum_n y_n \boldsymbol{x}_n$. To see how the mean-field approach works, we explicitly write down the elements of the posterior parameters

$$\boldsymbol{\mu} = \begin{pmatrix}\mu_1 \\ \mu_2\end{pmatrix}, \quad \boldsymbol{\Lambda} = \begin{pmatrix}\Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22}\end{pmatrix}, \quad \Lambda_{12} = \Lambda_{21},$$

Then by explicitly expanding the mean-field solution (13):

$$\begin{aligned}
\log q_1(\theta_1) &= \int q_2(\theta_2) \log p(\boldsymbol{\theta}, \mathcal{D}) d\theta_2 + \text{const} \\
&= \mathbb{E}_{q_2}\left[-\frac{1}{2}(\theta_1 - \mu_1)^2 \Lambda_{11} - (\theta_1 - \mu_1)\Lambda_{12}(\theta_2 - \mu_2)\right] + \text{const} \\
&= -\frac{1}{2}\theta_1^2 \Lambda_{11} + \theta_1 \mu_1 \Lambda_{11} - \theta_1 \Lambda_{12}(\mathbb{E}_{q_2}[\theta_2] - \mu_2) + \text{const} \\
&:= \log \mathcal{N}(\theta_1; m_1, \lambda^{-1}) + \text{const}
\end{aligned} \tag{14}$$

where the new mean $m_1$ and the precision $\lambda_1$ satisfies

$$m_1 = \mu_1 - \Lambda_{11}^{-1}\Lambda_{12}(\mathbb{E}_{q_2}[\theta_2] - \mu_2), \quad \lambda_1 = \Lambda_{11}.$$

It is important to note here that we do not assume the approximation to be a Gaussian distribution in order to obtain the last equation in (14). Rather the Gaussian distribution solution came out from the derivation of the global optimum (13) and the completion of the square form. One can derive the terms $m_2 = \mu_2 - \Lambda_{22}^{-1}\Lambda_{21}(\mathbb{E}_{q_1}[\theta_1] - \mu_1)$ and $\lambda_2 = \Lambda_{22}$ for $q_2$ in the same way, and show that $\boldsymbol{m} = \boldsymbol{\mu}$ is the only stable fixed point of this iterative update. So we have $q_1 = \mathcal{N}(\theta_1; \mu_1, \Lambda_{11}^{-1})$, and similarly $q_2 = \mathcal{N}(\theta_1; \mu_2, \Lambda_{22}^{-1})$ as the unique global optimum of variational mean-field approximation. A visualisation of the mean-field approximation is provided in Figure 1. Note here the variance parameter of $q(\theta_1)$ also correspond to the variance of the conditional distribution $p(\theta_1|\theta_2, \mathcal{D})$, which is smaller than the variance of the marginal distribution $p(\theta_1|\mathcal{D})$, and therefore mean-field VI under-estimates the posterior uncertainty in this case.[7]

[7] We also have $\mathbb{H}[p] = -\frac{1}{2}\log|\Lambda_{11}\Lambda_{22} - \Lambda_{12}\Lambda_{21}| + \text{const} \geq \mathbb{H}[q] = -\frac{1}{2}\log|\Lambda_{11}\Lambda_{22}| + \text{const}$.

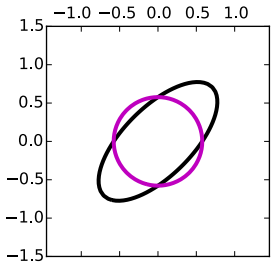


Figure 1: Mean-field approximation to the exact posterior distribution in the Bayesian linear regression example (with one-sigma contours). The exact posterior contour is shown in black and the variational approximation is in purple.

**Remark** (Does VI always fit to a local mode of the target distribution?)**.** VI is usually referred as a "mode seeking" method in that when a single-mode $q$ distribution (such as Gaussian) is fitted to a multi-mode distribution, the optimal $q$ solution often captures one of the modes. For example, consider the target distribution $p(\boldsymbol{\theta}) = \frac{1}{2}p_1(\boldsymbol{\theta}) + \frac{1}{2}p_2(\boldsymbol{\theta})$ where $supp(p_1) \cap supp(p_2) = \emptyset$. Due to the zero-forcing property of the exclusive KL-divergence (see § 1.1) $q$ is forced to fit one of the modes of $p$. However, if the mixture component $p_1$ and $p_2$ has a significant amount of overlapping density mass, then $q$ might not fit to one mode of $p$ only. For example, Figure 2 from Turner and Sahani [2011] showed that, the optimal variational approximation $q$ can even over-estimate the entropy of the target distribution $p$. Therefore it is a nice counter-example on the claim that variational inference approximation "always under-estimates the uncertainty".

## 1.4 Further reading

Jordan et al. [1999] presents VI in probabilistic graphical model context.

Wainwright and Jordan [2008]: a book-length paper that teaches both VI and message passing (and more!) in the context of probabilistic graphical models. I recommend reading chapters 2, 3 and 6 at this point, as they provide explanations on how variational methods relate to convex optimisation.

Beal [2003] is a highly-cited thesis mainly for variational inference. I recommend reading chapter 2 to start with, it shows how the EM algorithm relates to variational inference, and it also sketches variational EM as a fast variant.

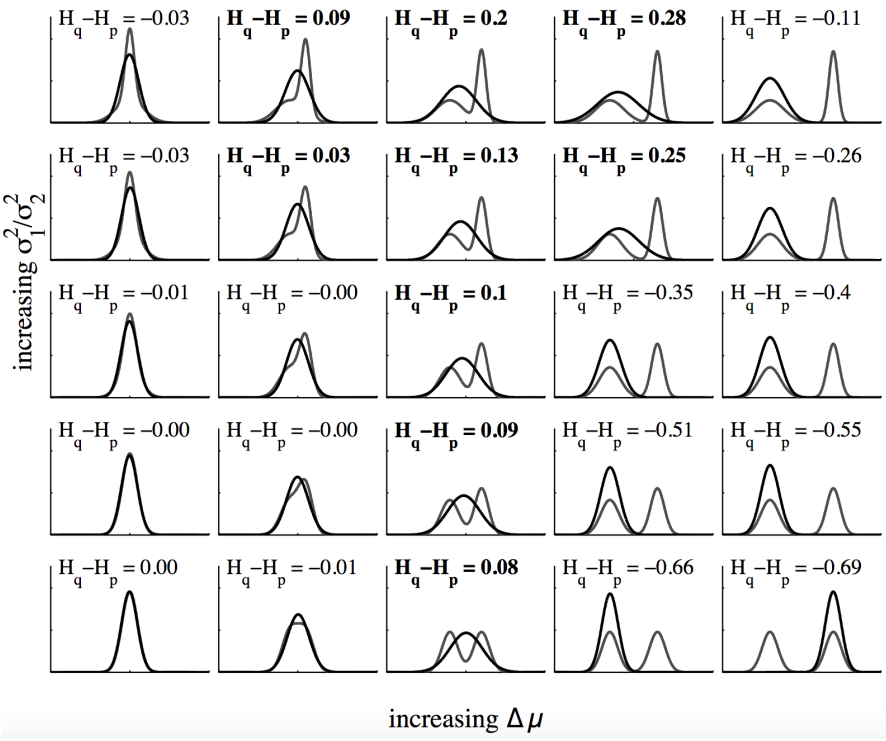


Figure 2: Variational approximation to the target distribution which is a mixture of two Gaussians: $p(\theta) = \frac{1}{2}\mathcal{N}(\theta; \mu_1, \sigma_1^2) + \frac{1}{2}\mathcal{N}(\theta; \mu_2, \sigma_2^2)$. The difference between means $\Delta\mu = \mu_1 - \mu_2$ increases from 0 for the left column panels to 10 for the right column panels. The ratio between variances $\sigma_1^2/\sigma_2^2$ (with fixed $\sigma_2^2 = 1$) also increases from 0 for the bottom row panels to 10 for the top row panels. Figure reproduced from Turner and Sahani [2011].

# 2 Monte Carlo variational inference

In the mean-field VI example we have discussed variational inference algorithms for linear regression. But real world problems are much more complicated. Often, the "reconstruction error" term $\mathbb{E}_q[\log p^*(\boldsymbol{\theta})]$ in the variational free energy lacks an analytical form. A prevalent example of such cases is variational inference for *large-scale data*: here the unnormalised distribution $p^*(\boldsymbol{\theta}) := p(\boldsymbol{\theta}, \mathcal{D})$ is proportional to the product of many likelihood functions, and evaluating $\mathbb{E}_q[\log p^*(\boldsymbol{\theta})]$ requires a pass of the whole dataset, which can be very expensive. On the other hand, insisting on having an analytical form of the entropy term $\mathbb{H}[q]$ (or $\mathrm{KL}[q||p_0]$) would restrict the selection of $q$ distributions to simple distributions like Gaussians. Usually the exact posterior is very complicated, and these simple distributions are expected to be poor approximations to the target distribution. Hence a key challenge here is, can we design a variational algorithm that applies to complex models, and scales to big data?

One solution to the above request is to develop further approximation techniques *specific to the chosen variational approximation*. Indeed in the early days researchers attempted to do so, e.g. see Jaakkola and Jordan [1998]. However these solutions are applicable only to a handful of special cases, making them impractical in many other interesting scenarios. Instead in this section we will review another approach which can be quickly applied to many cases with little effort. It has also been referred as a "black-box"[8] approach [Ranganath et al., 2014] due to this feature, but in the rest of the thesis we will refer it as Monte Carlo VI (MC-VI or MCVI) [Paisley et al., 2012, Wingate and Weber, 2013].

## 2.1 Monte Carlo estimation of the variational lower-bound

To see how MCVI works, consider approximating the exact posterior distribution $p(\boldsymbol{\theta}|\mathcal{D}) \propto \prod_n p(\boldsymbol{x}_n|\boldsymbol{\theta})p_0(\boldsymbol{\theta})$ by some simpler distribution $q(\boldsymbol{\theta})$. Rewriting the variational lower-bound:

$$\mathcal{L}_{\mathrm{VI}}(q; p) = \sum_{n=1}^{N} \mathbb{E}_q[\log p(\boldsymbol{x}_n|\boldsymbol{\theta})] + \mathbb{E}_q[\log p_0(\boldsymbol{\theta}) - \log q(\boldsymbol{\theta})], \tag{15}$$

we see that it is the analytical tractability requirement of computing the expectations that restrict the $q$ (and possibly the model $p$) distribution to be of simple form. This constraint can be removed by considering Monte Carlo (MC) approximation to the expectation, which estimates the expectation by, for example

$$\mathbb{E}_q[\log p(\boldsymbol{x}_n|\boldsymbol{\theta})] \approx \frac{1}{K} \sum_{k=1}^{K} \log p(\boldsymbol{x}_n|\boldsymbol{\theta}^k), \quad \boldsymbol{\theta}^k \sim q(\boldsymbol{\theta}). \tag{16}$$

This forms an *unbiased* estimation, and under mild assumptions, the RHS term in (16) converges to the exact expectation value as $K \to +\infty$. The KL-divergence

[8] Here "black-box" means that the method can be applied to inference problems of many probabilistic models without specific modifications of the optimisation algorithm. It does not imply the model $p$ or the $q$ distribution is a black-box (detailed in later topics).

term in the variational lower-bound can also be estimated with Monte Carlo in a similar manner. Also stochastic optimisation techniques can be extended here for scalability. In summary, with this "black-box" approach, one can approximate the variational lower-bound as

$$\mathcal{L}_{\text{VI}}^{MC}(q;p) = \frac{N}{|\mathcal{S}|}\sum_{n\in\mathcal{S}}\frac{1}{K}\sum_{k}\log p(\boldsymbol{x}_n|\boldsymbol{\theta}^k)+\frac{1}{K}\sum_{k}[\log p_0(\boldsymbol{\theta}^k)-\log q(\boldsymbol{\theta}^k)], \quad \boldsymbol{\theta}^k \sim q(\boldsymbol{\theta}), \tag{17}$$

and compute stochastic gradient descent on the MC approximation (17) with mini-batch $\mathcal{S} \sim \mathcal{D}^{|\mathcal{S}|}$.

> **Remark** (MC samples for different observations)**.** In the MC approximation (17) we assumed using the same set of samples $\{\boldsymbol{\theta}^k\}$ to estimate all the expectation terms. In general we can use different sets of samples to do so, for example, for every datapoint $\boldsymbol{x}_n \in \mathcal{S}$, we can sample different sets of $\boldsymbol{\theta}^k$ to estimate the associated reconstruction term $\mathbb{E}_q[\log p(\boldsymbol{x}_n|\boldsymbol{\theta})]$. Prevalent examples of this approach include stochastic regularisation techniques (SRTs) such as dropout [Srivastava et al., 2014, Kingma et al., 2015, Gal, 2016].

## 2.2 Computing the MCVI gradients

Practitioners care a lot more about the stochastic optimisation process of the MCVI algorithm compared to computing the MCVI bound. As the training of machine learning models relies on gradient descent based optimisation methods mainly, in this section we will detail some tricks of computing the gradient of the variational lower-bound using MC approximations.

### 2.2.1 LOTUS/reparameterisation trick and path gradients

We start by assuming $\boldsymbol{\theta}$ a continuous variable, and the discrete case will be discussed later. Here we introduce a neat trick called the *law of the unconscious statistician* (LOTUS). It has been extended to the *reparameterisation trick* in variational inference context [Salimans and Knowles, 2013, Kingma and Welling, 2014, Rezende et al., 2014]. This trick, along with MC approximation, makes the variational lower-bound easy to handle. It comes from a very simple observation: given a distribution $p(\boldsymbol{\theta})$, if sampling $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$ is equivalent to first sampling a "noise" variable $\boldsymbol{\epsilon} \sim \pi(\boldsymbol{\epsilon})$ and then computing a mapping $\boldsymbol{\theta} = \boldsymbol{f}(\boldsymbol{\epsilon})$, then the expectation of some function $F(\boldsymbol{\theta})$ under distribution $p(\boldsymbol{\theta})$ can be rewritten as

$$\mathbb{E}_{p(\boldsymbol{\theta})}[F(\boldsymbol{\theta})] = \mathbb{E}_{\pi(\boldsymbol{\epsilon})}[F(\boldsymbol{f}(\boldsymbol{\epsilon}))].$$

This derives from the change of variable in PDFs. LOTUS applies to any transformation, but the reparamterisation trick specifically asks the transformation $\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})$ to be differentiable w.r.t. its parameters $\boldsymbol{\phi}$. In variational inference context, the sampling procedure is required to be

$$\boldsymbol{\theta} \sim q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \Leftrightarrow \boldsymbol{\epsilon} \sim \pi(\boldsymbol{\epsilon}), \quad \boldsymbol{\theta} = \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}).$$

Then using the LOTUS rule, the variational lower-bound is computed as (we ignore the entropy term for a moment):

$$\mathcal{L}_{\text{VI}}(q_{\boldsymbol{\phi}}; p) = \mathbb{E}_{\pi(\boldsymbol{\epsilon})}[\log p(\mathcal{D}, \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))] + \mathbb{H}[q_{\boldsymbol{\phi}}], \tag{18}$$

and the gradient w.r.t. $\boldsymbol{\phi}$ is the following:

$$\begin{aligned}\nabla_{\boldsymbol{\phi}}\mathcal{L}_{\text{VI}}(q_{\boldsymbol{\phi}}; p) &= \nabla_{\boldsymbol{\phi}}\mathbb{E}_{\pi(\boldsymbol{\epsilon})}[\log p(\mathcal{D}, \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))] + \nabla_{\boldsymbol{\phi}}\mathbb{H}[q_{\boldsymbol{\phi}}] \\ &= \mathbb{E}_{\pi(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\phi}} \log p(\mathcal{D}, \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))] + \nabla_{\boldsymbol{\phi}}\mathbb{H}[q_{\boldsymbol{\phi}}] \quad \text{\# } \pi(\boldsymbol{\epsilon}) \text{ independent to } \boldsymbol{\phi} \\ &= \mathbb{E}_{\pi(\boldsymbol{\epsilon})}[\nabla_{\mathbf{f}} \log p(\mathcal{D}, \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))\nabla_{\boldsymbol{\phi}}\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})] + \nabla_{\boldsymbol{\phi}}\mathbb{H}[q_{\boldsymbol{\phi}}]. \quad \text{\# chain rule}\end{aligned} \tag{19}$$

The gradient derived by the chain rule is also called the *path* gradient. With MC approximation, the gradient of the "error" term in (19) is further approximated as

$$\frac{1}{K}\sum_{k=1}^{K} \nabla_{\mathbf{f}} \log p(\mathcal{D}, \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^k))\nabla_{\boldsymbol{\phi}}\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^k), \quad \boldsymbol{\epsilon}^k \sim \pi(\boldsymbol{\epsilon}). \tag{20}$$

Let us consider a simple but prevalent example, where the $q$ distribution is designed to be Gaussian, i.e. $q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$. In practice the covariance matrix $\boldsymbol{\Sigma}$ is often parameterised using its Cholesky decomposition $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$, in this case $\boldsymbol{\phi} = \{\boldsymbol{\mu}, \mathbf{L}\}$, and the transformation is written as $\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}) = \boldsymbol{\mu} + \mathbf{L}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This means the MC gradient of the error term is

$$\begin{aligned}\nabla_{\boldsymbol{\mu}} &= \frac{1}{K}\sum_{k=1}^{K} \nabla_{\boldsymbol{\mu}} \log p(\mathcal{D}, \boldsymbol{\mu} + \mathbf{L}\boldsymbol{\epsilon}^k), \\ \nabla_{\mathbf{L}} &= \frac{1}{K}\sum_{k=1}^{K} \nabla_{\mathbf{f}} \log p(\mathcal{D}, \boldsymbol{\mu} + \mathbf{L}\boldsymbol{\epsilon}^k)\boldsymbol{\epsilon}^k, \quad \boldsymbol{\epsilon}^k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).\end{aligned} \tag{21}$$

Choosing the number of MC samples $K$ in use requires a trade-off between speed and accuracy. In general, using small $K$ (even $K = 1$ in many cases) leads to high variance of the MC gradient. Using large $K$, on the other hand, can significantly slow down the training process in wall-clock time.

### 2.2.2 Path gradient of the entropy term

The derivation of the path gradient for the entropy term is slightly involved. This is because, apart from the transformation $\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})$, the PDF $q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ also depends on the variational parameters $\boldsymbol{\phi}$. Going back to the Gaussian example, we have the (log) PDF as

$$\log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu}) + \text{const},$$

in which the (log) normalising constant (or partition function) $\frac{1}{2}\log|\boldsymbol{\Sigma}| + \text{const}$ depends on $\boldsymbol{\phi}$ as well. Therefore, in general the gradient of the entropy term

w.r.t.$\boldsymbol{\phi}$ is

$$\begin{aligned}\nabla_{\boldsymbol{\phi}}\mathbb{H}[q_{\boldsymbol{\phi}}] &= -\nabla_{\boldsymbol{\phi}}\mathbb{E}_{\pi(\boldsymbol{\epsilon})}[\log q_{\boldsymbol{\phi}}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))] \\ &= -\mathbb{E}_{\pi(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))] \\ &= -\mathbb{E}_{\pi(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})} + \nabla_{\mathbf{f}} \log q_{\boldsymbol{\phi}}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))\nabla_{\boldsymbol{\phi}}\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})] \\ &= -\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}[\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})] - \mathbb{E}_{\pi(\boldsymbol{\epsilon})}[\nabla_{\mathbf{f}} \log q_{\boldsymbol{\phi}}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))\nabla_{\boldsymbol{\phi}}\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})].\end{aligned} \tag{22}$$

We see that due to the dependance of the PDF to $\boldsymbol{\phi}$ we have the gradient splitting into two terms. Interestingly, we can show that the first term (non-path gradient, which is also the expectation of the *score function*) in (22) eventually vanishes:

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}[\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})] &= \int q_{\boldsymbol{\phi}}(\boldsymbol{\theta})\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \int q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) q_{\boldsymbol{\phi}}(\boldsymbol{\theta})^{-1} \nabla_{\boldsymbol{\phi}} q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad \text{\# chain rule of log} \\ &= \int \nabla_{\boldsymbol{\phi}} q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \nabla_{\boldsymbol{\phi}} \int q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) d\boldsymbol{\theta} = 0. \quad \text{\# } q_{\boldsymbol{\phi}} \text{ always integrates to one}\end{aligned} \tag{23}$$

This means using the reparameterisation trick, the gradient of the entropy term w.r.t. the variational parameter $\boldsymbol{\phi}$ can also be derived as a path derivative:

$$\nabla_{\boldsymbol{\phi}}\mathbb{H}[q_{\boldsymbol{\phi}}] \approx -\frac{1}{K}\sum_{k=1}^{K} \nabla_{\mathbf{f}} \log q_{\boldsymbol{\phi}}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^k))\nabla_{\boldsymbol{\phi}}\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^k), \quad \boldsymbol{\epsilon}^k \sim \pi(\boldsymbol{\epsilon}). \tag{24}$$

Notice that if you implement the entropy term in a naive way like $\mathbb{H}[q_{\boldsymbol{\phi}}] \approx -\frac{1}{K}\sum_{k=1}^{K} \log q_{\boldsymbol{\phi}}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^k))$, and ask automatic differentiation to handle the gradient, the software will still include the (MC estimate of the) non-path gradient without knowing that it is actually zero in expectation. Roeder et al. [2017] empirically demonstrate that this can lead to high variance, and instead they suggest implement the MC approximated entropy in the following way:

$$\mathbb{H}[q_{\boldsymbol{\phi}}] \approx -\frac{1}{K}\sum_{k=1}^{K} \log q_{\boldsymbol{\phi}'}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^k)), \quad \boldsymbol{\phi}' = \text{stop\_gradient}(\boldsymbol{\phi}).$$

### 2.2.3 Log derivative trick and REINFORCE

The reparameterisation trick only works when there exists a differentiable transformation $\mathbf{f}_{\boldsymbol{\phi}}$ and the input noise variable $\boldsymbol{\epsilon}$ is independent to the variational parameter $\boldsymbol{\phi}$. This does not apply to discrete variables, although recent work has tried continuous relaxation techniques to enable path gradients [Maddison et al., 2017b, Jang et al., 2017]. So without any assumption on the random variable $\boldsymbol{\theta}$,

the gradient of the variational lower-bound w.r.t. $\boldsymbol{\phi}$ reads

$$\begin{aligned}\nabla_{\boldsymbol{\phi}}\mathcal{L}_{\text{VI}}(q_{\boldsymbol{\phi}};p) &= \int \nabla_{\boldsymbol{\phi}}\left(q_{\boldsymbol{\phi}}(\boldsymbol{\theta})\log\frac{p(\mathcal{D},\boldsymbol{\theta})}{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}\right)d\boldsymbol{\theta} \\ &= \int \log\frac{p(\mathcal{D},\boldsymbol{\theta})}{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}\nabla_{\boldsymbol{\phi}}q_{\boldsymbol{\phi}}(\boldsymbol{\theta})d\boldsymbol{\theta} + \int q_{\boldsymbol{\phi}}(\boldsymbol{\theta})\nabla_{\boldsymbol{\phi}}\log\frac{p(\mathcal{D},\boldsymbol{\theta})}{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}d\boldsymbol{\theta} \\ &= \int \log\frac{p(\mathcal{D},\boldsymbol{\theta})}{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}\nabla_{\boldsymbol{\phi}}q_{\boldsymbol{\phi}}(\boldsymbol{\theta})d\boldsymbol{\theta}. \quad \text{\# the second term is zero, see (23)}\end{aligned} \tag{25}$$

Now we apply the *log derivative trick* which is also named as the REINFORCE trick in reinforcement learning literature [Williams, 1992]. This trick is also closely related to the likelihood ratio method in statistics literature, e.g. see Glynn [1990]. It states that for any function $F$,

$$\begin{aligned}&\int F(\boldsymbol{\theta})\nabla_{\boldsymbol{\phi}}q_{\boldsymbol{\phi}}(\boldsymbol{\theta})d\boldsymbol{\theta} \\ =&\int F(\boldsymbol{\theta})\frac{\nabla_{\boldsymbol{\phi}}q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}q_{\boldsymbol{\phi}}(\boldsymbol{\theta})d\boldsymbol{\theta} \\ =&\int q_{\boldsymbol{\phi}}(\boldsymbol{\theta})F(\boldsymbol{\theta})\nabla_{\boldsymbol{\phi}}\log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})d\boldsymbol{\theta}.\end{aligned}$$

Therefore applying the log derivative trick to (25), we can see the MC approximated gradient is

$$\nabla_{\boldsymbol{\phi}}\mathcal{L}_{\text{VI}}(q_{\boldsymbol{\phi}};p) \approx \frac{1}{K}\sum_{k=1}^{K}\log\frac{p(\mathcal{D},\boldsymbol{\theta}^k)}{q_{\boldsymbol{\phi}}(\boldsymbol{\theta}^k)}\nabla_{\boldsymbol{\phi}}\log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}^k), \tag{26}$$

which is an unbiased, but high variance estimator of the exact gradient.

## 2.3 Variance reduction for MCVI gradients

It is now clear that in practice the optimisation problem of variational inference is often solved using MC approximated gradients. Therefore, the variance of the MC gradients is key to the performance, since if the variance is too high, then the MC gradient can point to wrong directions and thus slow down the convergence. In fact, variance reduction has become a major research topic not only in approximate inference but also in reinforcement learning and optimisation.[9] In this section we will briefly cover some important techniques for variance reduction in the context of variational inference.

### 2.3.1 Rao-Blackwellization

Let us start from a very simple example. We assume for now $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ and we would like to estimate $\mathbb{E}_{q(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}[F(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)]$ with Monte Carlo. Then the Rao-Blackwell theorem [Rao, 1965, Blackwell, 1947, Kolmogorov, 1950] states that,

[9] In NIPS 2016 three tutorials on these topics had spent considerate amount of time discussing variance reduction techniques.

the variance of the estimates can be reduced by conditioning on either $\boldsymbol{\theta}_1$ or $\boldsymbol{\theta}_2$. Mathematically, if we write $F_2(\boldsymbol{\theta}_2) = \mathbb{E}_{q(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)}[F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)]$, then the variance is

$$\begin{aligned}\mathbb{V}_{q(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}[F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)] =& \mathbb{E}_{q(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}[(F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) - \mathbb{E}_{q(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}[F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)])^2] \\ =& \mathbb{E}_{q(\boldsymbol{\theta}_2)}\mathbb{E}_{q(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)}[(F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) - F_2(\boldsymbol{\theta}_2) + F_2(\boldsymbol{\theta}_2) - \mathbb{E}_{q(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}[F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)])^2] \\ =& \mathbb{E}_{q(\boldsymbol{\theta}_2)}[(F_2(\boldsymbol{\theta}_2) - \mathbb{E}_{q(\boldsymbol{\theta}_2)}[F_2(\boldsymbol{\theta}_2)])^2] + \mathbb{E}_{q(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}[(F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) - F_2(\boldsymbol{\theta}_2))^2] \\ =& \mathbb{V}_{q(\boldsymbol{\theta}_2)}[F_2(\boldsymbol{\theta}_2)] + \mathbb{E}_{q(\boldsymbol{\theta}_2)}[\mathbb{V}_{q(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)}[F(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)]] \\ \geq& \mathbb{V}_{q(\boldsymbol{\theta}_2)}[F_2(\boldsymbol{\theta}_2)].\end{aligned} \tag{27}$$

How does Rao-Blackwellization apply to variance reduction for MCVI gradients? If we assume the approximate posterior factorises as $q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) = q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1)q_{\boldsymbol{\phi}_2}(\boldsymbol{\theta}_2)$, then the MCVI gradient for $\boldsymbol{\phi}_1$ (and similarly for $\boldsymbol{\phi}_2$) reads

$$\begin{aligned}\nabla_{\boldsymbol{\phi}_1}\mathcal{L}_{\text{VI}}(q_{\boldsymbol{\phi}}; p) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}\left[\log \frac{p(\mathcal{D}, \boldsymbol{\theta})}{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})} \nabla_{\boldsymbol{\phi}_1} \log q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1)\right] \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1)}\left[\mathbb{E}_{q_{\boldsymbol{\phi}_2}(\boldsymbol{\theta}_2)}\left[\log \frac{p(\mathcal{D}, \boldsymbol{\theta})}{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}\right] \nabla_{\boldsymbol{\phi}_1} \log q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1)\right] \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1)}\left[\left[\mathbb{E}_{q_{\boldsymbol{\phi}_2}(\boldsymbol{\theta}_2)}[\log p(\mathcal{D}, \boldsymbol{\theta})] - \log q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1) + \mathbb{H}[q_{\boldsymbol{\phi}_2}(\boldsymbol{\theta}_2)]\right] \nabla_{\boldsymbol{\phi}_1} \log q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1)\right] \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1)}\left[\left[\mathbb{E}_{q_{\boldsymbol{\phi}_2}(\boldsymbol{\theta}_2)}[\log p(\mathcal{D}, \boldsymbol{\theta})] - \log q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1)\right] \nabla_{\boldsymbol{\phi}_1} \log q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1)\right],\end{aligned} \tag{28}$$

where in the last line derivation we used the fact that $\mathbb{H}[q_{\boldsymbol{\phi}_2}(\boldsymbol{\theta}_2)]$ is a constant w.r.t. $q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1)$ and the same trick as in (23). This means, if we can compute $\mathbb{E}_{q_{\boldsymbol{\phi}_2}(\boldsymbol{\theta}_2)}[\log p(\mathcal{D}, \boldsymbol{\theta})]$ (or at least approximate it with many samples in a fast way), then the following MCVI gradient

$$\nabla_{\boldsymbol{\phi}_1}\mathcal{L}_{\text{VI}}(q_{\boldsymbol{\phi}}; p) \approx \frac{1}{K}\sum_{k=1}^{K}\left[\mathbb{E}_{q_{\boldsymbol{\phi}_2}(\boldsymbol{\theta}_2)}[\log p(\mathcal{D}, \boldsymbol{\theta}_1^k, \boldsymbol{\theta}_2)] - \log q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1^k)\right] \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1^k) \tag{29}$$

with $\boldsymbol{\theta}_1^k \sim q_{\boldsymbol{\phi}_1}(\boldsymbol{\theta}_1)$ will have smaller variance than the original version (26).

**Remark** (Local expectation gradient as Rao-blackwellization)**.** What if $q(\boldsymbol{\theta})$ represents some *structured* approximation to the exact posterior? In this case $q$ is specified by a *directed graphical model*

$$q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) = \prod_i q_{\boldsymbol{\phi}_i}(\theta_i|\text{pa}_i),$$

and $\text{pa}_i$ denotes the parents of node $\theta_i$. We also use $\boldsymbol{\theta}_{-i}$ to collect all the other entries $\theta_j, j \neq i$. Then, going through similar derivations as (28), we have

$$\nabla_{\boldsymbol{\phi}_i}\mathcal{L}_{\text{VI}}(q_{\boldsymbol{\phi}}; p) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\theta}_{-i})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}_i}(\theta_i|\text{mb}_i)}\left[\log \frac{p(\mathcal{D}, \boldsymbol{\theta})}{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}\right] \nabla_{\boldsymbol{\phi}_i} \log q_{\boldsymbol{\phi}_i}(\theta_i|\text{pa}_i)\right],$$

in which $\text{mb}_i$ denotes the Markov blanket of random variable $\theta_i$. Then the local expectation gradient method [Titsias and Lázaro-Gredilla, 2015] developed for

discrete variables applies similar Rao-blackwellization trick as in above: given a sample $\boldsymbol{\theta}_{-i} \sim q_{\boldsymbol{\phi}}(\boldsymbol{\theta}_{-i})$, the MCVI gradient can be computed as

$$\nabla_{\phi_i} \mathcal{L}_{\text{VI}}(q_{\boldsymbol{\phi}}; p) \approx \sum_{\theta_i} q_{\boldsymbol{\phi}_i}(\theta_i|\text{mb}_i) \left[\log \frac{p(\mathcal{D}, \theta_i, \boldsymbol{\theta}_{-i})}{q_{\boldsymbol{\phi}}(\theta_i, \boldsymbol{\theta}_{-i})}\right] \nabla_{\phi_i} \log q_{\boldsymbol{\phi}_i}(\theta_i|\text{pa}_i). \quad (30)$$

Notice here that the log-derivative trick is non-applicable because we never simulate samples from $q_{\boldsymbol{\phi}_i}(\theta_i|\text{mb}_i)$.

### 2.3.2 Control variate: general idea

Another important idea for variance reduction is control variate [Hammersley and Handscomb, 1964, Boyle, 1977]. Again we describe it for the general case, and the specific application of it to MCVI gradients is discussed in the next section. Assume we are interested in estimating $\mathbb{E}_{q(\boldsymbol{\theta})}[F(\boldsymbol{\theta})]$ with Monte Carlo. Then one can easily show that for any function $G(\boldsymbol{\theta})$ that has finite mean $\mathbb{E}_{q(\boldsymbol{\theta})}[G(\boldsymbol{\theta})]$, we have

$$\mathbb{E}_{q(\boldsymbol{\theta})}[F(\boldsymbol{\theta})] = \mathbb{E}_{q(\boldsymbol{\theta})}[F(\boldsymbol{\theta}) - G(\boldsymbol{\theta}) + \mathbb{E}_{q(\boldsymbol{\theta})}[G(\boldsymbol{\theta})]]. \quad (31)$$

Denote $\hat{F}(\boldsymbol{\theta}) = F(\boldsymbol{\theta}) - G(\boldsymbol{\theta}) + \mathbb{E}_{q(\boldsymbol{\theta})}[G(\boldsymbol{\theta})]$ and assume $\mathbb{V}_{q(\boldsymbol{\theta})}[G(\boldsymbol{\theta})] < +\infty$. Then the variance of $\hat{F}$ under $q$ is

$$\mathbb{V}_{q(\boldsymbol{\theta})}[\hat{F}(\boldsymbol{\theta})] = \mathbb{V}_{q(\boldsymbol{\theta})}[F(\boldsymbol{\theta})] + \mathbb{V}_{q(\boldsymbol{\theta})}[G(\boldsymbol{\theta})] - 2\text{Cov}_{q(\boldsymbol{\theta})}[F(\boldsymbol{\theta}), G(\boldsymbol{\theta})]. \quad (32)$$

This means, a clever choice of the $G$ function would make

$$\mathbb{V}_{q(\boldsymbol{\theta})}[G(\boldsymbol{\theta})] - 2\text{Cov}_{q(\boldsymbol{\theta})}[F(\boldsymbol{\theta}), G(\boldsymbol{\theta})] < 0,$$

and therefore it leads to a lower-variance estimator of $\mathbb{E}_{q(\boldsymbol{\theta})}[F(\boldsymbol{\theta})] \approx \hat{F}(\boldsymbol{\theta}), \boldsymbol{\theta} \sim q(\boldsymbol{\theta})$. Such choice of the $G$ function is called a control variate of $F$.

### 2.3.3 Some notable control variate methods for MCVI gradients

Now let us consider some notable examples of control variate that researchers has developed for MCVI. In this case the target function we consider is

$$F(\boldsymbol{\theta}) = \log \frac{p(\mathcal{D}, \boldsymbol{\theta})}{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})} \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}).$$

In some papers $f(\boldsymbol{\theta}) = \log \frac{p(\mathcal{D}, \boldsymbol{\theta})}{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}$ is also referred as the learning signal for the variational parameters $\boldsymbol{\phi}$ [Mnih and Gregor, 2014]. Many of the techniques described in below also applies to variance reduction for policy gradient in reinforcement learning.

- **Optimal scaling for score functions**:
  If we further define $G(\boldsymbol{\theta}) = \lambda g(\boldsymbol{\theta})$, then by (32), the reduction of variance is

$$\mathbb{V}_{q(\boldsymbol{\theta})}[F(\boldsymbol{\theta})] - \mathbb{V}_{q(\boldsymbol{\theta})}[\hat{F}(\boldsymbol{\theta})] = 2\lambda \text{Cov}_{q(\boldsymbol{\theta})}[F(\boldsymbol{\theta}), g(\boldsymbol{\theta})] - \lambda^2 \mathbb{V}_{q(\boldsymbol{\theta})}[g(\boldsymbol{\theta})].$$

Therefore, the optimal scaling $\lambda$ can be derived by maximising the variance reduction, which gives

$$\lambda^* = \frac{\mathrm{Cov}_{q(\boldsymbol{\theta})}[F(\boldsymbol{\theta}), g(\boldsymbol{\theta})]}{\mathbb{V}_{q(\boldsymbol{\theta})}[g(\boldsymbol{\theta})]}.$$

Ranganath et al. [2014] considers using the score function as a control variate, i.e. $g(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$, where from (23) we have $\mathbb{E}_{q(\boldsymbol{\theta})}[G(\boldsymbol{\theta})] = \mathbf{0}$.[10] This implies

$$\hat{F}_{\text{opt}}(\boldsymbol{\theta}) = [f(\boldsymbol{\theta}) - \lambda^*] \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}).$$

However in practice it is impossible to compute the exact optimal scaling $\lambda^*$, and instead we typically form an (MC) estimate $\hat{\lambda} \approx \lambda^*$, which also introduces extra variances. Still empirically, Ranganath et al. [2014] shows that this control variate works well in some practical cases.

- **Neural variational inference and learning (NVIL)**:
  Mnih and Gregor [2014] also considers $G(\boldsymbol{\theta}) = \lambda \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$. However, instead of estimating the optimal scaling, the authors notice that $\lambda$ can be any function that is independent to $\boldsymbol{\theta}$. Therefore, they parameterise $\lambda = C_{\boldsymbol{\psi}}(\mathcal{D}) - c$ where $C_{\boldsymbol{\psi}}(\mathcal{D})$ is a neural network with parameters $\boldsymbol{\psi}$,[11] and train the neural network by minimising the $\ell_2$ error

  $$\min_{\boldsymbol{\psi}} \quad \mathbb{E}_{q_{\boldsymbol{\phi}}} \left[ \left( f(\boldsymbol{\theta}) - C_{\boldsymbol{\psi}}(\mathcal{D}) - c \right)^2 \right].$$

  Therefore $C_{\boldsymbol{\psi}}(\mathcal{D}) + c$ is also called "data dependent baselines". Although the optimal solution in this case does not correspond to the optimal scaling, Mnih and Gregor [2014] argued that this alternative objective is much easier to minimise, and empirically, the resulting control variate performed quite well in their experiments. To further reduce variance in the scaling, NVIL also normalises the difference term with an estimate of its standard deviation (unless when $\boldsymbol{\phi}$ is close to a local optimum), making the final gradient as

  $$\hat{F}_{\text{NVIL}}(\boldsymbol{\theta}) = \frac{f(\boldsymbol{\theta}) - C_{\boldsymbol{\psi}}(\mathcal{D}) - c}{\hat{\sigma}[f(\boldsymbol{\theta}) - C_{\boldsymbol{\psi}}(\mathcal{D})]} \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}).$$

  I would doubt whether this normalisation step is necessary, if scale-invariant gradient descent methods like Adagrad [Duchi et al., 2011], RMSprop [Tieleman and Hinton, 2012] and Adam [Kingma and Ba, 2015] is in use. In their experiments Mnih and Gregor [2014] applied stochastic gradient descent, and in that case the normalisation technique can be of great help. In fact they estimated the standard-deviation using exponential moving average, which makes the resulting algorithm closely related to RMSprop [Tieleman and Hinton, 2012].

[10] Interestingly as we have mentioned, Roeder et al. [2017] has shown that in path gradient settings including the score function term can lead to high variance. However the authors didn't consider optimal scalings for the control variate.

[11] In the applications that Mnih and Gregor [2014] considered, $\boldsymbol{\theta}$ corresponds to the latent variables and $\mathcal{D} = \boldsymbol{x}$.

- **Taylor expansion for the learning signal**:
  The control variate for MCVI is not limited to the score function up to constant scaling. Indeed, if we define $G(\boldsymbol{\theta}) = [h(\boldsymbol{\theta}) + b]\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$, then (31) expands to

  $$\mathbb{E}_{q(\boldsymbol{\theta})}[F(\boldsymbol{\theta})] = \mathbb{E}_{q(\boldsymbol{\theta})}[(f(\boldsymbol{\theta}) - b - h(\boldsymbol{\theta}))\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})] + \mathbb{E}_{q(\boldsymbol{\theta})}[h(\boldsymbol{\theta})\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})].$$

  Now we assume $f(\boldsymbol{\theta})$ is differentiable w.r.t. $\boldsymbol{\theta}$ which is often true for the density ratio in the VI case. Then we can use Taylor expansion at some location $\boldsymbol{\theta}_0$ that is independent to $\boldsymbol{\theta}$ to define the control variate, for example $b + h(\boldsymbol{\theta}) = f(\boldsymbol{\theta}_0) + \nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{\theta}_0)(\boldsymbol{\theta} - \boldsymbol{\theta}_0)$. By rearranging terms and applying the identity (23), we have

  $$\mathbb{E}_{q(\boldsymbol{\theta})}[F(\boldsymbol{\theta})] = \mathbb{E}_{q(\boldsymbol{\theta})}[\epsilon_f(\boldsymbol{\theta}, \boldsymbol{\theta}_0)\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})] + \mathbb{E}_{q(\boldsymbol{\theta})}[\nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{\theta}_0)\boldsymbol{\theta}\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})],$$

  with $\epsilon_f(\boldsymbol{\theta}, \boldsymbol{\theta}_0) = f(\boldsymbol{\theta}) - f(\boldsymbol{\theta}_0) - \nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{\theta}_0)(\boldsymbol{\theta} - \boldsymbol{\theta}_0)$ as the Taylor expansion error. One can further use the log derivative trick to show that the second term reduces to $\mathbb{E}_{q(\boldsymbol{\theta})}[\nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{\theta}_0)\boldsymbol{\theta}\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})] = \nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{\theta}_0)\mathbb{E}_{q(\boldsymbol{\theta})}[\boldsymbol{\theta}]$. Higher order Taylor expansion has also been explored, see Paisley et al. [2012], Gu et al. [2016].

## 2.4 Further reading

Ranganath et al. [2014] described the black-box VI (BBVI) algorithm that is referred as MCVI in this note. Paisley et al. [2012], Wingate and Weber [2013] also described very similar approaches, however Ranganath et al. [2014] has further detailed discussions on variance reduction techniques.

Should definitely read [Kingma and Welling, 2014] for the reparameterisation trick. Around the same time the trick was also described in Salimans and Knowles [2013], Rezende et al. [2014], which unfortunately get less citations. I will also suggest a come-back reading on these papers when we discuss applications of approximate inference to generative models.

Opper and Archambeau [2009] describes another gradient estimator for the variational parameters of Gaussian approximations. The idea is less well-known but still very interesting.

We can construct "reparameterisations" for discrete variables as well if we can compute the inverse CDF. For general invertible transforms, Ruiz et al. [2016] constructed a generalised reparameterisation gradient that contains both path gradient terms and REINFORCE like terms. Similar ideas have also been explored in e.g. Tucker et al. [2017].

In practice many of these MCVI methods are implemented using automatic differentiation, therefore it might be useful to understand how automatic differentiation works. For example see Baydin et al. [2015] for a survey.

I briefly discussed control variate methods applied to MCVI (and policy gradients). If you are interested in variance reduction methods for stochastic/distributed optimisation, then I would suggest reading papers for the following algorithms to start with: SAG [Le Roux et al., 2012, Schmidt et al., 2013], SVRG [Johnson and Zhang, 2013], and SAGA [Defazio et al., 2014].

# 3 Approximate inference with implicit distributions

We have yet to discuss the design of approximate distributions which is also a very important research topic and will be the focus of theme II. However, in this section I will review another very new research direction: concretely, instead of designing approximate posterior distributions that fit into standard frameworks like VI, we would like to develop optimisation algorithms that enable approximations of *arbitrary* form. The introduction in this section is rewritten from Chapter 5 of my thesis and Li and Liu [2016], where we name this regime as "wild approximate inference".

## 3.1 Revisiting tractability issues in approximate inference

In § 0.3 the definition of an approximate inference procedure is identified. However:

*What does tractability mean for an approximate inference algorithm?*

To answer the above question, we first start by revisiting the definition of *approximate inference*, with Bayesian posterior inference as an illustrating example. Assume a model with prior distribution $p(\boldsymbol{z})$ and likelihood function $p(\boldsymbol{x}|\boldsymbol{z})$. Then *inference* means computing the expectation of some function $F(\boldsymbol{z})$ under the exact posterior, which is $\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[F(\boldsymbol{z})]$. Examples of such $F$ functions include:

- $F(\boldsymbol{z}) = \boldsymbol{z}^k$, i.e. computing the moments of $p$;
- $F(\boldsymbol{z}) = p(\mathbf{y}^*|\boldsymbol{z}, \boldsymbol{x}^*)$ if in supervised learning and $\boldsymbol{z}$ represents the model parameters;
- $F(\boldsymbol{z}) = \delta_A$ if one wishes to evaluate $p(\boldsymbol{z} \in A|\boldsymbol{x}) = \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[\delta_A]$.

For simplicity in the rest of the discussion we assume the evaluation of $F(\boldsymbol{z})$ can be done using available computational resources, otherwise it needs more approximations.

The core idea of (optimisation based) approximate inference is to fit an approximate posterior distribution $q(\boldsymbol{z}|\boldsymbol{x})$ in a "tractable" distribution family $\mathcal{Q}$ to the exact posterior $p(\boldsymbol{z}|\boldsymbol{x})$, such that $\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[F(\boldsymbol{z})]$ can be well approximated by

$$\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[F(\boldsymbol{z})] \approx \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[F(\boldsymbol{z})]. \tag{33}$$

Critically, the primary tractability requirement here for the approximate posterior is the *fast computation* of the *approximate expectation* on the RHS given the function $F$.

Historically, approximate distributions of simple forms, such as mean-field approximations and factorised Gaussians [Jordan et al., 1999], have been proposed to obtain analytical solutions of the approximated expectation. These approaches often require the probabilistic model to comprise conjugate exponential families,

which excludes a broad range of powerful models, e.g. those who warp noise variables through non-linear mappings. Instead, modern approximate inference introduces Monte Carlo (MC) estimation techniques to approximate the predictive likelihood [Paisley et al., 2012, Ranganath et al., 2014] that we reviewed in § 2. The MC method enables a wider class of models to be amendable to VI (the requirement is that the log-joint can be computed point-wise), and is key to modern training methods of generative models such as the VAE [Kingma and Welling, 2014, Rezende et al., 2014].

Precisely, at inference time, the MC approximation method samples $\{\boldsymbol{z}^1, ..., \boldsymbol{z}^K\}$ from the approximate posterior $q$, and estimates the required quantity by

$$\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[F(\boldsymbol{z})] \approx \frac{1}{K}\sum_{k=1}^{K} F(\boldsymbol{z}^k), \quad \boldsymbol{z}^k \sim q(\boldsymbol{z}|\boldsymbol{x}). \tag{34}$$

Consequently, this converts the *fast expectation computation* requirement to *fast sampling* from the approximate posterior, as the expectation is further approximated by the empirical average. Fast sampling is arguably a stronger condition compared to fast expectation computation, as for the latter, one would normally expect fast calculations for a *given* function $F$. The latter is typically the case for traditional numerical integration methods since for different functions one would select different quadrature rules. On the other hand, once we have obtained the samples from the approximate posterior, we can use them to compute an empirical estimate of the expectation for *any* function. Hence methods that entail fast sampling might be preferred for tasks that require estimating expectations of a set of functions.

Unfortunately, except a few very recent attempts that will be detailed later, most approximate inference algorithms impose further constraints to the design of $q$. For example, recall the MC-VI objective in § 2:

$$\mathcal{L}_{\text{VI}}^{\text{MC}}(q; p) = \frac{1}{K}\sum_{k=1}^{K} \log p(\boldsymbol{x}, \boldsymbol{z}^k) - \log q(\boldsymbol{z}^k|\boldsymbol{x}), \quad \boldsymbol{z}^k \sim q(\boldsymbol{z}^k|\boldsymbol{x}). \tag{35}$$

Then it is clear that the training procedure requires *fast density evaluation*, or at leat *fast log-density gradient evaluation* for $q(\boldsymbol{z}|\boldsymbol{x})$ given a configuration of $\boldsymbol{z}$. Importantly, this requirement is only presented in the VI optimisation procedure to seek for the best fit of $q$: once obtain a (local) optimum, MC inference only requires evaluating the empirical expectation thus no need to compute the density point-wise.

The above observations raise an outstanding research question: *can we design efficient approximate inference algorithms to train flexible approximate posterior distributions which are implicit, i.e. without access to an explicit density function?* We will answer this in the next sections, by discussing proposals for training *wild approximate inference algorithms*,[12] which, in fact, allows fitting approximations of *arbitrary* forms. We will also provide examples of some approximate posterior

[12] Not to be confused with *black-box* variational inference [Ranganath et al., 2014].

distributions that is not possible to be fitted using conventional approximate inference algorithms. Before that, I first address potential skepticisms in below, and hopefully this will assure the readers why the new research direction is outstanding.

### 3.1.1 Is it necessary to evaluate the approximate posterior density?

One might argue that having an accessible $q$ density allows the user to understand the properties of the exact posterior better. It might be true for low dimensional cases, as we can easily visualise the density function, and compare density values between samples to determine which is more probable. But I would disagree with this argument for the scenario of approximating multi-modal posterior distributions in high dimensions, which is typically the case that wild approximations apply. Some reasons are:

- First, enforcing the tractable density constraint means that in many cases, either we fit the posterior with a rather simple distribution (which has limited representational power), or a complex model such as a mixture density (which entails high computational costs).
- Second, even when setting aside the computational issues for density evaluation, visualising high dimensional distributions is itself still an open research problem. In this regard, many data visualisation techniques consider dimension reduction methods such as PCA, self-organising map [Kohonen, 1998, Venna and Kaski, 2003] and t-SNE [Maaten and Hinton, 2008], which in fact only require samples from the distribution, not the density values.
- Finally as motivated above, the MC integration task does not require evaluating or comparing density values on samples. For those which do require density evaluation, one can then fit a density estimator on the samples from $q$. It can still be very convenient as in the MC estimation setting we typically assume fast sampling from the approximate posterior.

### 3.1.2 Comparisons to sampling-based methods

Many Bayesian statisticians prefer sampling methods – and in fact it is the emergence of sampling methods such as importance sampling (IS), sequential Monte Carlo (SMC) [Doucet et al., 2001] and Markov chain Monte Carlo (MCMC) that contributes to the rapid development of Bayesian statistics. They have very nice theoretical guarantees, for example, IS and SMC provide unbiased estimates of the integral and are asymptotically exact when the number of samples $K \to +\infty$. MCMC has similar asymptotic exactness guarantee but it also requires the number of transitions $T \to +\infty$. However, I view all these sampling methods as approximate inference algorithms, simply due to the fact that in practice one can never obtain an infinite number of samples, nor simulating the MCMC dynamics for an infinite amount of time. Furthermore, even some of these methods do construct *implicit* approximate posterior distributions in practice, they still add more constraints to the inference procedures, detailed in below.

- (Adaptive) importance sampling (IS) and sequential Monte Carlo (SMC).
  Importance sampling has a long history in statistics, e.g. see Geweke [1989]. Roughly speaking, it proposes sampling from a rather simple distribution $\pi(\boldsymbol{z}|\boldsymbol{x})$, then "correcting" the sampling estimate by incorporating the importance weight

$$\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[F(\boldsymbol{z})] \approx \frac{1}{K}\sum_{k=1}^{K} w_k F(\boldsymbol{z}^k), \quad w_k = \frac{p(\boldsymbol{z}^k|\boldsymbol{x})}{\pi(\boldsymbol{z}^k|\boldsymbol{x})}, \quad \boldsymbol{z}^k \sim \pi(\boldsymbol{z}|\boldsymbol{x}). \tag{36}$$

  One can easily see the unbiasedness of the IS estimate, and under mild conditions one can also show it is consistent. Also self-normalised IS is sometimes used to obtain approximate posterior samples, which effectively constructs $q$ as

$$q(\boldsymbol{z}|\boldsymbol{x}) = \sum_{k=1}^{K} \hat{w}_k \delta(\boldsymbol{z} = \boldsymbol{z}^k), \quad \hat{w}_k = \frac{w_k}{\sum_{k=1}^{K} w_j}, \quad \boldsymbol{z}^k \sim \pi(\boldsymbol{z}|\boldsymbol{x}). \tag{37}$$

  In this case the $q$ distribution depends on the proposal $\pi$ and the number of samples $K$, and again under mild conditions $q \to p$ when $K \to +\infty$. Importantly, the $q$ distribution is tractable and requires fast evaluation of the $\pi$ density. SMC can be viewed as importance sampling applied to time-series models (such as hidden Markov models), typically with extra techniques to improve sample efficiency.

  However, IS and SMC provide terrible approximations to the desired integral if the proposal $\pi$ is very different from the target distribution $p$, mainly due to the high variance of the estimator. To address this issue, researchers have considered adapting the initial distribution to reduce the variance, therefore improving sample efficiency that is key to the success of IS in practice, Indeed the (unnormalised) optimal proposal distribution for IS is proportional to $|F(\boldsymbol{z})|p(\boldsymbol{z}|\boldsymbol{x})$, and in some cases the resulting estimator has zero variance, indicating that it requires only one (!) sample to compute the exact integral. Recently there is a plenty of research work on how to adapt the initial distribution and combine with amortised inference [Cornebise, 2009, Gu et al., 2015, Burda et al., 2016, Paige and Wood, 2016, Le et al., 2017, Naesseth et al., 2017, Maddison et al., 2017a]. But still, the tractability constraint of $\pi(\boldsymbol{z}|\boldsymbol{x})$ largely restricts its analytic form to those have been used for VI.

- Markov Chain Monte Carlo (MCMC).
  An MCMC algorithm is typically specified by a *transition distribution* (or transition kernel) $\mathcal{T}(\boldsymbol{z}'|\boldsymbol{z})$ where the following holds:

  (i) $\mathcal{T}$ has the target distribution $p(\boldsymbol{z}|\boldsymbol{x})$ as the *unique* stationary distribution:

$$p(\boldsymbol{z}'|\boldsymbol{x}) = \int \mathcal{T}(\boldsymbol{z}'|\boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z}.$$

(ii) If defining

$$\mathcal{T}_T(\boldsymbol{z}_T|\boldsymbol{z}_0) = \int \prod_{t=0}^{T-1} \mathcal{T}(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t) d\boldsymbol{z}_{0:T-1},$$

then for any initial distribution $q(\boldsymbol{z}|\boldsymbol{x})$ the MCMC dynamics converges to the target distribution as $T \to +\infty$:

$$\lim_{T\to+\infty} q_T(\boldsymbol{z}|\boldsymbol{x}) = p(\boldsymbol{z}|\boldsymbol{x}), \quad q_T(\boldsymbol{z}|\boldsymbol{x}) := \int \mathcal{T}_T(\boldsymbol{z}|\boldsymbol{z}')q(\boldsymbol{z}'|\boldsymbol{x})d\boldsymbol{z}'.$$

Conditions that such transition kernel $\mathcal{T}$ requires are described in e.g. [Gelman et al., 1995, chapter 11].

In practice one often specifies an initial distribution $q_0(\boldsymbol{z}|\boldsymbol{x})$ to draw starting particles, and stops simulating the transitions after $T$ steps according to his/her computational budget. Consequently, this truncated Markov chain also induces an implicit $q$ distribution

$$q(\boldsymbol{z}|\boldsymbol{x}) = \int \mathcal{T}_T(\boldsymbol{z}|\boldsymbol{z}')q_0(\boldsymbol{z}'|\boldsymbol{x})d\boldsymbol{z}'. \tag{38}$$

In many applications the computational budget only allows simulations of a small number of transitions, e.g. when training big models with EM. Thus having a rapidly mixed chain would significantly reduce $T$, and to achieve this goal a lot of work has explored different designs of the transition kernel, to name a few see Ahn et al. [2012], Duane et al. [1987], Neal et al. [2011], Girolami and Calderhead [2011], Ding et al. [2014]. However, these methods still ensure asymptotic exactness of the resulting MCMC algorithm, which again imposes restrictions on the transition kernel design.

Observing the above, I would argue that recent advances of sampling-based inference method do not achieve the best *speed-accuracy trade-off* that is one of the most important topics in approximate inference. Indeed as we will not be able to obtain the exact posterior from $T$-step MCMC simulations anyway, removing the asymptotic exactness requirement can potentially allow the best fit of the transition kernel, which makes $q(\boldsymbol{z}|\boldsymbol{x}) = q_T(\boldsymbol{z}|\boldsymbol{x})$ the best approximator to the exact posterior in such a $\mathcal{Q}$ class. Similarly for IS, there exist estimators that use some "super-efficient" weights to allow significantly faster convergence rates than $\mathcal{O}(K^{-\frac{1}{2}})$ the usual convergence rate for the IS estimator (36) [Liu and Lee, 2017, O'Hagan, 1991, Ghahramani and Rasmussen, 2003, Oates et al., 2017]. More specifically, the recipe provided by Liu and Lee [2017] does not require a tractable initial distribution $\pi$ at all. Although these estimators can be biased, in practice they often provide better speed-accuracy trade-off due to their better sample efficiency.

In summary, from an approximation perspective, it is also an interesting to allow constructions of implicit initial distributions and transition kernels for sampling-based inference methods. Much work is still to be done in this vein, and those approaches will be included in this note in the future. In the following we only discuss wild approximate inference methods to directly fit an approximate posterior.

## 3.2 Algorithmic options

Implicit distributions cannot be fitted using traditional approximate inference methods such as MC-VI. In this section, we discuss algorithmic options for training these approximations to the posterior.[13] In short, I will cover:

- **energy approximation:** methods to approximate the variational lower-bound given an implicit $q$ distribution;
- **gradient approximation:** methods to approximate the gradient of the variational lower-bound;
- **alternative divergence minimisation:** two methods using the Stein's discrepancy;
- **amortising deterministic/stochastic dynamics:** training a fast sampler that can (approximately) produce MCMC/SVGD samples;
- **other approaches:** e.g. Titsias [2017].

As the readers might have noticed, the references provided in this section are very freshly baked papers. And the categories I include here are not exhaustive. Readers who are interested in this research direction are much encouraged to come up their own solutions. Also most of the approaches covered here assume $q$ to be reparameterisable, i.e. $\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x}) \Leftrightarrow \boldsymbol{\epsilon} \sim \pi(\boldsymbol{\epsilon}), \boldsymbol{z} = \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x})$. It remains a main challenge to extend implicit distribution fitting to discrete distributions in general.

### 3.2.1 Energy approximation

Assume $q$ is reparameterisable, then by the chain rule, the gradient $\nabla_{\boldsymbol{\phi}}\mathcal{L}$ is computed as $\nabla_{\mathbf{f}}\mathcal{L}\nabla_{\boldsymbol{\phi}}\mathbf{f}$. Therefore, if we have an approximation $\hat{\mathcal{L}}$ to the objective function, then we can approximate the gradient as $\nabla_{\boldsymbol{\phi}}\mathcal{L} \approx \nabla_{\mathbf{f}}\hat{\mathcal{L}}\nabla_{\boldsymbol{\phi}}\mathbf{f}$. We refer this approach as *energy/objective approximation*.

A popular idea considers density ratio estimation methods [Qin, 1998, Sugiyama et al., 2009, 2012] for energy approximation, which is concurrently considered in Li and Liu [2016], Karaletsos [2016], Mescheder et al. [2017], Huszár [2017], Tran et al. [2017] and later in Shi et al. [2017]. This is done by introducing an auxiliary distribution $\tilde{q}$ and rewrite the variational lower-bound:

$$\mathcal{L}_{\text{VI}}(\boldsymbol{\theta}, q; \boldsymbol{x}) = \mathbb{E}_q \left[ \log \frac{p_0(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta})}{\tilde{q}(\boldsymbol{z}|\boldsymbol{x})} + \log \frac{\tilde{q}(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z}|\boldsymbol{x})} \right]. \tag{39}$$

The auxiliary distribution $\tilde{q}$ is required to have tractable density and is easy to sample. Then one can use sample-based density ratio estimation methods to fit a model $\tilde{R}$ for the ratio between $\tilde{q}$ and $q$. The gradient approximation for general $\tilde{q}$ distributions can be derived similarly as

$$\nabla_{\boldsymbol{\phi}}\mathcal{L}_{\text{VI}} = \mathbb{E}_q \left[ \nabla_{\boldsymbol{\phi}} \log \frac{p_0(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta})}{\tilde{q}(\boldsymbol{z}|\boldsymbol{x})} + \nabla_{\boldsymbol{z}}\tilde{R}(\boldsymbol{z}, \boldsymbol{x})\nabla_{\boldsymbol{\phi}}\mathbf{f} \right]. \tag{40}$$

[13] We note here that the discussed options are applicable to tractable $q$ distributions as well.

In the following we briefly show that the original GAN approach [Goodfellow et al., 2014] can be applied as a density ratio estimator. Consider a discriminator $D(\boldsymbol{z}, \boldsymbol{x})$ which outputs the probability of a sample $\boldsymbol{z}$ coming from the auxiliary distribution $\tilde{q}(\boldsymbol{z}|\boldsymbol{x})$. Using the same calculation in Goodfellow et al. [2014] one can easily show that the optimal discriminator is

$$D^*(\boldsymbol{z}, \boldsymbol{x}) = \frac{\tilde{q}(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z}|\boldsymbol{x}) + \tilde{q}(\boldsymbol{z}|\boldsymbol{x})} = \frac{1}{1 + \exp(-\log \frac{\tilde{q}(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z}|\boldsymbol{x})})} = \text{sigmoid}\left(\log \frac{\tilde{q}(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z}|\boldsymbol{x})}\right). \tag{41}$$

As in practice the discriminator is often parameterised using a neural network with sigmoid activation, we can define $D(\boldsymbol{z}, \boldsymbol{x}) = \text{sigmoid}(\tilde{R}(\boldsymbol{z}, \boldsymbol{x}))$ and use $\tilde{R}(\boldsymbol{z}, \boldsymbol{x})$ as the density-ratio estimator. Using the GAN training this density-ratio estimator will get improved towards the exact ratio. Note that this density-ratio estimator can also be obtained using $f$-GAN [Nowozin et al., 2016].

An alternative approach in Shi et al. [2017] applies kernel methods for density ratio estimation. In short, the authors considers minimising the following objective to fit an approximation $R(\boldsymbol{z}, \boldsymbol{x}) \approx \frac{\tilde{q}(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z}|\boldsymbol{x})}$:

$$\begin{aligned} \mathcal{L}(R) &= \mathbb{E}_q \left[ \left( R(\boldsymbol{z}, \boldsymbol{x}) - \frac{\tilde{q}(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z}|\boldsymbol{x})} \right)^2 \right] \\ &= \mathbb{E}_q \left[ R(\boldsymbol{z}, \boldsymbol{x})^2 \right] - 2\mathbb{E}_{\tilde{q}} \left[ R(\boldsymbol{z}, \boldsymbol{x}) \right] + C \end{aligned} \tag{42}$$

Then they follow the kernel density ratio estimator method (e.g. see Sugiyama et al. [2009]), which parameterises $R$ with a kernel machine, and obtains analytical solutions for the linear coefficients. By doing so, neither discriminator nor double loop training are required.

**Remark** (the choice of the auxiliary $\tilde{q}$)**.** A simple example considers $\tilde{q} = p_0$ and the classification approach for ratio estimation [Karaletsos, 2016, Huszár, 2017, Tran et al., 2017]. However in practice, density ratio estimation works poorly if the two distribution in comparison are very different, especially in the case that the regions contains most of the probability mass have little overlap. Instead Mescheder et al. [2017] discussed an advanced technique termed as *adaptive contrast*, which takes $\tilde{q}$ as a Gaussian approximation to the wild distribution $q$. In this case the estimation of $\tilde{q}$ requires many samples from $q$ which can significantly slow down training. To address this issue, the authors further constructed a specific type of implicit $q$ distributions, which allows sharing randomness between different $q(\boldsymbol{z}_n|\boldsymbol{x}_n)$ distributions and thus reducing the total number of MC samples computed on a mini-batch of data. This trick improves the approximation accuracy by a significant margin as density ratio estimation is accurate when the two distributions are similar to each other.

### 3.2.2 Direct gradient approximation

The recent development of machine learning algorithms, including VI and SG-MCMC, rely on advanced optimisation tools such as stochastic gradient descent

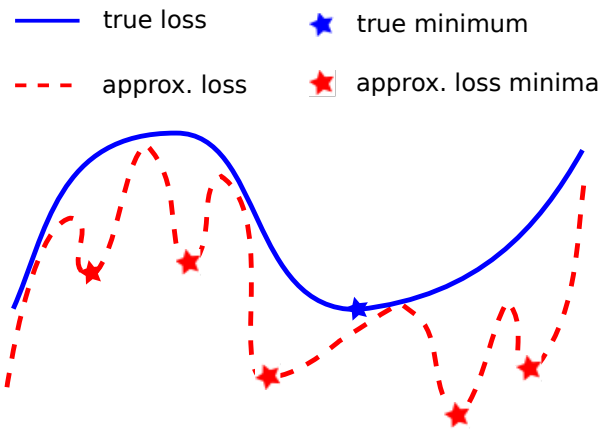


Figure 3: A visualisation of the exact/approximate loss. As one would typically use a deep neural network to help approximate the exact loss function, without careful control of its representation power, the deep net can potentially return a loss function that is accurate at the evaluated points, but has over-complicated shapes in general. It could result in strongly biased gradient updates of the training parameters and bad local optimum.

with adaptive learning rates. Informally the optimisation procedure works as the following: given the current mini-batch of data, we first compute the gradients, then feed them to the optimiser to construct the final update of the training parameters. In the above energy approximation example, this gradient computation is done by first approximating the original objective function $\hat{\mathcal{L}} \approx \mathcal{L}$, then differentiating this approximate energy to obtain an approximate descending direction. However, even when $\hat{\mathcal{L}}$ approximates $\mathcal{L}$ very well at the points from the gradient descent trajectory, the approximate gradient $\nabla_\phi \hat{\mathcal{L}}$ can still be a poor estimator for the exact gradient $\nabla_\phi \mathcal{L}$. We depict this phenomenon in Figure 3.

**Remark** (stabilising GANs by regularising discriminators)**.** To me, another interesting explanation for why WGAN [Arjovsky et al., 2017] and WGAN-GP [Gulrajani et al., 2017] work is that the power of the discriminator (or the test function) is constrained. More specifically the discriminator in WGAN is restricted to be Lipschitz. On the other hand, in the original GAN case, over-fitting frequently happens, particularly at the beginning of training, as neural network classifiers can easily fit to almost any data, even for that with random labels as claimed by Zhang et al. [2017]. Since smoothness is typically lost when over-fitting appears, it leads to poor approximations to the actual gradient and instability during training. Very interestingly, Kodali et al. [2017] showed that similar smoothness constraint applied to the original GAN algorithm also improves stability, which again backed up the observation here.

We see that the energy approximation approach can be problematic if not done in a correct way, therefore a *direct gradient approximation* to the exact gradient might be preferred. To see how the gradient approximation idea applies to the VI case, consider the gradient $\nabla_\phi \mathcal{L}_{\text{VI}}$ using the reparameterisation trick (also see

§ 2.2.1 and § 2.2.2):

$$\nabla_{\boldsymbol{\phi}}\mathcal{L}_{\text{VI}} = \nabla_{\boldsymbol{\phi}}\mathbb{E}_{\pi(\boldsymbol{\epsilon})}[(\nabla_{\mathbf{f}}\log p(\boldsymbol{x}, \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x})) - \nabla_{\mathbf{f}}\log q_{\boldsymbol{\phi}}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x})|\boldsymbol{x}))\nabla_{\boldsymbol{\phi}}\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x})]. \quad (43)$$

Therefore to perform gradient based optimisation, it remains to approximate $\nabla_{\boldsymbol{z}}\log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, as $\nabla_{\boldsymbol{z}}\log p(\boldsymbol{x}, \boldsymbol{z})$ and $\nabla_{\boldsymbol{\phi}}\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x})$ are tractable by assumption. However traditional methods, e.g. Stone [1985], Zhou and Wolfe [2000], Ruppert and Wand [1994], Fan and Gijbels [1996], De Brabanter et al. [2013], do not apply because they require at least a noisy version of $\nabla_{\boldsymbol{z}}\log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, which is intractable in our case. Instead, we will discuss three gradient approximation methods based on kernel methods as follows.

- KDE plug-in estimator.
  A naive idea would again fit another approximation $\hat{q}$ using samples from $q$, then use $\nabla_z \log \hat{q}(\boldsymbol{z}|\boldsymbol{x})$ to approximate the gradient. In kernel methods context, Singh [1977] applied a kernel estimator directly to the first and higher order derivatives. However this method still approximates the target gradient function in an indirect way, and depending on the bandwidth selection, the fitted KDE density can be less smooth or too smooth, making the gradient approximation error high.

- Score matching gradient estimator.
  As motivated, it is preferred to directly minimising the approximation error of the gradient function. Here Sasaki et al. [2014], Strathmann et al. [2015] considered the $\ell_2$ error between the true gradient $\nabla_{\boldsymbol{z}}\log q(\boldsymbol{z}|\boldsymbol{x})$ and the approximation $\hat{\boldsymbol{g}}(\boldsymbol{z}) = (\hat{g}_1(\boldsymbol{z}), ..., \hat{g}_d(\boldsymbol{z}))^{\mathrm{T}}$:

  $$\mathcal{F}(\hat{\boldsymbol{g}}) = \mathbb{E}_q\left[||\hat{\boldsymbol{g}}(\boldsymbol{z}) - \nabla_{\boldsymbol{z}}\log q(\boldsymbol{z}|\boldsymbol{x})||_2^2\right]. \quad (44)$$

  Although the $\ell_2$ error still contains the exact gradient $\nabla_{\boldsymbol{z}}\log q(\boldsymbol{z}|\boldsymbol{x})$, Hyvärinen [2005] showed that by using *integration by parts* and assuming the *boundary condition* $\lim_{\boldsymbol{z}\to\infty}\hat{\boldsymbol{g}}(\boldsymbol{z})q(\boldsymbol{z}|\boldsymbol{x}) = \mathbf{0}$, the $\ell_2$ error can be rewritten as

  $$\mathcal{F}(\hat{\boldsymbol{g}}) = \mathbb{E}_q\left[||\hat{\boldsymbol{g}}(\boldsymbol{z})||_2^2 + 2\langle\nabla, \hat{\boldsymbol{g}}(\boldsymbol{z})\rangle\right], \quad \langle\nabla, \hat{\boldsymbol{g}}(\boldsymbol{z})\rangle = \sum_{i=1}^{d}\nabla_{\boldsymbol{z}_i}\hat{g}_i(\boldsymbol{z}). \quad (45)$$

  The above loss is also referred as the *score matching* objective, and therefore the optimum of $\hat{\boldsymbol{g}}$ is also called the score matching gradient estimator. Since (45) requires computing the gradient of $\hat{\boldsymbol{g}}$, Sasaki et al. [2014], Strathmann et al. [2015] designed a *parametric model*

  $$\hat{\boldsymbol{g}}(\boldsymbol{z}) = \sum_{k=1}^{K} a_k\nabla_{\boldsymbol{z}}\mathcal{K}(\boldsymbol{z}, \boldsymbol{z}^k), \quad \boldsymbol{z}^k \sim q(\boldsymbol{z}|\boldsymbol{x}),$$

  and proposed fitting the linear coefficients $a_k$ by minimising (45).

- Stein gradient estimator.
  Using the same trick as to derive (45) Stein's identity can also be derived.

Given a *test function* $\mathbf{h}(\boldsymbol{z}) : \mathbb{R}^d \to \mathbb{R}^{d'}$ and assuming the *boundary condition* again $\lim_{\boldsymbol{z}\to\infty} \mathbf{h}(\boldsymbol{z})q(\boldsymbol{z}|\boldsymbol{x}) = \mathbf{0}$, Stein's identity [Stein, 1981, Gorham and Mackey, 2015, Liu et al., 2016] is the following:

$$\mathbb{E}_q \left[\mathbf{h}(z)\nabla_{\boldsymbol{z}} \log q(\boldsymbol{z}|\boldsymbol{x})^{\mathrm{T}} + \nabla_{\boldsymbol{z}}\mathbf{h}(\boldsymbol{z})\right] = \mathbf{0}. \tag{46}$$

Observing this, Li and Turner [2017] proposed inverting Stein's identity to obtain an estimator of $\nabla_{\boldsymbol{z}} \log q(\boldsymbol{z}|\boldsymbol{x})$. They first approximated (46) with Monte Carlo, and then performed Ridge regression to obtain a *non-parametric* estimate of the gradient (by noticing that the MC approximation to (46) is linear in $\nabla_{z^k} \log q(\boldsymbol{z}^k|\boldsymbol{x})$).

**Remark** (denoising auto-encoder as a score function estimator)**.** It has been shown in [Särelä and Valpola, 2005, Alain and Bengio, 2014] that denoising auto-encoders (DAEs) [Vincent et al., 2008], once trained, can be used to compute the score function approximately. Briefly speaking, a DAE learns to reconstruct a datum $\boldsymbol{x}$ from a corrupted input $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \sigma\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ by minimising the mean square error. Then the optimal DAE can be used to approximate the score function as $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) \approx \frac{1}{\sigma^2}(\mathrm{DAE}^*(\boldsymbol{x}) - \boldsymbol{x})$. Sonderby et al. [2017] deployed this idea to train an implicit model for image super-resolutions, providing some promising results in some metrics. However applying similar ideas to variational inference can be very expensive, because the estimation of $\nabla_{\boldsymbol{z}} \log q(\boldsymbol{z}|\boldsymbol{x})$ is a sub-routine for VI which is repeatedly required.

### 3.2.3 Alternative optimisation objectives

In variational inference, the KL-divergence $\mathrm{KL}[q||p]$ is minimised to obtain the approximate posterior. In general, the KL-divergence minimisation can be replaced by other optimisation-based approximation methods, as long as with the guarantee of recovering the exact posterior if $\Omega$ contains it. However simply replacing the objective with some other $f$-divergence will not make the optimisation easier as $q$ has an intractable density. Variational techniques for estimating $f$-divergence [Nguyen et al., 2007, 2010] do not apply either, as the exact posterior is difficult to sample.

One promising direction is to replace the KL divergence with Stein discrepancy [Stein, 1972, Barbour, 1988, Gorham and Mackey, 2015], which has a special form that does not require evaluating $q$ nor sampling from $p$. Briefly speaking, Stein discrepancy involves a linear functional operator $\mathbf{O}$, called Stein operator, on a set of test functions $\mathcal{H} = \{h(\boldsymbol{z})\}$ such that $\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[(\mathbf{O}h)(\boldsymbol{z})] = 0$ for $\forall h \in \mathcal{H}$. Then the associated *Stein discrepancy* is defined as $\mathcal{S}(q, p) = \sup_{h\in\mathcal{H}} \mathbb{E}_q[(\mathbf{O}h)(\boldsymbol{z})]$. For continuous density functions, a generic Stein operator is $(\mathbf{O}h)(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}, \boldsymbol{x})^{\mathrm{T}} h(\boldsymbol{z}) + \langle \nabla, h(\boldsymbol{z})\rangle$, for which $\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[(\mathbf{O}h)(\boldsymbol{z})] = 0$, called Stein's identity [Stein, 1972, 1981], can be easily verified using integration by parts. Putting them together, we have the Stein discrepancy

$$\mathcal{S}^2(q, p) = \sup_{h\in\mathcal{H}} \left(\mathbb{E}_q[\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}, \boldsymbol{x})^{\mathrm{T}} h(\boldsymbol{z}) + \langle \nabla, h(\boldsymbol{z})\rangle]\right)^2, \tag{47}$$

which only requires samples from $q$ and the score function $\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}, \boldsymbol{x})$ thus indeed tractable.

Very recently Stein's method has been introduced to the approximate inference community. Ranganath et al. [2016] defined $\mathcal{H}$ as parametric functions represented by neural networks, and approximate the minimax optimisation with gradient descent in an analogous way to GAN training [Goodfellow et al., 2014]. In contrast, analytic solution of the supremum in (47) exists if $\mathcal{H}$ is defined as the unit ball in an RKHS, where Liu et al. [2016] and Chwialkowski et al. [2016] termed the corresponding measure as the kernelised Stein discrepancy (KSD). Liu and Feng [2016] further developed an approximate inference algorithm by directly minimising the KSD between the exact and approximate posterior distributions.

### 3.2.4 Amortising dynamics

MCMC and particle-based approximate inference methods [Dai et al., 2015, Liu and Wang, 2016], though very accurate, become inefficient when inference from multiple different distributions is repeatedly required. As an example consider learning a (deep) generative model, where fast (approximate) marginalisation of latent variables is desirable. Here we consider amortised inference to learn an inference network to mimic a selected stochastic dynamics. More precisely, the algorithm works in three steps:

1. We sample $\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x})$;
2. We improve the sample $\boldsymbol{z}$ to $\boldsymbol{z}_T$ by running $T$-step stochastic/deterministic dynamics;
3. We use the improved samples $\boldsymbol{z}_T$ as targets to improve $q(\boldsymbol{z}|\boldsymbol{x})$.

Li et al. [2017] considered an MCMC sampler as such a dynamics that we want to amortise. The theoretical intuition behind this approach is illustrated in Figure 4. Since the MCMC "oracle" always improves the sample quality in terms of approximating the target distribution,[14] by following the MCMC dynamics, the $q$ distribution will also get improved, until the stage when $\boldsymbol{z}_T$ has the same distribution as $\boldsymbol{z}$ which means $q = p$. Similar intuition also applies to other deterministic dynamics as long as they generate particles that are always approaching to the target distribution. For example, Wang and Liu [2016] used this idea to amortised a deterministic dynamics called Stein variational gradient descent (SVGD). The "catch-up" step for $q(\boldsymbol{z}|\boldsymbol{x})$ can be defined as

$$\boldsymbol{\phi}^{\text{new}} = \arg\min_{\boldsymbol{\phi}} \mathrm{D}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||q_T(\boldsymbol{z}|\boldsymbol{x})], \quad q_T(\boldsymbol{z}|\boldsymbol{x}) = \int \mathcal{T}_T(\boldsymbol{z}|\boldsymbol{z}')q_{\boldsymbol{\phi}}(\boldsymbol{z}'|\boldsymbol{x})d\boldsymbol{z}', \quad (48)$$

where D denotes any divergence/discrepancy/distance and $q_T$ is fixed as the target and does not get differentiated through. In practice only one gradient step is performed, i.e.

$$\boldsymbol{\phi}^{\text{new}} = \boldsymbol{\phi} - \eta \nabla_{\boldsymbol{\phi}} \mathrm{D}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||q_T(\boldsymbol{z}|\boldsymbol{x})]. \quad (49)$$

[14] We have $\mathrm{KL}[q_t||p] \geq \mathrm{KL}[q_{t+1}||p]$ iff $q_t \to p$ for any $q_0 = q$ [Cover and Thomas, 1991].

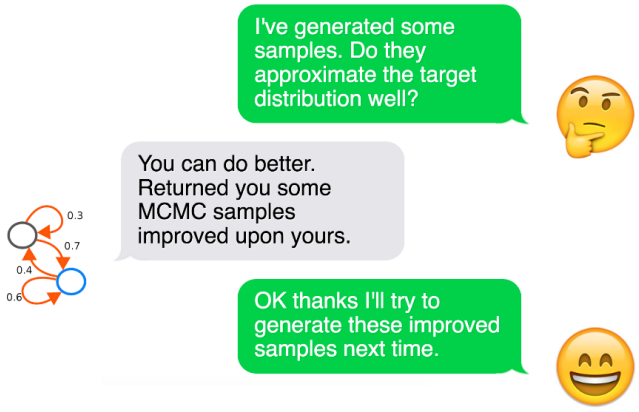


Figure 4: A cartoon illustration of the amortised MCMC idea in Li et al. [2017].

Liu and Wang [2016] used $\ell_2$ distance in sample space $||\boldsymbol{z} - \boldsymbol{z}_T||_2^2$ and therefore the "catch-up" step is defined by deliberately chaining the gradients $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \eta \mathbb{E}_q[\nabla_{\boldsymbol{\phi}} \boldsymbol{z}(\boldsymbol{z}_T - \boldsymbol{z})]$. However for stochastic dynamics the $\ell_2$ distance in sample space does not work well, and instead Li et al. [2017] proposed using any GAN idea to match the $q$ distribution (as fake data) to the target $q_T$ (as real data).

### 3.2.5 Other approaches

This section contains my notes on some other recent papers that fits an implicit $q$ distribution to the posterior. Readers are very welcomed to suggest related papers.

- **Reparameterised MCMC**
  Titsias [2017] proposed a hybrid approach combining invertible transformations and MCMC to construct an implicit approximate posterior. The key observation is that, MCMC algorithms often converge much faster on simpler distributions such as Gaussians. Therefore, if there exists an invertible mapping $\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x})$ which transforms the complicated exact posterior $p(\boldsymbol{z}|\boldsymbol{x})$ to a considerably simpler distribution $p(\boldsymbol{\epsilon}|\boldsymbol{x}) = p(\mathbf{f}(\boldsymbol{\epsilon}, \boldsymbol{x})|\boldsymbol{x})|\nabla_{\boldsymbol{\epsilon}}\mathbf{f}|$, then one can perform MCMC to obtain $\boldsymbol{\epsilon}^k \sim p(\boldsymbol{\epsilon}|\boldsymbol{x})$ (approximately) then map them back to $\boldsymbol{z}^k = \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x}) \sim p(\boldsymbol{z}|\boldsymbol{x})$ (approximately).

  Now Let us define $q(\boldsymbol{z}|\boldsymbol{x}) = \pi(\boldsymbol{\epsilon})|\nabla_{\boldsymbol{\epsilon}=\mathbf{f}^{-1}(\boldsymbol{z},\boldsymbol{x})}\mathbf{f}|^{-1}$, then the variational lower-bound with $q$

$$\mathcal{L}_{\text{VI}}(q) = \mathbb{E}_q\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})|\nabla_{\boldsymbol{\epsilon}=\mathbf{f}^{-1}(\boldsymbol{z},\boldsymbol{x})}\mathbf{f}|}{\pi(\mathbf{f}^{-1}(\boldsymbol{z}, \boldsymbol{x}))}\right]. \tag{50}$$

  Using the LOTUS trick discussed in § 2.2.1, the variational lower-bound can be reformulated:

$$\mathcal{L}_{\text{VI}}(q) = \mathbb{E}_\pi\left[\log \frac{p(\boldsymbol{x}, \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x}))|\nabla_{\boldsymbol{\epsilon}}\mathbf{f}|}{\pi(\boldsymbol{\epsilon})}\right]. \tag{51}$$

  Here we notice that the variational parameters only appears in the numerator, so that the actual optimisation objective for $\boldsymbol{\phi}$ is

$$\mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}_\pi\left[\log p(\boldsymbol{x}, \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x}))|\nabla_{\boldsymbol{\epsilon}}\mathbf{f}|\right]. \tag{52}$$

which can be further approximated by Monte Carlo:

$$\mathcal{L}(\boldsymbol{\phi}) = \frac{1}{K}\sum_{k=1}^{K} \log p(\boldsymbol{x}, \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^k, \boldsymbol{x}))|\nabla_{\boldsymbol{\epsilon}^k}\mathbf{f}|, \quad \boldsymbol{\epsilon}^k \sim \pi(\boldsymbol{\epsilon}). \tag{53}$$

In short, the variational parameter $\boldsymbol{\phi}$ is now transformed to be the "model parameter" of the "model" $p(\boldsymbol{\epsilon}|\boldsymbol{x})$ in $\boldsymbol{\epsilon}$ space. Therefore, the optimisation of $\boldsymbol{\phi}$ becomes a "hyper-parameter optimisation" problem, and observing this, Titsias [2017] proposed a clever idea that is based on MCMC-EM. In E-step, it uses $\pi(\boldsymbol{\epsilon}) = p_T(\boldsymbol{\epsilon}|\boldsymbol{x})$ that is the marginal distribution of $T$-step HMC on $p(\boldsymbol{\epsilon}|\boldsymbol{x}) = p(\mathbf{f}(\boldsymbol{\epsilon}, \boldsymbol{x})|\boldsymbol{x})|\nabla_{\boldsymbol{\epsilon}}\mathbf{f}|$. and in M-step the objective (53) is optimised.

# References

Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian posterior sampling via stochastic gradient fisher scoring. *arXiv preprint arXiv:1206.6380*, 2012.

Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, 2014.

Martin Arjovsky, Soumith Chintala, and Leon Bottou. Wasserstein gan. In *ICML*, 2017.

Hagai Attias. Inferring parameters and structure of latent variable models by variational bayes. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 21–30. Morgan Kaufmann Publishers Inc., 1999.

Hagai Attias. A variational baysian framework for graphical models. In *Advances in neural information processing systems*, pages 209–215, 2000.

Andrew D Barbour. Stein's method and poisson process convergence. *Journal of Applied Probability*, pages 175–184, 1988.

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *arXiv preprint arXiv:1502.05767*, 2015.

Thomas Bayes and Richard Price. An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfrs. *Philosophical Transactions of the Royal Society of London*, 53(0):370–418, 1763.

Matthew James Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, University of London, 2003.

Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

Hans Bethe. Statistical theory of superlattices. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 150(871):552–575, 1935.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

David Blackwell. Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, pages 105–110, 1947.

Phelim P Boyle. Options: A monte carlo approach. *Journal of financial economics*, 4(3):323–338, 1977.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations (ICLR)*, 2016.

Kacper Chwialkowski, Heiko Strathmann, and Arthur Gretton. A kernel test of goodness of fit. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2606–2615, 2016.

Julien Cornebise. *Adaptive Sequential Monte Carlo Methods*. PhD thesis, University Pierre and Marie Curie?Paris 6, 2009.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 1991.

Bo Dai, Niao He, Hanjun Dai, and Le Song. Provable bayesian inference via particle mirror descent. *arXiv preprint arXiv:1506.03101*, 2015.

Kris De Brabanter, Jos De Brabanter, Bart De Moor, and Irène Gijbels. Derivative estimation with local polynomial fitting. *The Journal of Machine Learning Research*, 14(1):281–301, 2013.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in neural information processing systems*, pages 3203–3211, 2014.

Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216 – 222, 1987.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Jianqing Fan and Irène Gijbels. *Local polynomial modelling and its applications*. Chapman & Hall, 1996.

Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall, 1995.

Andrew Gelman, Aki Vehtari, Pasi Jylänki, Christian Robert, Nicolas Chopin, and John P. Cunningham. Expectation propagation as a way of life. *arXiv:1412.4869*, 2014.

John Geweke. Bayesian inference in econometric models using monte carlo integration. *Econometrica: Journal of the Econometric Society*, pages 1317–1339, 1989.

Zoubin Ghahramani. Factorial learning and the em algorithm. In *Advances in neural information processing systems*, pages 617–624, 1995.

Zoubin Ghahramani and Carl E Rasmussen. Bayesian monte carlo. In *Advances in neural information processing systems*, pages 505–512, 2003.

Mark Girolami and Ben Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

Peter W Glynn. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84, 1990.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

Jackson Gorham and Lester Mackey. Measuring sample quality with stein's method. In *Advances in Neural Information Processing Systems*, pages 226–234, 2015.

Shixiang Gu, Zoubin Ghahramani, and Richard E Turner. Neural adaptive sequential monte carlo. In *NIPS*, 2015.

Shixiang Gu, Sergey Levine, Ilya Sutskever, and Andriy Mnih. Muprop: Unbiased backpropagation for stochastic neural networks. In *ICLR*, 2016.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

John M Hammersley and DC Handscomb. *Monte Carlo methods*. Methuen, 1964.

Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *COLT*, pages 5–13. ACM, 1993.

Ferenc Huszár. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*, 2017.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(Apr):695–709, 2005.

Tommi S Jaakkola and Michael I Jordan. Improving the mean field approximation via the use of mixture distributions. *NATO ASI SERIES D BEHAVIOURAL AND SOCIAL SCIENCES*, 89:163–174, 1998.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Theofanis Karaletsos. Adversarial message passing for graphical models. *arXiv preprint arXiv:1612.05048*, 2016.

MC Kennedy and A O'Hagan. Iterative rescaling for bayesian quadrature. *Bayesian Statistics*, 5:639–645, 1996.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *NIPS*, 2015.

Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. How to train your dragan. *arXiv preprint arXiv:1705.07215*, 2017.

Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1):1–6, 1998.

Andrei Nikolaevich Kolmogorov. Unbiased estimates. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, 14(4):303–326, 1950.

Solomon Kullback. *Information theory and statistics*. John Wiley & Sons, 1959.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Pierre Simon Laplace. *Théorie analytique des probabilités*. Courcier, 1820.

Neil D Lawrence, Christopher M Bishop, and Michael I Jordan. Mixture representations for inference and learning in boltzmann machines. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 320–327. Morgan Kaufmann Publishers Inc., 1998.

Tuan Anh Le, Maximilian Igl, Tom Jin, Tom Rainforth, and Frank Wood. Auto-encoding sequential monte carlo. *arXiv preprint arXiv:1705.10306*, 2017.

Nicolas Le Roux, Mark Schmidt, and Francis R Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in Neural Information Processing Systems*, pages 2663–2671, 2012.

Yingzhen Li and Qiang Liu. Wild variational approximations. *NIPS 2016 approximate inference workshop*, 2016.

Yingzhen Li and Richard E Turner. Gradient estimators for implicit models. *arXiv preprint arXiv:1705.07107*, 2017.

Yingzhen Li, Richard E Turner, and Qiang Liu. Approximate inference with amortised mcmc. *arXiv preprint arXiv:1702.08343*, 2017.

Qiang Liu and Yihao Feng. Two methods for wild variational inference. *arXiv preprint arXiv:1612.00081*, 2016.

Qiang Liu and Jason Lee. Black-box importance sampling. In *Artificial Intelligence and Statistics*, pages 952–961, 2017.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2370–2378, 2016.

Qiang Liu, Jason D Lee, and Michael I Jordan. A kernelized stein discrepancy for goodness-of-fit tests and model evaluation. In *ICML*, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Chris J Maddison, Dieterich Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Whye Teh. Filtering variational objectives. *arXiv preprint arXiv:1705.09279*, 2017a.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017b.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *arXiv preprint arXiv:1701.04722*, 2017.

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1791–1799, 2014.

Christian A Naesseth, Scott W Linderman, Rajesh Ranganath, and David M Blei. Variational sequential monte carlo. *arXiv preprint arXiv:1705.11140*, 2017.

Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2:113–162, 2011.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In *NIPS*, 2007.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*, 2016.

Chris J Oates, Mark Girolami, and Nicolas Chopin. Control functionals for monte carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):695–718, 2017.

Anthony O'Hagan. Bayes–hermite quadrature. *Journal of statistical planning and inference*, 29(3):245–260, 1991.

Manfred Opper and Cédric Archambeau. The variational gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.

Brooks Paige and Frank Wood. Inference networks for sequential monte carlo in graphical models. In *International Conference on Machine Learning*, pages 3040–3049, 2016.

John Paisley, David Blei, and Michael Jordan. Variational Bayesian inference with stochastic search. In *Proceedings of The 29th International Conference on Machine Learning (ICML)*, 2012.

Giorgio Parisi. *Statistical field theory*. Addison-Wesley, 1988.

Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *The Second National Conference on Artificial Intelligence (AAAI-82)*, 1982.

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

Jing Qin. Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika*, 85(3):619–630, 1998.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *AISTATS*, pages 814–822, 2014.

Rajesh Ranganath, Jaan Altosaar, Dustin Tran, and David M. Blei. Operator variational inference. In *NIPS*, 2016.

Calyampudi Radhakrishna Rao. *Linear statistical inference and its applications*. Wiley New York, 1965.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of The 30th International Conference on Machine Learning (ICML)*, 2014.

Geoffrey Roeder, Yuhuai Wu, and David Duvenaud. Sticking the landing: An asymptotically zero-variance gradient estimator for variational inference. *arXiv preprint arXiv:1703.09194*, 2017.

Francisco JR Ruiz, Michalis K Titsias, and David M Blei. The generalized reparameterization gradient. In *NIPS*, 2016.

David Ruppert and Matthew P Wand. Multivariate locally weighted least squares regression. *The annals of statistics*, pages 1346–1370, 1994.

Tim Salimans and David A Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

Jaakko Särelä and Harri Valpola. Denoising source separation. *Journal of machine learning research*, 6(Mar):233–272, 2005.

Hiroaki Sasaki, Aapo Hyvärinen, and Masashi Sugiyama. Clustering via mode seeking by direct estimation of the gradient of a log-density. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 19–34. Springer, 2014.

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

Lawrence K Saul and Michael I Jordan. Exploiting tractable substructures in intractable networks. In *Advances in neural information processing systems*, pages 486–492, 1996.

Lawrence K Saul, Tommi Jaakkola, and Michael I Jordan. Mean field theory for sigmoid belief networks. *Journal of artificial intelligence research*, 4:61–76, 1996.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*, 2013.

Jiaxin Shi, Shengyang Sun, and Jun Zhu. Implicit variational inference with kernel density ratio fitting. *arXiv preprint arXiv:1705.10119*, 2017.

Radhey S Singh. Improvement on some known nonparametric uniformly consistent estimators of derivatives of a density. *The Annals of Statistics*, pages 394–399, 1977.

Casper Kaae Sonderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. In *ICLR*, 2017.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Charles Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*, pages 583–602, 1972.

Charles M Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981.

Charles J Stone. Additive regression and other nonparametric models. *The annals of Statistics*, pages 689–705, 1985.

Heiko Strathmann, Dino Sejdinovic, Samuel Livingstone, Zoltan Szabo, and Arthur Gretton. Gradient-free hamiltonian monte carlo with efficient kernel exponential families. In *Advances in Neural Information Processing Systems*, pages 955–963, 2015.

Masashi Sugiyama, Takafumi Kanamori, Taiji Suzuki, Shohei Hido, Jun Sese, Ichiro Takeuchi, and Liwei Wang. A density-ratio framework for statistical data processing. *Information and Media Technologies*, 4(4):962–987, 2009.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044, 2012.

T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

Michalis Titsias and Miguel Lázaro-Gredilla. Local expectation gradients for black box variational inference. In *Advances in neural information processing systems*, pages 2638–2646, 2015.

Michalis K Titsias. Learning model reparametrizations: Implicit variational inference by fitting mcmc distributions. *arXiv preprint arXiv:1708.01529*, 2017.

Dustin Tran, Rajesh Ranganath, and David M Blei. Deep and hierarchical implicit models. *arXiv preprint arXiv:1702.08896*, 2017.

George Tucker, Andriy Mnih, Chris J Maddison, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *arXiv preprint arXiv:1703.07370*, 2017.

R. E. Turner and M. Sahani. Two problems with variational expectation maximisation for time-series models. In D. Barber, T. Cemgil, and S. Chiappa, editors, *Bayesian Time series models*, chapter 5, pages 109–130. Cambridge University Press, 2011.

Jarkko Venna and Samuel Kaski. Visualizing high-dimensional posterior distributions in bayesian modeling. In *Artificial Neural Networks and Neural Information Processing-Supplementary proceedings ICANN/ICONIP 2003*. Citeseer, 2003.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

Dilin Wang and Qiang Liu. Learning to draw samples: With application to amortized mle for generative adversarial learning. *arXiv preprint arXiv:1611.01722*, 2016.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

David Wingate and Theophane Weber. Automated variational inference in probabilistic programming. *arXiv preprint arXiv:1301.1299*, 2013.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Shanggang Zhou and Douglas A Wolfe. On derivative estimation in spline regression. *Statistica Sinica*, pages 93–108, 2000.