Black-Box α-Divergence Minimization

José Miguel Hernández-Lobato[1*] JMH@SEAS.HARVARD.EDU
Yingzhen Li[2*] YL494@CAM.AC.UK
Mark Rowland[2] MR504@CAM.AC.UK
Daniel Hernández-Lobato[3] DANIEL.HERNANDEZ@UAM.ES
Thang D. Bui[2] TDB40@CAM.AC.UK
Richard E. Turner[2] RET26@CAM.AC.UK
[1]Harvard University, [2]University of Cambridge, [3]Universidad Autónoma de Madrid, [*]Both authors contributed equally.

Abstract

Black-box alpha (BB-α) is a new approximate inference method based on the minimization of α-divergences. BB-α scales to large datasets because it can be implemented using stochastic gradient descent. BB-α can be applied to complex probabilistic models with little effort since it only requires as input the likelihood function and its gradients. These gradients can be easily obtained using automatic differentiation. By changing the divergence parameter α, the method is able to interpolate between variational Bayes (VB) ($\alpha \rightarrow 0$) and an algorithm similar to expectation propagation (EP) ($\alpha = 1$). Experiments on probit regression and neural network regression and classification problems show that BB-α with non-standard settings of α, such as $\alpha = 0.5$, usually produces better predictions than with $\alpha \rightarrow 0$ (VB) or $\alpha = 1$ (EP).

1. Introduction

Bayesian probabilistic modelling provides useful tools for making predictions from data. The formalism requires a probabilistic model $p(\boldsymbol{x}|\boldsymbol{\theta})$, parameterized by a parameter vector $\boldsymbol{\theta}$, over the observations $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^N$. Bayesian inference treats $\boldsymbol{\theta}$ as a random variable and predictions are then made by averaging with respect to the posterior belief

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto \left[\prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{\theta})\right] p_0(\boldsymbol{\theta}) ,$$

where $p(\boldsymbol{x}_n|\boldsymbol{\theta})$ is a likelihood factor and $p_0(\boldsymbol{\theta})$ is the prior. Unfortunately the computation of this posterior distribution is often intractable for many useful probabilistic models. One can use approximate inference techniques to sidestep this difficulty. Two examples are variational Bayes (VB) (Jordan et al., 1999) and expectation propagation (EP) (Minka, 2001a). These methods adjust the parameters of a tractable distribution so that it is close to the true posterior, by finding an stationary point of an energy function. Both VB and EP are specific cases of local α-divergence minimization, where the parameter $\alpha \in (-\infty, +\infty) \setminus \{0\}$ specifies the properties of the divergence to be minimized (Minka, 2005). If $\alpha \rightarrow 0$, VB is obtained and $\alpha = 1$ gives EP (Minka, 2005). Power EP (PEP) (Minka, 2004) extends EP to general settings of α, whose optimal value may be model, dataset and/or task specific.

EP can provide better solutions than VB in specific cases. For instance, VB provides poor approximations when non-smooth likelihood functions are used (Cunningham et al., 2011; Turner & Sahani, 2011b; Opper & Winther, 2005). EP also often performs better when factored approximations are employed (Turner & Sahani, 2011a; Minka, 2001b). There are, however, issues that hinder the wide deployment of EP, and by extension of power EP too. First, EP requires to store in memory local approximations of the likelihood factors. This has prohibitive memory cost for large datasets and big models. Second, efficient implementations of EP based on message passing have no convergence guarantees to an stationary point of the energy function (Minka, 2001a).

Previous work has addressed the first issue by using “factor tying”/“local parameter sharing” through the “stochastic EP”/“averaged EP” (SEP/AEP) algorithms (Li et al., 2015; Dehaene & Barthelmé, 2015). However, the energy function of SEP/AEP is unknown, making the attempts for proving convergence of these algorithms difficult. On the second issue, Heskes & Zoeter (2002) and Opper & Winther (2005) derived a convergent double-loop implementation of EP. However, it can be far slower than the

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

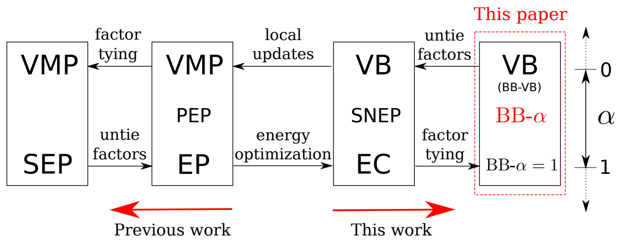

Figure 1. Connections between the proposed black-box alpha (BB-α) algorithm and a number of existing approaches. EC abbreviates expectation consistent approximate inference (Opper & Winther, 2005) which is a double-loop convergent algorithm similar to Heskes & Zoeter (2002). VMP stands for variational message passing described in Minka (2005), which should be distinguished from another version in Winn & Bishop (2005). Other acronyms are explained in the main text.

original message passing procedure. Teh et al. (2015) proposed the stochastic natural-gradient EP (SNEP) method, which is also double-loop-like, but in practice, they only perform one-step inner loop update to speed up training.

Buoyed by the success of factor tying methods (SEP/AEP) , we propose to apply the same idea directly to the power EP energy function, rather than through the power EP message passing updates as in SEP/AEP. We call this new method Black-box alpha (BB-α). Figure 1 illustrates its differences and connections to other existing methods. Besides being memory efficient as SEP/AEP, BB-α has an analytic energy form that does not require double-loop procedures and can be directly optimized using gradient descent. This means that popular stochastic optimization methods can be used for large-scale learning with BB-α.

An advantage of BB-α, that gives origin to its name, is that it is a black-box method that can be straightforward applied to very complicated probabilistic models. In such models, the energy function of VB, EP and power EP does not exist in an analytic form. *Black-box* VB (Ranganath et al., 2014) sidesteps this difficulty by using Monte Carlo approximations of the VB energy, see (Salimans et al., 2013) for a related technique. In this work, we follow a similar approach and also approximate the BB-α energy by Monte Carlo. Similar approximations have already been applied to EP (Barthelmé & Chopin, 2011; Gelman et al., 2014; Xu et al., 2014; Teh et al., 2015). However, these methods do not use the factor tying idea and consequently, are based on expensive double-loop approaches or on message passing algorithms that lack convergence guarantees.

2. α-Divergence and power EP

Let us begin by briefly reviewing the α-divergence upon which our method is based. Consider two probability densities p and q of a random variable $\boldsymbol{\theta}$; one fundamental question is to assess how close the two distributions are.

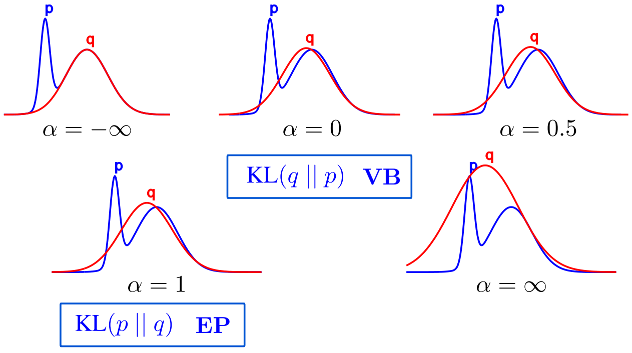

Figure 2. An illustration of approximating distributions by α-divergence minimization. Here p and q shown in the graphs are unnormalized probability densities. Reproduced from Minka (2005). Best viewed in color.

The α-divergence (Amari, 1985) measures the "similarity" between two distributions, and in this paper we adopt a more convenient form[1] (Zhu & Rohwer, 1995):

$$\mathrm{D}_\alpha[p||q] = \frac{1}{\alpha(1-\alpha)}\left(1 - \int p(\boldsymbol{\theta})^\alpha q(\boldsymbol{\theta})^{1-\alpha} d\boldsymbol{\theta}\right) . \quad (1)$$

The following examples with different α values are interesting special cases:

$$\mathrm{D}_1[p||q] = \lim_{\alpha\to 1} \mathrm{D}_\alpha[p||q] = \mathrm{KL}[p||q] \,, \quad (2)$$

$$\mathrm{D}_0[p||q] = \lim_{\alpha\to 0} \mathrm{D}_\alpha[p||q] = \mathrm{KL}[q||p] \,, \quad (3)$$

$$\mathrm{D}_{\frac{1}{2}}[p||q] = 2\int \left(\sqrt{p(\boldsymbol{\theta})} - \sqrt{q(\boldsymbol{\theta})}\right)^2 d\boldsymbol{\theta} = 4\mathrm{Hel}^2[p||q] \,. \quad (4)$$

For the first two limiting cases $\mathrm{KL}[p||q]$ denotes the *Kullback-Leibler (KL) divergence* given by $\mathrm{KL}[p||q] = \mathbf{E}_p[\log p(\boldsymbol{\theta}) - \log q(\boldsymbol{\theta})]$. In (4) $\mathrm{Hel}[p||q]$ denotes the *Hellinger distance* between two distributions; $\mathrm{D}_{\frac{1}{2}}$ is the only member of the family of α-divergences that is symmetric in p and q.

To understand how the choice of α might affect the result of approximate inference, consider the problem of approximating a complicated distribution p with a tractable Gaussian distribution q by minimizing $\mathrm{D}_\alpha[p||q]$. The resulting (unnormalized) approximations obtained for different values of α are visualized in Figure 2. This shows that when α is a large positive number the approximation q tends to cover all the modes of p, while for $\alpha \to -\infty$ (assuming the divergence is finite) q is attracted to the mode with the largest probability mass. The optimal setting of α might reasonably be expected to depend on the learning task that is being considered.

[1] Equivalent to the original definition by setting $\alpha' = 2\alpha - 1$ in Amari's notation.

Setting aside the analytic tractability of the computations, we note that the minimization of a global α-divergence might not always be desirable. If the true posterior has many modes, a global approximation of this flavor that is refined using $\alpha \geq 1$ will cover the modes, and can place substantial probability in the area where the true posterior has low probability (see the last plot in Figure 2). The power EP algorithm (Minka, 2001a; 2004) minimizes instead a set of local α-divergences. We now give a brief review of the power EP algorithm. Recall the definition of the typically intractable posterior distribution

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto \left[\prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{\theta})\right] p_0(\boldsymbol{\theta}) \,. \tag{5}$$

Here for simplicity, the prior distribution $p_0(\boldsymbol{\theta}) = \exp\{\mathbf{s}(\boldsymbol{\theta})^T \boldsymbol{\lambda}_0 - \log Z(\boldsymbol{\lambda}_0)\}$ is assumed to have an exponential family form, where $\boldsymbol{\lambda}_0$ and $\mathbf{s}(\boldsymbol{\theta})$ are vectors of natural parameters and sufficient statistics, respectively, and $Z(\boldsymbol{\lambda}_0)$ is the normalization constant or partition function required to make $p_0(\boldsymbol{\theta})$ a valid distribution. We can use power EP to approximate the true posterior $p(\boldsymbol{\theta}|\mathcal{D})$. We define the site approximation $f_n(\boldsymbol{\theta}) = \exp\{\mathbf{s}(\boldsymbol{\theta})^T \boldsymbol{\lambda}_n\}$, which is is within the same exponential family as the prior, and is used to approximate the effect of the n-th likelihood factor $p(\boldsymbol{x}_n|\boldsymbol{\theta})$. The approximate posterior is then defined as the product of all f_n and the prior: $q(\boldsymbol{\theta}) \propto \exp\{\mathbf{s}(\boldsymbol{\theta})^T(\sum_n \boldsymbol{\lambda}_n + \boldsymbol{\lambda}_0)\}$. In the following sections we use $\boldsymbol{\lambda}_q$ to denote the natural parameters of $q(\boldsymbol{\theta})$, and in the EP context we define $\boldsymbol{\lambda}_q = \sum_n \boldsymbol{\lambda}_n + \boldsymbol{\lambda}_0$. According to Minka (2004) and Seeger (2005), the power EP energy function with power α is

$$\begin{aligned} E(\boldsymbol{\lambda}_0, \{\boldsymbol{\lambda}_n\}) &= \log Z(\boldsymbol{\lambda}_0) + \left(\frac{N}{\alpha} - 1\right) \log Z(\boldsymbol{\lambda}_q) \\ &- \frac{1}{\alpha} \sum_{n=1}^{N} \log \int p(\boldsymbol{x}_n|\boldsymbol{\theta})^{\alpha} \exp\{\mathbf{s}(\boldsymbol{\theta})^T(\boldsymbol{\lambda}_q - \alpha\boldsymbol{\lambda}_n)\} d\boldsymbol{\theta} \,. \end{aligned} \tag{6}$$

This energy is equal to minus the logarithm of the power EP approximation of the model evidence $p(\mathcal{D})$, that is, the normalizer of the right-hand side of (5). Therefore, minimizing (6) with respect to $\{\boldsymbol{\lambda}_n\}$ is arguably a sensible way to tune these variational parameters. However, power EP does not directly perform gradient descent to minimize $E(\boldsymbol{\lambda}_0, \{\boldsymbol{\lambda}_n\})$. Instead, it finds a stationary solution to the optimization problem by running a message passing algorithm that repeatedly applies the following four steps for every site approximation f_n:

1 Compute the cavity distribution:

$$q^{\backslash n}(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta}) / f_n(\boldsymbol{\theta})^{\alpha},$$

i.e. $\boldsymbol{\lambda}^{\backslash n} \leftarrow \boldsymbol{\lambda}_q - \alpha\boldsymbol{\lambda}_n$;

2 Compute the "tilted" distribution by inserting the exact likelihood factor raised to the power α:

$$\tilde{p}_n(\boldsymbol{\theta}) \propto q^{\backslash n}(\boldsymbol{\theta}) p(\boldsymbol{x}_n|\boldsymbol{\theta})^{\alpha};$$

3 Adjust q by matching moments:

$$E_q[\boldsymbol{s}(\boldsymbol{\theta})] \leftarrow E_{\tilde{p}_n}[\boldsymbol{s}(\boldsymbol{\theta})];$$

4 Recover the site approximation $f_n(\boldsymbol{\theta})$ by setting $\boldsymbol{\lambda}_n \leftarrow \boldsymbol{\lambda}_q - \boldsymbol{\lambda}^{\backslash n}$, and compute the final update for $q(\boldsymbol{\theta})$ by $\boldsymbol{\lambda}_q \leftarrow \sum_n \boldsymbol{\lambda}_n + \boldsymbol{\lambda}_0$.

Notice in step 3 moment matching is equivalent to updating the q distribution by minimizing an α-divergence, with the target proportional to $p(\boldsymbol{x}_n|\boldsymbol{\theta}) \exp\{\boldsymbol{s}(\boldsymbol{\theta})^T(\boldsymbol{\lambda}_q - \boldsymbol{\lambda}_n)\}$. To see this, consider approximating some p distribution with q by minimizing the α-divergence; the gradient of $\mathrm{D}_\alpha[p||q]$ w.r.t. the natural parameters $\boldsymbol{\lambda}_q$ is:

$$\begin{aligned} \nabla_{\boldsymbol{\lambda}_q} \mathrm{D}_\alpha[p||q] &= -\frac{1}{\alpha} \int p(\boldsymbol{\theta})^{\alpha} q(\boldsymbol{\theta})^{1-\alpha} \nabla_{\boldsymbol{\lambda}_q} \log q(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \frac{Z_{\tilde{p}}}{\alpha} \left(\mathbf{E}_q[\boldsymbol{s}(\boldsymbol{\theta})] - \mathbf{E}_{\tilde{p}}[\boldsymbol{s}(\boldsymbol{\theta})]\right) \,, \end{aligned} \tag{7}$$

where $\tilde{p} \propto p^{\alpha} q^{1-\alpha}$ with normalization constant $Z_{\tilde{p}}$ and $\alpha \neq 0$. Substituting p with $p'(\boldsymbol{\theta}) \propto p(\boldsymbol{x}_n|\boldsymbol{\theta}) \exp\{\boldsymbol{s}(\boldsymbol{\theta})^T(\boldsymbol{\lambda}_q - \boldsymbol{\lambda}_n)\}$ and zeroing the gradient results in step 3 that matches moments between the tilted distribution and the approximate posterior. Also Minka (2004) derived the stationary condition of (6) as

$$\mathbf{E}_{\tilde{p}_n}[\mathbf{s}(\boldsymbol{\theta})] = \mathbf{E}_q[\mathbf{s}(\boldsymbol{\theta})], \ \forall n \,, \tag{8}$$

so that it agrees with (7). This means, at convergence, $q(\boldsymbol{\theta})$ minimizes the α-divergences from all the tilted distributions to the approximate posterior.

Alternatively, Heskes & Zoeter (2002) and Opper & Winther (2005) proposed a convergent double-loop algorithm to solve the energy minimization problem for normal EP ($\alpha = 1$) (see supplementary material). This algorithm first rewrites the energy (6) as a function of the cavity parameters $\boldsymbol{\lambda}^{\backslash n}$ and adds the constraint $(N-1)\boldsymbol{\lambda}_q + \boldsymbol{\lambda}_0 = \sum_n \boldsymbol{\lambda}^{\backslash n}$ that ensures agreement between the global approximation and the local component approximate factors. It then alternates between an optimization of the cavity parameters in the inner loop and an optimization of the parameters of the posterior approximation $\boldsymbol{\lambda}_q$ in the outer loop. However, this alternating optimization procedure often requires too many iterations to be useful in practice.

3. Approximate local minimization of α-divergences

In this section we introduce *black-box alpha* (BB-α), which approximates power EP with a simplified objective. Now we constrain all the site parameters to be equal, i.e. $\boldsymbol{\lambda}_n = \boldsymbol{\lambda}$ for all n. This is equivalent to tying all the local factors, where now $f_n(\boldsymbol{\theta}) = f(\boldsymbol{\theta})$ for all n. Then all the cavity distributions are the same with natural parameter $\boldsymbol{\lambda}^{\backslash n} = (N - \alpha)\boldsymbol{\lambda} + \boldsymbol{\lambda}_0$, and the approximate posterior is parameterized by $\boldsymbol{\lambda}_q = N\boldsymbol{\lambda} + \boldsymbol{\lambda}_0$. Recall that $f_n(\boldsymbol{\theta})$ captures the contribution of the n-th likelihood to the posterior. Now with shared site parameters we are using an "average site approximation" $f(\boldsymbol{\theta}) = \exp\{\boldsymbol{s}(\boldsymbol{\theta})^T\boldsymbol{\lambda}\}$ that approximates the average effect of each likelihood term on the posterior. Under this assumption we rewrite the energy function (6) by replacing $\boldsymbol{\lambda}_n$ with $\boldsymbol{\lambda}$:

$$\begin{aligned} E(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}) =& \log Z(\boldsymbol{\lambda}_0) - \log Z(\boldsymbol{\lambda}_q) \\ & - \frac{1}{\alpha}\sum_{n=1}^{N} \log \mathbf{E}_q\left[\left(\frac{p(\boldsymbol{x}_n|\boldsymbol{\theta})}{f(\boldsymbol{\theta})}\right)^{\alpha}\right]. \end{aligned} \tag{9}$$

Figure 3 illustrates the comparison between the original power EP and the proposed method. Also, as there is a one-to-one correspondence between $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_q$ given $\boldsymbol{\lambda}_0$, i.e. $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_q - \boldsymbol{\lambda}_0)/N$, we can therefore rewrite (9) as $E(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_q)$ using the global parameter $\boldsymbol{\lambda}_q$.

The factor tying constraint was proposed in Li et al. (2015) and Dehaene & Barthelmé (2015) to obtain versions of EP called *stochastic EP* (SEP) and *averaged EP* (AEP), respectively, thus the new method also inherits the advantage of memory efficiency. However, applying this same idea directly to the energy function (6) results in a different class of algorithms from AEP, SEP and power SEP. The main difference is that, when an exponential family approximation is considered, SEP averages the *natural parameters* of the approximate posteriors obtained in step 3 of the message passing algorithm from the previous section. However, in BB-α the *moments*, also called the *mean parameters*, of the approximate posteriors are averaged and then converted to the corresponding natural parameters[2]. To see this, one can take derivatives of $E(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_q)$ w.r.t. $\boldsymbol{\lambda}_q$ and obtain the new stationary conditions for $\alpha \neq 0$ and $\alpha \neq N$:

$$\mathbf{E}_q[\mathbf{s}(\boldsymbol{\theta})] = \frac{1}{N}\sum_{n=1}^{N} \mathbf{E}_{\tilde{p}_n}[\mathbf{s}(\boldsymbol{\theta})]\,. \tag{10}$$

Therefore, the moments of the q distribution, i.e. the expectation of $\mathbf{s}(\boldsymbol{\theta})$ with respect to $q(\boldsymbol{\theta})$, is equal to the average of the expectation of $\mathbf{s}(\boldsymbol{\theta})$ across the different tilted distributions $\tilde{p}_n(\boldsymbol{\theta}) \propto p(\boldsymbol{x}_n|\boldsymbol{\theta})^{\alpha} q^{\backslash n}(\boldsymbol{\theta})$, for $n = 1, \ldots, N$. It might not always be sensible to average moments, e.g. this approach is shown to be biased even in the simple case where the likelihood terms also belong to the same exponential family as the prior and approximate posterior (see supplementary material). Moreover, BB-α can mode-average, e.g. when approximating the posterior distribution of component means in a Gaussian Mixture Model, which may or may not be desired depending on the application. But unlike SEP, the new method explicitly defines an energy function as an optimization objective, which enables analysis of convergence and applications of stochastic optimization/adaptive learning rate methods. Moreover, it can provide an approximate posterior with better uncertainty estimates than VB (see supplementary), which is desirable. The energy also makes hyper-parameter optimization simple, which is key for many applications.

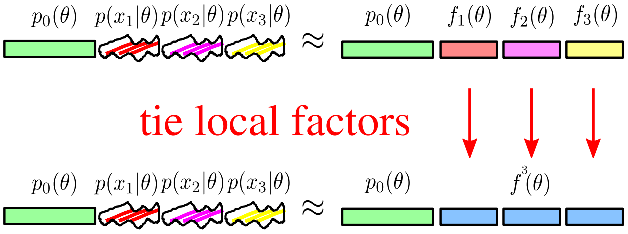

Figure 3. A cartoon for BB-α's factor tying constraint. Here we assume the dataset has $N = 3$ observations. Best seen in color.

We prove the convergence of the new approximate EP method by showing that the new energy function (9) is bounded below for $\alpha \leq N$ when the energy function is finite. First, using Jensen's inequality, we can prove that the third term $-\frac{1}{\alpha}\sum_{n=1}^{N} \log \mathbf{E}_q\left[\left(\frac{p(\boldsymbol{x}_n|\boldsymbol{\theta})}{f(\boldsymbol{\theta})}\right)^{\alpha}\right]$ in (9) is non-increasing in α, because for arbitrary numbers $0 < \alpha < \beta$ or $\beta < \alpha < 0$ the function $x^{\frac{\alpha}{\beta}}$ is strictly concave on $x \geq 0$. For convenience we shorthand this term as $G(\alpha)$. Then the proof is finished by subtracting from (9) $\tilde{G}(N) := -\frac{1}{N}\sum_n \log \int p_0(\boldsymbol{\theta}) p(\boldsymbol{x}_n|\boldsymbol{\theta})^N d\boldsymbol{\theta}$, a function that is independent of the q distribution:

$$\begin{aligned} & E(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_q) - \tilde{G}(N) \\ =& G(\alpha) + \log Z(\boldsymbol{\lambda}_0) - \log Z(\boldsymbol{\lambda}_q) - \tilde{G}(N) \\ =& G(\alpha) - G(N) \geq 0\,. \end{aligned}$$

The stationary point of (6) is expected to converge to the stationary point of (9) when more and more data are available. More precisely, as N grows, we expect $q(\boldsymbol{\theta})$ and the cavities to become very concentrated. When this happens, the contribution of each likelihood factor $p(\boldsymbol{x}_n|\boldsymbol{\theta})$ to the tilted distribution $\tilde{p}_n(\boldsymbol{\theta})$ becomes very small because the likelihood is a rather flat function when compared to the cavity distribution $q^{\backslash n}(\boldsymbol{\theta})$. Therefore, as the amount of data N increases, we expect all the terms $\mathbf{E}_{\tilde{p}_n}[\mathbf{s}(\boldsymbol{\theta})]$ in (10) to be very similar to each other. When all of them are equal, we have that (10) implies (8).

As in power EP, which value of α returns the best approximation depends on the particular application. In the limit

[2] Under a minimal exponential family assumption, there exists a one-to-one correspondence between the natural parameter and the mean parameter.

that α approaches zero, the BB-α energy (9) converges to the variational free energy:

$$\begin{aligned}
&\lim_{\alpha\to 0} E(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_q) \\
&= \log Z(\boldsymbol{\lambda}_0) - \log Z(\boldsymbol{\lambda}_q) - \sum_{n=1}^{N} \mathbf{E}_q \left[\log \frac{p(\boldsymbol{x}_n|\boldsymbol{\theta})}{f(\boldsymbol{\theta})} \right] \\
&= -\mathbf{E}_q \left[\log \frac{\prod_n p(\boldsymbol{x}_n|\boldsymbol{\theta}) p_0(\boldsymbol{\theta})}{\exp\{\boldsymbol{s}(\boldsymbol{\theta})^T \boldsymbol{\lambda}_q\}/Z(\boldsymbol{\lambda}_q)} \right] \\
&= -\mathbf{E}_q \left[\log p(\boldsymbol{\theta}, \mathcal{D}) - \log q(\boldsymbol{\theta}) \right] \\
&= E_{VB}(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_q)\,.
\end{aligned}$$

Note also, the α-divergence as a function of α is smooth in $[0, 1]$. Therefore, by adjusting the α parameter, we can interpolate between the solutions given by variational Bayes and the solutions given by the resulting approximation to EP. In the supplementary material we demonstrate this continuity in α with a linear regression example.

The prior hyper-parameters $\boldsymbol{\lambda}_0$ can be optimally adjusted by minimizing (9). This can be done through a variational EM-like procedure. First we optimize the energy function w.r.t. the q distribution until convergence. Then we take the gradients w.r.t. $\boldsymbol{\lambda}_0$ and equate them to zero to obtain the update for the prior, which is $\mathbf{E}_q[\mathbf{s}(\boldsymbol{\theta})] = \mathbf{E}_{p_0}[\mathbf{s}(\boldsymbol{\theta})]$. However, this procedure is inefficient in practice. Instead we jointly optimize the approximate posterior q and the prior distribution p_0 similar to the approach of Hernández-Lobato & Hernández-Lobato (2016) for normal EP.

3.1. Large scale learning

When N is large, it might be beneficial to minimize (9) using stochastic optimization techniques. In particular, we can uniformly sample a mini-batch of data $\mathbf{S} \subseteq \{1, \ldots, N\}$ and construct the noisy estimate of the energy function given by

$$\begin{aligned}
E(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_q) &\approx \log Z(\boldsymbol{\lambda}_0) - \log Z(\boldsymbol{\lambda}_q) \\
&- \frac{1}{\alpha} \frac{N}{|\mathbf{S}|} \sum_{n\in\mathbf{S}} \log \mathbf{E}_q \left[\left(\frac{p(\boldsymbol{x}_n|\boldsymbol{\theta})}{f(\boldsymbol{\theta})} \right)^{\alpha} \right]\,. \qquad (11)
\end{aligned}$$

The gradients of (11) can then be used to minimize the original objective by stochastic gradient descent. Under mild conditions, as discussed by Bottou (1998), and using a learning rate schedule $\{\gamma_t\}$ that satisfies the Robbins-Monro conditions

$$\sum_{t=1}^{\infty} \gamma_t = \infty, \quad \sum_{t=1}^{\infty} \gamma_t^2 < \infty,$$

the stochastic optimization procedure will converge to a fixed point of the exact energy function (9).

Similar to SEP/AEP (Li et al., 2015; Dehaene & Barthelmé, 2015), BB-α only maintains the global parameter $\boldsymbol{\lambda}_q$ and the prior parameter $\boldsymbol{\lambda}_0$. This has been shown to achieve a significant amount of memory saving. On the other hand, recent work on parallelizing EP (Gelman et al., 2014; Xu et al., 2014; Teh et al., 2015), whether in a synchronous manner or not, extends to BB-α naturally. But unlike EP, which computes different cavity distributions for different data points, BB-α uses the same cavity distribution for each data point.

3.2. Black-box α-divergence minimization

In complicated probabilistic models, we might not be able to analytically compute the expectation over the approximate distribution $\mathbf{E}_q \left[\left(\frac{p(\boldsymbol{x}_n|\boldsymbol{\theta})}{f(\boldsymbol{\theta})} \right)^{\alpha} \right]$ in (11) involving the likelihood factors. However, we can obtain an estimate of these integrals by Monte Carlo. In this work we use the simplest method for doing this, but techniques like SMC and MCMC could also have the potential to be deployed. We draw K samples $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$ from $q(\boldsymbol{\theta})$ and then approximate the integrals by expectations with respect to those samples. This produces the following noisy estimate of the energy function:

$$\begin{aligned}
\hat{E}(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_q) &= \log Z(\boldsymbol{\lambda}_0) - \log Z(\boldsymbol{\lambda}_q) \\
&- \frac{1}{\alpha} \frac{N}{|\mathbf{S}|} \sum_{n\in\mathbf{S}} \log \frac{1}{K} \sum_{k} \left(\frac{p(\boldsymbol{x}_n|\boldsymbol{\theta}_k)}{f(\boldsymbol{\theta}_k)} \right)^{\alpha}\,. \qquad (12)
\end{aligned}$$

Note, however, that the resulting stochastic gradients will be biased because the energy function (12) applies a nonlinear transformation (the logarithm) to the Monte Carlo estimator of the integrals. Nevertheless, this bias can be reduced by increasing the number of samples K. Our experiments indicate that when $K \geq 10$ the bias is almost negligible w.r.t. the variance from sub-sampling the data using minibatches, for the models considered here.

There are two other tricks we have used in our implementation. The first one is the *reparameterization trick* (Kingma & Welling, 2014), which has been used to reduce the variance of the Monte Carlo approximation to the variational free energy. Consider the case of computing expectation $\mathbf{E}_{q(\boldsymbol{\theta})}[F(\boldsymbol{\theta})]$. This expectation can also be computed as $\mathbf{E}_{p(\boldsymbol{\epsilon})}[F(g(\boldsymbol{\epsilon}))]$, if there exists a mapping $g(\cdot)$ and a distribution $p(\boldsymbol{\epsilon})$ such that $\boldsymbol{\theta} = g(\boldsymbol{\epsilon})$ and $q(\boldsymbol{\theta})d\boldsymbol{\theta} = p(\boldsymbol{\epsilon})d\boldsymbol{\epsilon}$. Now consider a Gaussian approximation $q(\boldsymbol{\theta})$ as a running example. Since the Gaussian distribution also has an (minimal) exponential family form, there exists a one-to-one correspondence between the mean parameters $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ and the natural parameters $\boldsymbol{\lambda}_q = \{\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}\}$. Furthermore, sampling $\boldsymbol{\theta} \sim q(\boldsymbol{\theta})$ is equivalent to $\boldsymbol{\theta} = g(\boldsymbol{\epsilon}) = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Thus the sampling approximation $\hat{E}(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_q)$ can be computed as

$$\hat{E}(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_q) = \log Z(\boldsymbol{\lambda}_0) - \log Z(\boldsymbol{\lambda}_q) \\ - \frac{1}{\alpha} \frac{N}{|\mathbf{S}|} \sum_{n \in \mathbf{S}} \log \frac{1}{K} \sum_k \left(\frac{p(\boldsymbol{x}_n | g(\boldsymbol{\epsilon}_k))}{f(g(\boldsymbol{\epsilon}_k))} \right)^\alpha , \tag{13}$$

with $\boldsymbol{\epsilon}_1, ..., \boldsymbol{\epsilon}_K$ sampled from a zero mean, unit variance Gaussian. A further trick is resampling $\{\boldsymbol{\epsilon}_k\}$ every $M > 1$ iterations. In our experiments with neural networks, this speeds-up the training process since it reduces the transference of randomness to the GPU, which slows down computations.

Given a new probabilistic model, one can then use the proposed approach to quickly implement, in an automatic manner, an inference algorithm based on the local minimization of α-divergences. For this one only needs to write code that evaluates the likelihood factors $f_1, \ldots, f_N$ in (13). After this, the most difficult task is the computation of the gradients of (13) so that stochastic gradient descent with minibatches can be used to optimize the energy function. However, the computation of these gradients can be easily automated by using automatic differentiation tools such as Autograd (`http://github.com/HIPS/autograd`) or Theano (Bastien et al., 2012). This approach allows us to quickly implement and test different modeling assumptions with little effort.

4. Experiments

We evaluated the proposed algorithm black-box alpha (BB-α), on regression and classification problems using a probit regression model and Bayesian neural networks. The code for BB-α is publicly available[3] We also compare with a method that optimizes a Monte Carlo approximation to the variational lower bound (Ranganath et al., 2014). This approximation is obtained in a similar way to the one described in Section 3.2, where one can show its equivalence to BB-α by limiting $\alpha \to 0$. We call this method black-box variational Bayes (BB-VB). In the implementation of BB-α and BB-VB shown here, the posterior approximation q is always a factorized Gaussian (but more complex distributions can easily be handled). The mean parameters of q are initialized by independently sampling from a zero mean Gaussian with standard deviation 10^{-1}. We optimize the logarithm of the variance parameters in q to avoid obtaining negative variances. We use -10 as the initial value for the log-variance parameters, which results in very low initial variance in q. This makes the stochastic optimizer initially resemble a point estimator method which quickly finds a solution for the mean parameters with good predictive properties on the training data. After this, the stochastic optimizer progressively increases the variance parameters to capture the uncertainty around the mean of q. This trick considerably improves the performance of the stochastic optimization method for BB-α and BB-VB. The prior $p(\mathbf{x})$ is always taken to be a factorized Gaussian with zero mean and unit standard deviation. The implementation of each of the analyzed methods is available for reproducible research.

4.1. Probit regression

We perform experiments with a Bayesian probit regression model to validate the proposed black-box approach. We optimize the different objective functions using minibatches of size 32 and Adam (Kingma & Ba, 2014) with its default parameter values during 200 epochs. BB-α and BB-VB are implemented by drawing $K = 100$ Monte Carlo samples for each minibatch. The following values $\alpha = 1$, $\alpha = 0.5$ and $\alpha = 10^{-6}$ for BB-α are considered in the experiments. The performance of each method is evaluated on 50 random training and test splits of the data with 90% and 10% of the data instances, respectively.

Table 1 shows the average test log-likelihood and test error obtained by each technique in the probit regression datasets from the UCI data repository (Lichman, 2013). We also show the average rank obtained by each method across all the train/test splits of the data. Overall, all the methods obtain very similar results although BB-α with $\alpha = 1.0$ seems to perform slightly better. Importantly, BB-α with $\alpha = 10^{-6}$ produces the same results as BB-VB, which verifies our theory of continuous interpolations near the limit $\alpha \to 0$.

4.2. Neural network regression

We perform additional experiments with neural networks for regression with 100 units in a single hidden layer with ReLUs and Gaussian additive noise at the output. We consider several regression datasets also from the UCI data repository (Lichman, 2013). We use the same training procedure as before, but using 500 epochs instead of 200. The noise variance is learned in each method by optimizing the corresponding objective function: evidence lower bound in BB-VB and (9) in BB-α.

The average test log-likelihood across 50 splits of the data into training and test sets are visualized in Figure 4. The performance of BB-$\alpha = 10^{-6}$ is again almost indistinguishable from that of BB-VB. None of the tested α settings clearly dominates the other choices in all cases, indicating that the optimal α may vary for different problems. However, $\alpha = 0.5$ produces good overall results. This might be because the Hellinger distance is the only symmetric divergence measure in the α-divergence family which balances the tendencies of capturing a mode ($\alpha <$

[3] `https://bitbucket.org/jmh233/code_black_box_alpha_icml_2016`

Table 1. Probit regression experiment results

	Average Test Log-likelihood				Average Test Error			
Dataset	**BB-α=1.0**	**BB-α=0.5**	**BB-α=10^{-6}**	**BB-VB**	**BB-α=1.0**	**BB-α=0.5**	**BB-α=10^{-6}**	**BB-VB**
Ionosphere	-0.333±0.022	-0.333±0.022	-0.333±0.022	**-0.333±0.022**	0.124±0.008	0.124±0.008	**0.123±0.008**	0.123±0.008
Madelon	**-0.799±0.006**	-0.920±0.008	-0.953±0.009	-0.953±0.009	**0.445±0.005**	0.454±0.004	0.457±0.005	0.457±0.005
Pima	**-0.501±0.010**	-0.501±0.010	-0.501±0.010	-0.501±0.010	**0.234±0.006**	0.234±0.006	0.235±0.006	0.235±0.006
Colon Cancer	**-2.261±0.402**	-2.264±0.403	-2.268±0.404	-2.268±0.404	**0.303±0.028**	0.307±0.028	0.307±0.028	0.307±0.028
Avg. Rank	**1.895±0.097**	2.290±0.038	2.970±0.073	2.845±0.072	**2.322±0.048**	2.513±0.039	2.587±0.031	2.578±0.031

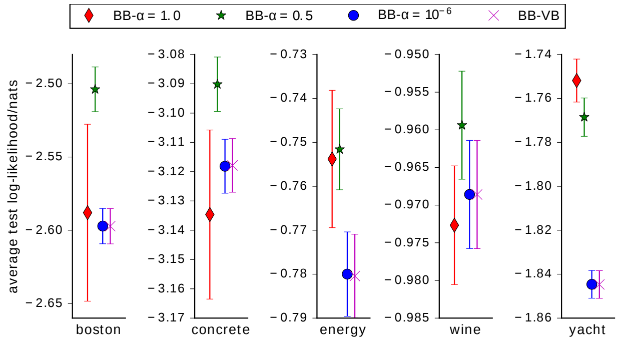

Figure 4. Average test log-likelihood and the ranking comparisons. Best viewed in color.

0.5) and covering the whole probability mass ($\alpha > 0.5$). There are no significant differences in regression error so the results are not shown.

4.3. Neural network classification

The next experiment considers the MNIST digit classification problem. We use neural networks with 2 hidden layers with 400 hidden units per layer, ReLUs and softmax outputs. In this problem, we initialize the mean parameters in q as recommended in Glorot & Bengio (2010) by sampling from a zero-mean Gaussian with variance given by $2/(\mathrm{d_{in}}+\mathrm{d_{out}})$, where $\mathrm{d_{in}}$ and $\mathrm{d_{out}}$ are the dimensionalities of the previous and the next layer in the neural network. In this case we also try $\alpha = -1.0$ in BB-α. We use minibatches of size 250 and run the different methods for 250 epochs. BB-α and BB-VB use now $K = 50$ Monte Carlo samples to approximate the expectations with respect to q in each minibatch. We implemented the neural networks in Theano and ran the different methods on GPUs using Adam with its default parameter values and a learning rate of 0.0001. In this case, to reduce the transference of data from the main memory to the GPU, we update the randomness in the Monte Carlo samples only after processing 10 minibatches instead of after processing each minibatch.

Table 2 summarizes the average test error and test log-likelihood obtained by each method over 20 random initializations. For non-negative alpha settings BB-VB ($\alpha = 0$) returns the best result, and again BB-$\alpha = 10^{-6}$ performs almost identically to the variational inference approach BB-VB. The best performing method is BB-α with $\alpha = -1$, which is expected to move slightly towards fitting a mode. We note here that this is different from the Laplace approximation, which, depending on the curvature at the MAP solution, might return an approximate posterior that also covers spaces other than the mode. On this dataset $\alpha = -1$ returns both higher test log-likelihood and lower test error than all the tested non-negative settings for α.

Table 2. Average Test Error and Log-likelihood in MNIST

Setting	Error/100	Rank	LL/100	Rank
BB-$\alpha = 1.0$	1.51	4.97	-5.51	5.00
BB-$\alpha = 0.5$	1.44	4.03	-5.09	4.00
BB-$\alpha = 10^{-6}$	1.36	2.15	-4.68	2.55
BB-VB	1.36	2.12	-4.68	2.45
BB-$\alpha = -1.0$	**1.33**	**1.73**	**-4.47**	**1.00**

4.4. Clean energy project data

We perform additional experiments with data from the Harvard Clean Energy Project, which is the world's largest materials high-throughput virtual screening effort (Hachmann et al., 2014). It has scanned a large number of molecules of organic photovoltaics to find those with high power conversion efficiency (PCE) using quantum-chemical techniques. The target value within this dataset is the PCE of each molecule. The input features for all molecules in the data set are 512-bit Morgan circular fingerprints, calculated with a bond radius of 2, and derived from the canonical smiles, implemented in the RDkit. We use 50,000 molecules for training and 10,000 molecules for testing.

We consider neural networks with 2 hidden layers with 400 hidden units per layer, ReLUs and Gaussian additive noise at the output. The noise variance is fixed to 0.16, which is the optimal value according to the results reported in Pyzer-Knapp et al. (2015). The initialization and training process is the same as with the MNIST dataset, but using the default learning rate in Adam.

Table 3 summarizes the average test error and test log-likelihood obtained by each method over 20 random initializations. BB-$\alpha = 10^{-6}$ obtains again very similar results to BB-VB. In this case, the best performing method is BB-α with $\alpha = 0.5$, both in terms of test error and test log-likelihood. This was also the best performing method

Table 3. Average Test Error and Test Log-likelihood in CEP Dataset.

CEP Dataset	**BB-**α=1.0	**BB-**α=0.5	**BB-**α=10^{-6}	**BB-VB**
Avg. Error	1.28±0.01	**1.08±0.01**	1.13±0.01	1.14±0.01
Avg. Rank	4.00±0.00	**1.35±0.15**	2.05±0.15	2.60±0.13
Avg. Log-likelihood	-0.93±0.01	**-0.74±0.01**	-1.39±0.03	-1.38±0.02
Avg. Rank	1.95±0.05	**1.05±0.05**	3.40±0.11	3.60±0.11

Table 4. Average Bias.

Dataset	**BB-**α=1.0	**BB-**α=0.5	**BB-**α=10^{-6}
$K=1$	0.2774	0.1214	0.0460
$K=5$	0.0332	0.0189	0.0162
$K=10$	0.0077	0.0013	0.0001

Table 5. Average Standard Deviation Gradient.

Dataset	**BB-**α=1.0	**BB-**α=0.5	**BB-**α=10^{-6}
$K=1$	14.1209	14.0159	13.9109
$K=5$	12.7953	12.8418	12.8984
$K=10$	12.3203	12.4101	12.5058

in the experiments from Section 4.2, which indicates that $\alpha = 0.5$ may be a generally good setting in neural network regression problems. Table 3 shows that $\alpha = 0.5$ attains a balance between the tendency of $\alpha = 1.0$ to perform well in terms of test log-likelihood and the tendency of $\alpha = 10^{-6}$ to perform well in terms of test squared error.

4.5. Analysis of the bias and variance in the gradients

We perform another series of experiments to analyze the bias and the variance in the gradients of (13) as a function of the number K of Monte Carlo samples used to obtain a noisy approximation of the objective function and the value of α used. To estimate the bias in BB-α we run the Adam optimizer for 100 epochs on the Boston housing dataset as described in Section 4.2. Then, we estimate the biased gradient using $K = 1, 5, 10$ Monte Carlo samples from q, which is repeated 1,000 times to obtain an averaged estimate. We also compute an approximation to the ground truth for the unbiased gradient by using $K = 10,000$ Monte Carlo samples. The whole process is performed across 15 minibatches of data points from each split. We then define the bias in the gradient as the averaged L2 norm between the ground truth gradient and the biased gradient across these 15 minibatches, divided by the square root of the dimension of the gradient vector. This definition of bias is not 0 for methods that use unbiased estimators of the gradient, such as BB-VB, because of the variance by sampling on each minibatch. However, we expect this procedure to report larger bias values for BB-α than for BB-VB. Therefore, we subtract from the bias values obtained for BB-α the corresponding bias values obtained for BB-VB. This eliminates from the bias values any additional variance that is produced by having to sample to estimate the unbiased and biased gradient on each minibatch.

Table 4 shows the average bias obtained for each value of K and α across the 50 splits. We observe that the bias is reduced as we increase K and as we make α closer to zero. For $K = 10$ the bias is already very low. To put these bias numbers into context, we also computed a standard deviation-like measure by the square root of the average empirical variance per dimension in the noisy gradient vector over the 15 minibatches. Table 5 shows the average values obtained across the 50 splits, where entries are almost constant as a function of α, and up to 5 orders of magnitude larger than the entries of Table 4 for $K = 10$. This means the bias in the gradient used by BB-α is negligible when compared with the variability that is obtained in the gradient by subsampling the training data.

5. Conclusions and future work

We have proposed BB-α as a black-box inference algorithm to approximate power EP. This is done by considering the energy function used by power EP and constraining the form of the site approximations in this method. The proposed method locally minimizes the α-divergence that is a rich family of divergence measures between distributions including the Kullback-Leibler divergence. Such a method is guaranteed to converge under certain conditions, and can be implemented by optimizing an energy function without having to use inefficient double-loop algorithms. Scalability to large datasets can be achieved by using stochastic gradient descent with minibatches. Furthermore, a combination of a Monte Carlo approximation and automatic differentiation methods allows our technique to be applied in a straightforward manner to a wide range of probabilistic models with complex likelihood factors. Experiments with neural networks applied to small and large datasets demonstrate both the accuracy and the scalability of the proposed approach. The evaluations also indicate the optimal setting for α may vary for different tasks. Future work should provide a theoretical guide and/or automatic tools for modelling with different factors and different α values.

Acknowledgements

JMHL acknowledges support from the Rafael del Pino Foundation. YL thanks the Schlumberger Foundation Faculty for the Future fellowship on supporting her PhD study. MR acknowledges support from UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/L016516/1 for the University of Cambridge Centre for Doctoral Training, the Cambridge Centre for Analysis. TDB thanks Google for funding his European Doctoral Fellowship. DHL acknowledge support from Plan National I+D+i, Grant TIN2013-42351-P and TIN2015-70308-REDT, and from Comunidad de Madrid, Grant S2013/ICE-2845 CASI-CAM-CM. RET thanks EPSRC grant #EP/L000776/1 and #EP/M026957/1.

References

Amari, Shun-ichi. *Differential-Geometrical Methods in Statistic*. Springer, New York, 1985.

Barthelmé, Simon and Chopin, Nicolas. Abc-ep: Expectation propagation for likelihoodfree Bayesian computation. In *ICML*, 2011.

Bastien, Frédéric, Lamblin, Pascal, Pascanu, Razvan, Bergstra, James, Goodfellow, Ian J., Bergeron, Arnaud, Bouchard, Nicolas, and Bengio, Yoshua. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

Bottou, Léon. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):25, 1998.

Cunningham, John P, Hennig, Philipp, and Lacoste-Julien, Simon. Gaussian probabilities and expectation propagation. *arXiv preprint arXiv:1111.6832*, 2011.

Dehaene, Guillaume and Barthelmé, Simon. Expectation propagation in the large-data limit. *arXiv:1503.08060*, 2015.

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

Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pp. 249–256, 2010.

Hachmann, Johannes, Olivares-Amaya, Roberto, Jinich, Adrian, Appleton, Anthony L, Blood-Forsythe, Martin A, Seress, László R, Román-Salgado, Carolina, Trepte, Kai, Atahan-Evrenk, Sule, Er, Süleyman, et al. Lead candidates for high-performance organic photovoltaics from high-throughput quantum chemistry–the harvard clean energy project. *Energy & Environmental Science*, 7(2):698–704, 2014.

Hernández-Lobato, Daniel and Hernández-Lobato, José Miguel. Scalable gaussian process classification via expectation propagation. In *AISTATS*, 2016.

Heskes, Tom and Zoeter, Onno. Expectation propagation for approximate inference in dynamic Bayesian networks. In *UAI*, pp. 216–223. Morgan Kaufmann Publishers Inc., 2002.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *ICLR*, 2014.

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Li, Yingzhen, Hernandez-Lobato, Jose Miguel, and Turner, Richard E. Stochastic expectation propagation. In *NIPS*, pp. 2323–2331, 2015.

Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Minka, Thomas P. Expectation propagation for approximate Bayesian inference. In *UAI*, pp. 362–369. Morgan Kaufmann Publishers Inc., 2001a.

Minka, Thomas P. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Cambridge, MA, USA, 2001b. AAI0803033.

Minka, Thomas P. Power ep. Technical report, Technical report, Microsoft Research, Cambridge, 2004.

Minka, Thomas P. Divergence measures and message passing. Technical report, Technical report, Microsoft Research, 2005.

Opper, Manfred and Winther, Ole. Expectation consistent approximate inference. *The Journal of Machine Learning Research*, 6:2177–2204, 2005.

Pyzer-Knapp, Edward O, Li, Kewei, and Aspuru-Guzik, Alan. Learning from the harvard clean energy project: The use of neural networks to accelerate materials discovery. *Advanced Functional Materials*, 25(41):6495–6502, 2015.

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

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

Seeger, Matthias. Expectation propagation for exponential families. Technical report, 2005.

Teh, Yee Whye, Hasenclever, Leonard, Lienart, Thibaut, Vollmer, Sebastian, Webb, Stefan, Lakshminarayanan, Balaji, and Blundell, Charles. Distributed Bayesian learning with stochastic natural-gradient expectation propagation and the posterior server. *arXiv:1512.09327*, 2015.

Turner, Richard E. and Sahani, Maneesh. Two problems with variational expectation maximisation for time series models. In *Bayesian Time Series Models*, pp. 104–124. Cambridge University Press, 2011a. Cambridge Books Online.

Turner, Richard E. and Sahani, Maneesh. Probabilistic amplitude and frequency demodulation. In *NIPS*, pp. 981–989. 2011b.

Winn, John M and Bishop, Christopher M. Variational message passing. In *Journal of Machine Learning Research*, pp. 661–694, 2005.

Xu, Minjie, Lakshminarayanan, Balaji, Teh, Yee Whye, Zhu, Jun, and Zhang, Bo. Distributed Bayesian posterior sampling via moment sharing. In *NIPS*, pp. 3356–3364, 2014.

Zhu, Huaiyu and Rohwer, Richard. Information geometric measurements of generalisation. Technical report, Technical Report NCRG/4350. Aston University., 1995.