# Doubly Stochastic Variational Bayes for non-Conjugate Inference

**Michalis K. Titsias** MTITSIAS@AUEB.GR
Department of Informatics, Athens University of Economics and Business, Greece

**Miguel Lázaro-Gredilla** MIGUEL@TSC.UC3M.ES
Dpt. Signal Processing & Communications, Universidad Carlos III de Madrid, Spain

## Abstract

We propose a simple and effective variational inference algorithm based on stochastic optimisation that can be widely applied for Bayesian non-conjugate inference in continuous parameter spaces. This algorithm is based on stochastic approximation and allows for efficient use of gradient information from the model joint density. We demonstrate these properties using illustrative examples as well as in challenging and diverse Bayesian inference problems such as variable selection in logistic regression and fully Bayesian inference over kernel hyperparameters in Gaussian process regression.

## 1. Introduction

Modern machine learning and statistical applications require large scale inference in complex models. Bayesian learning provides a probabilistic framework for inference that combines prior knowledge with observed data in a principled manner. However, apart from simple cases involving conjugate models, the Bayesian computations are intractable and approximations based on either Markov Chain Monte Carlo (MCMC) (Robert & Casella, 1999) or variational Bayesian inference (Jordan et al., 1999; Neal & Hinton, 1999; Wainwright & Jordan, 2008) are needed. While MCMC can provide unbiased estimates of Bayesian expectations, in practice designing MCMC algorithms that reliably converge to the stationary posterior distribution can be a notoriously difficult task especially in complex non-conjugate models. On the other hand, variational methods formulate Bayesian inference as an optimization problem, where the objective function is constructed to be a lower bound on the marginal likelihood. This can allow for faster algorithms having a simpler mechanism for monitoring convergence. Despite that, the variational approach cannot be applied as widely as MCMC and this is because the variational objective function requires a high dimensional expectation that becomes intractable for non-conjugate and highly non-linear models.

In this paper, we expand the range of applicability of variational inference algorithms by introducing a simple stochastic optimization algorithm that can be widely applied in non-conjugate models where the joint probability densities are differentiable functions of the parameters. This algorithm is based on stochastic approximation (Robbins & Monro, 1951) and differs from other work on non-conjugate stochastic variational inference (Paisley et al., 2012; Ranganath et al., 2014; Mnih & Gregor, 2014) by allowing for efficient use of gradient information from the model joint density. We demonstrate these properties using illustrative examples as well as in challenging and diverge Bayesian estimation problems such as variable selection in logistic regression and fully Bayesian inference over kernel hyperparameters in Gaussian process regression (Rasmussen & Williams, 2006). For the former application we also introduce a variational objective function, suitable for general-purpose sparse inference, which is hyperparameter-free in the sense that the optimisation over the initial sparsity-determining hyperparameters is dealt with analytically. For the latter application our method provides a general variational inference technique for hyperparameters in Gaussian process regression that is widely applicable to differentiable kernel functions, demonstrating also a very close agreement with ground-truth Monte Carlo estimates obtained by much slower MCMC runs.

Furthermore, the proposed algorithm introduces stochasticity by sampling from the variational distribution. This differs from the data sub-sampling stochasticity used in the variational framework proposed by Hoffman et al. (2010; 2013). We show how to combine the two types of stochasticity, thus deriving a doubly stochastic variational inference algorithm, allowing for efficient non-conjugate inference for large scale problems. We demonstrate experimen-

*Proceedings of the $31^{st}$ International Conference on Machine Learning*, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

tally this doubly stochastic scheme in large-scale Bayesian logistic regression.

Independently from our work, Kingma & Welling (2013) and Rezende et al. (2014) also derived doubly stochastic variational inference algorithms by utilising gradients from the joint probability density. Our work provides an additional perspective and it specialises also on different type of applications such as variable selection and Gaussian process hyperparameter learning.

## 2. Theory

Consider a random vector $\mathbf{z} \in \mathbb{R}^D$ that follows a distribution with a continuous density function $\boldsymbol{\phi}(\mathbf{z})$. We shall assume $\boldsymbol{\phi}(\mathbf{z})$ to exist in standard form so that any parameter mean vector is set to zero and scale parameters are set to one. For instance, $\boldsymbol{\phi}(\mathbf{z})$ could be the standard normal distribution, the standard $t$ distribution, a product of standard logistic distributions, etc. We often refer to $\phi(\mathbf{z})$ as the *standard distribution* and for the time being, we shall leave it unspecified and develop the theory in a general setting. A second assumption we make about $\boldsymbol{\phi}(\mathbf{z})$ is that it permits straightforward simulation of independent samples. We aim to utilise the standard distribution as a building block for constructing correlated variational distributions. While $\boldsymbol{\phi}(\mathbf{z})$ has currently no structure, we can add correlation by applying an invertible transformation,

$$\boldsymbol{\theta} = C\mathbf{z} + \boldsymbol{\mu},$$

where the scale matrix $C$ is taken to be a lower triangular positive definite matrix (i.e. its diagonal elements are strictly positive) and $\boldsymbol{\mu}$ is a real vector. Given that the Jacobian of the inverse transformation is $\frac{1}{|C|}$, the distribution over $\boldsymbol{\theta}$ takes the form

$$q(\boldsymbol{\theta}|\boldsymbol{\mu}, C) = \frac{1}{|C|}\boldsymbol{\phi}\left(C^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu})\right), \tag{1}$$

which is a multivariate and generally correlated distribution having as adjustable parameters the mean vector $\boldsymbol{\mu}$ and the scale matrix $C$. We wish to employ $q(\boldsymbol{\theta}|\boldsymbol{\mu}, C)$ as a variational distribution for approximating the exact Bayesian posterior in the general setting where we have a non-conjugate model. More precisely, we consider a probabilistic model with the joint density

$$g(\boldsymbol{\theta}) = p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}), \tag{2}$$

where $\mathbf{y}$ are data and $\boldsymbol{\theta} \in \mathbb{R}^D$ are all unobserved random variables which can include both latent variables and parameters. Following the standard variational Bayes inference method (Jordan et al., 1999; Neal & Hinton, 1999; Wainwright & Jordan, 2008) we seek to minimise the KL divergence $\mathrm{KL}[q(\boldsymbol{\theta}|\boldsymbol{\mu}, C)||p(\boldsymbol{\theta}|\mathbf{y})]$ between the variational and the true posterior distribution. This can equivalently formulated as the maximisation of the following lower bound on the log marginal likelihood,

$$\mathcal{F}(\boldsymbol{\mu}, C) = \int q(\boldsymbol{\theta}|\boldsymbol{\mu}, C) \log \frac{g(\boldsymbol{\theta})}{q(\boldsymbol{\theta}|\boldsymbol{\mu}, C)} d\boldsymbol{\theta}, \tag{3}$$

where $q(\boldsymbol{\theta}|\boldsymbol{\mu}, C)$ is given by (1). By changing variables according to $\mathbf{z} = C^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu})$, the above is written as

$$\begin{aligned}\mathcal{F}(\boldsymbol{\mu}, C) &= \int \boldsymbol{\phi}(\mathbf{z}) \log \frac{g(C\mathbf{z} + \boldsymbol{\mu})|C|}{\boldsymbol{\phi}(\mathbf{z})} d\mathbf{z} \\ &= \mathbb{E}_{\boldsymbol{\phi}(\mathbf{z})}\left[\log g(C\mathbf{z} + \boldsymbol{\mu})\right] + \log|C| + \mathcal{H}_{\boldsymbol{\phi}},\end{aligned} \tag{4}$$

where $\log|C| = \sum_{d=1}^D \log C_{dd}$ and $\mathcal{H}_{\boldsymbol{\phi}}$ denotes the entropy of $\boldsymbol{\phi}(\mathbf{z})$ which is constant with respect to the variational parameters $(\boldsymbol{\mu}, C)$ and therefore it can be ignored when maximising the bound. Also notice that the above requires integration over the distribution $\boldsymbol{\phi}(\mathbf{z})$ which exists in standard form and therefore it does not depend on the variational parameters $(\boldsymbol{\mu}, C)$. These parameters somehow have been transferred inside the logarithm of the joint density.

Further, it is worth noticing that when the logarithm of the joint model density, i.e. $\log g(\boldsymbol{\theta})$, is concave with respect to $\boldsymbol{\theta}$, the lower bound in (4) is also concave with respect to the variational parameters $(\boldsymbol{\mu}, C)$ and this holds for any standard distribution $\boldsymbol{\phi}(\mathbf{z})$; see the supplementary material for a formal statement and proof. This generalises the result of Challis & Barber (2011; 2013), who proved it for the variational Gaussian approximation, and it is similar to the generalisation presented in (Staines & Barber, 2012).

To fit the variational distribution to the true posterior, we need to maximise the bound (4) and therefore we consider the gradients over $\boldsymbol{\mu}$ and $C$,

$$\nabla_{\boldsymbol{\mu}} \mathcal{F}(\boldsymbol{\mu}, C) = \mathbb{E}_{\boldsymbol{\phi}(\mathbf{z})}\left[\nabla_{\boldsymbol{\mu}} \log g(C\mathbf{z} + \boldsymbol{\mu})\right], \tag{5}$$

$$\nabla_C \mathcal{F}(\boldsymbol{\mu}, C) = \mathbb{E}_{\boldsymbol{\phi}(\mathbf{z})}\left[\nabla_C \log g(C\mathbf{z} + \boldsymbol{\mu})\right] + \Delta_C, \tag{6}$$

where $\Delta_C$ denotes the diagonal matrix with elements $(1/C_{11}, \ldots, 1/C_{DD})$ in the diagonal. Also the term $\nabla_C \log g(C\mathbf{z} + \boldsymbol{\mu})$ in eq. (6) should be understood as the partial derivatives w.r.t. $C$ stored in a lower triangular matrix so that there is one-to-one correspondence with the elements in $C$. A first observation about the gradients above is that they involve taking derivatives of the logarithm of the joint density by adding randomness through $\mathbf{z}$ and then averaging out. To gain more intuition, we can equivalently express them in the original space of $\boldsymbol{\theta}$ by changing variables in the reverse direction according to $\boldsymbol{\theta} = C\mathbf{z} + \boldsymbol{\mu}$. Using the chain rule we have that $\nabla_{\boldsymbol{\mu}} \log g(C\mathbf{z} + \boldsymbol{\mu}) = \nabla_{C\mathbf{z}+\boldsymbol{\mu}} \log g(C\mathbf{z} + \boldsymbol{\mu})$ and similarly $\nabla_C \log g(C\mathbf{z}+\boldsymbol{\mu}) = \nabla_{C\mathbf{z}+\boldsymbol{\mu}} \log g(C\mathbf{z}+\boldsymbol{\mu})\mathbf{z}^T$ where again $\nabla_{C\mathbf{z}+\boldsymbol{\mu}} \log g(C\mathbf{z} + \boldsymbol{\mu})\mathbf{z}^T$ should be understood as taking the lower triangular part after performing the outer vector

**Algorithm 1** Doubly stochastic variational inference

```
Input: $\boldsymbol{\phi}$, $\mathbf{y}$, $\boldsymbol{\theta}$, $\nabla \log g$.
Initialise $\boldsymbol{\mu}^{(0)}, C^{(0)}, t = 0$.
repeat
    $t = t + 1$;
    $\mathbf{z} \sim \boldsymbol{\phi}(\mathbf{z})$;
    $\boldsymbol{\theta}^{(t-1)} = C^{(t-1)}\mathbf{z} + \boldsymbol{\mu}^{(t-1)}$;
    $\boldsymbol{\mu}^{(t)} = \boldsymbol{\mu}^{(t-1)} + \rho_t \nabla_{\boldsymbol{\theta}} \log g(\boldsymbol{\theta}^{(t-1)})$ ;
    $C^{(t)} = C^{(t-1)} + \rho_t \left( \nabla_{\boldsymbol{\theta}} \log g(\boldsymbol{\theta}^{(t-1)}) \times \mathbf{z}^T + \Delta_{C^{(t-1)}} \right)$;
until convergence criterion is met.
```

product. These observations allow to transform (5) and (6) in the original space of $\boldsymbol{\theta}$ as follows,

$$\nabla_{\boldsymbol{\mu}} \mathcal{F}(\boldsymbol{\mu}, C) = \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\mu},C)} \left[ \nabla_{\boldsymbol{\theta}} \log g(\boldsymbol{\theta}) \right], \tag{7}$$

$$\nabla_C \mathcal{F}(\boldsymbol{\mu}, C) = \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\mu},C)} \left[ \nabla_{\boldsymbol{\theta}} \log g(\boldsymbol{\theta}) \times (\boldsymbol{\theta} - \boldsymbol{\mu})^T C^{-T} \right] + \Delta_C, \tag{8}$$

Eq. (7) is particularly intuitive as it says that the gradient over $\boldsymbol{\mu}$ is simply the gradient of the logarithm of the joint density with respect to the parameters $\boldsymbol{\theta}$ averaged over the variational distribution.

We would like now to optimise the variational lower bound over $(\boldsymbol{\mu}, C)$ using a stochastic approximation procedure. To this end, we need to provide stochastic gradients having as expectations the exact quantities. Based on the expressions (5)-(6) or their equivalent counterparts (7)-(8) we can proceed by firstly drawing $\boldsymbol{\theta}^{(s)} \sim q(\boldsymbol{\theta}|\boldsymbol{\mu}, C)$, and then using $\nabla_{\boldsymbol{\theta}} \log g(\boldsymbol{\theta}^{(s)})$ as the stochastic direction for updating $\boldsymbol{\mu}$ and $\nabla_{\boldsymbol{\theta}} \log g(\boldsymbol{\theta}^{(s)}) \times (\boldsymbol{\theta}^{(s)} - \boldsymbol{\mu})^T C^{-T}$ as the direction for updating $C$. To draw $\boldsymbol{\theta}^{(s)}$, we need first to sample $\mathbf{z}$ from $\boldsymbol{\phi}(\mathbf{z})$ (which by assumption is possible) and then deterministically obtain $\boldsymbol{\theta}^{(s)} = C\mathbf{z} + \boldsymbol{\mu}$. Based on the latter $(\boldsymbol{\theta}^{(s)} - \boldsymbol{\mu})^T C^{-T}$ is just $\mathbf{z}^T$, therefore the computationally efficient way to implement the whole stochastic approximation scheme is as summarised in Algorithm 1.

Based on the theory of stochastic approximation (Robbins & Monro, 1951), using a schedule of the learning rates $\{\rho_t\}$ such that $\sum \rho_t = \infty$, $\sum \rho_t^2 < \infty$, the iteration in Algorithm 1 will converge to a local maxima of the bound in (3) or to the global maximum when this bound is concave. For notational simplicity we have assumed common learning rate sequences for $\boldsymbol{\mu}$ and $C$, however, in practice we can use different sequences and the algorithm remains valid.

We will refer to the above stochastic approximation algorithm as *doubly stochastic variational inference* (DSVI) because it introduces stochasticity in a different direction than the standard stochastic variational inference proposed by (Hoffman et al., 2010; 2013). The latter is based on subsampling the training data and performing online parameter updates by using each time a single data point or a small "mini-batch" which is analogous to other online learning algorithms (Bottou, 1998; Bottou & Bousquet, 2008). Instead, our algorithm introduces stochasticity by sampling from the variational distribution. Notice that the latter type of stochasticity was first introduced by (Paisley et al., 2012) who proposed a different stochastic gradient for variational parameters that we compare against in Section 2.2. For joint probability models with a factorised likelihood, the two types of stochasticity can be combined so that in each iteration the stochastic gradients are computed by both sampling from the variational distribution and using a mini-batch of $n \ll N$ data points. It is straightforward to see that such doubly stochastic gradients are unbiased estimates of the true gradients and therefore the whole scheme is valid. In the experiments, we demonstrate the double stochasticity for learning from very large data sets in Bayesian logistic regression. However, for simplicity in the remainder of our presentation we will not analyse further the mini-batch type of stochasticity.

Finally, for inference problems where the dimensionality of $\boldsymbol{\theta}$ is very large and therefore it is impractical to optimise over a full scale matrix C, we can consider a diagonal matrix in which case the scales can be stored in a $D$-dimensional strictly positive vector $\mathbf{c}$. Then, the update over $C$ in Algorithm 1 is replaced by

$$c_d^{(t)} = c_d^{(t-1)} + \rho_t \left( \frac{\partial \log g(\boldsymbol{\theta}^{(t-1)})}{\partial \theta_d} z_d + \frac{1}{c_d^{(t-1)}} \right), \tag{9}$$

where $d = 1, \ldots, D$. Notice that when the initial standard distribution $\boldsymbol{\phi}(\mathbf{z})$ is fully factorised the above leads to a fully factorised variational approximation $q(\boldsymbol{\theta}|\boldsymbol{\mu}, \mathbf{c}) = \prod_{d=1}^D q_d(\theta_d|\mu_d, c_d)$. While such approach can have lower accuracy than using the full scale matrix $C$, it has the great advantage that it can scale up to thousands or millions of parameters. In Section 3.2, we use this scheme for variable selection in logistic regression and we also introduce a novel variational objective function for sparse inference following the idea of automatic relevance determination.

In the following two sections we elaborate more on the properties of DSVI by drawing connections with the Gaussian approximation (Section 2.1) and analysing convergence properties (Section 2.2).

### 2.1. Connection with the Gaussian approximation

The Gaussian approximation, see (Barber & Bishop, 1998; Seeger, 1999) and more recently (Opper & Archambeau, 2009), assumes a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma)$ as a variational distribution to approximate the exact model posterior which leads to the maximisation of the lower bound

$$\mathcal{F}(\boldsymbol{\mu}, \Sigma) = \int \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma) \log \frac{g(\boldsymbol{\theta})}{\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma)} d\boldsymbol{\theta}. \tag{10}$$

The maximisation relies on analytical integration (or in the worst case in one-dimensional numerical integration) for computing $\int \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma) \log g(\boldsymbol{\theta}) d\boldsymbol{\theta}$, which subsequently can allow to tune the variational parameters $(\boldsymbol{\mu}, \Sigma)$ using gradient optimization methods (Opper & Archambeau, 2009; Honkela et al., 2011). More recently, Challis & Barber (2011; 2013) use this framework with the parametrisation $\Sigma = CC^T$ and show that when $\log g(\boldsymbol{\theta})$ is concave, the bound is concave w.r.t. $(\boldsymbol{\mu}, C)$. The limitation of these approaches is that they rely on $\log g(\boldsymbol{\theta})$ having a simple form so that the integral $\int \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma) \log g(\boldsymbol{\theta}) d\boldsymbol{\theta}$ is analytically tractable. Unfortunately, this constraint excludes many interesting Bayesian inference problems, such as inference over kernel hyperparameters in Gaussian process models, inference over weights in neural networks and others.

In contrast, our stochastic variational framework only relies on $\log g(\boldsymbol{\theta})$ being a differentiable function of $\boldsymbol{\theta}$. If we specify the distribution $\boldsymbol{\phi}(\mathbf{z})$ to be the standard normal $\mathcal{N}(\mathbf{z}|\mathbf{0}, I)$, the lower bound in (4) becomes the Gaussian approximation bound in (10) with the parametrisation $\Sigma = CC^T$. Subsequently, if we apply the DSVI iteration according to Algorithm 1 with the specialization that $\mathbf{z}$ is drawn from $\mathcal{N}(\mathbf{z}|\mathbf{0}, I)$, the algorithm will stochastically maximise the Gaussian approximation bound. Therefore, DSVI allows to apply the Gaussian approximation to a much wider range of models.

A different direction of flexibility in the DSVI framework is concerned with the choice of the standard distribution $\boldsymbol{\phi}(\mathbf{z})$. Clearly, if we choose a non-Gaussian form we obtain non-Gaussian variational approximations. For instance, when this distribution is the standard $t$ with $\nu$ degrees of freedom, i.e. $\boldsymbol{\phi}(\mathbf{z}) = \text{St}(\mathbf{z}, \nu, \mathbf{0}, I)$, the variational distribution $q(\boldsymbol{\theta}|\boldsymbol{\mu}, C)$ becomes the general $t$ distribution with $\nu$ degrees of freedom, i.e. $q(\boldsymbol{\theta}|\boldsymbol{\mu}, C) = \text{St}(\mathbf{z}, \nu, \boldsymbol{\mu}, CC^T)$. A flexible way to define a standard distribution is to assume a fully factorised form $\boldsymbol{\phi}(\mathbf{z}) = \prod_{d=1}^D \phi_d(z_d)$ and then select the univariate marginals, $\phi_d(z)$ with $d = 1, \ldots, D$, from a family of univariate distributions for which exact simulation is possible. While in such cases the resulting $q(\boldsymbol{\theta}|\boldsymbol{\mu}, C)$ can be of non-standard form, simulating exact samples from this distribution is always straightforward since $\boldsymbol{\theta}^{(s)} = C\mathbf{z} + \boldsymbol{\mu}$, $\mathbf{z} \sim \boldsymbol{\phi}(\mathbf{z})$ is by construction an independent sample from $q(\boldsymbol{\theta}|\boldsymbol{\mu}, C)$. In the current experimental study presented in Section 3 we only consider the DSVI algorithm for stochastic maximisation of the Gaussian approximation bound. We defer the experimentation with other forms of the $\phi(\mathbf{z})$ distribution for future work.

### 2.2. Illustrative convergence analysis

In this section, we informally analyse the convergence behaviour of DSVI, i.e., its ability to locate a local maximum of the variational lower bound. We will use an illustrative example where $g(\boldsymbol{\theta})$ is proportional to a multivariate Gaussian and we will compare our method with an alternative doubly stochastic approximation approach proposed in (Paisley et al., 2012). For simplicity next we will be using the notation $f(\boldsymbol{\theta}) = \log g(\boldsymbol{\theta})$.

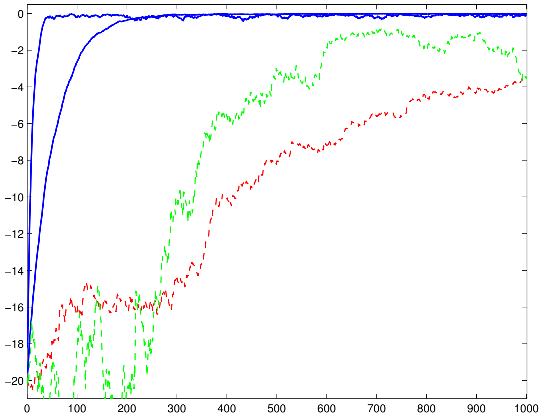

*Figure 1.* The evolution of the lower bound (optimal value is zero) obtained by the two stochastic approximation methods employing two alternative stochastic gradients for fitting a 10-dimensional Gaussian distribution $\mathcal{N}(\boldsymbol{\theta}|\mathbf{m}, I)$ where $\mathbf{m}$ was set to the vector of twos. The variational mean was initialised to the zero vector. For each method two realisations are shown (one with small and one with large learning rate). Blue solid lines correspond to DSVI while green and red lines to the alternative algorithm.

Recall eq. (7), which gives the gradient over the variational mean $\boldsymbol{\mu}$, that we repeat here for convenience,

$$\int \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma) \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{11}$$

where we have also specified the variational distribution $q(\boldsymbol{\theta}|\boldsymbol{\mu}, C)$ to be the Gaussian $\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma)$ with $\Sigma = CC^T$. Based on the above, the implied single-sample stochastic approximation of the gradient is $\nabla_{\boldsymbol{\theta}^{(s)}} f(\boldsymbol{\theta}^{(s)})$, where $\boldsymbol{\theta}^{(s)} \sim \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma)$, which is precisely what DSVI uses. The question that arises now is whether exists an alternative way to write the exact gradient over $\boldsymbol{\mu}$ that can give rise to a different stochastic gradient and more importantly how the different stochastic gradients compare with one another in terms of convergence. In turns out that an alternative expression for the gradient over $\boldsymbol{\mu}$ is obtained by directly differentiating the initial bound in (10) which gives

$$\int \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma) f(\boldsymbol{\theta}) \Sigma^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}) \, d\boldsymbol{\theta}. \tag{12}$$

This form can be also obtained by the general method in (Paisley et al., 2012) according to which the gradient over some variational parameter $\psi$ in a variational distribution $q(\boldsymbol{\theta}|\psi)$ is computed based on $\int f(\boldsymbol{\theta}) q(\boldsymbol{\theta}|\psi) \nabla_{\psi} [\log q(\boldsymbol{\theta}|\psi)] \, d\boldsymbol{\theta}$ which in the case

$q(\boldsymbol{\theta}|\psi) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma)$ and $\psi = \boldsymbol{\mu}$ reduces to the expression in (12). This alternative expression suggests as one-sample stochastic gradient the quantity $f(\boldsymbol{\theta}^{(s)})\Sigma^{-1}\left(\boldsymbol{\theta}^{(s)} - \boldsymbol{\mu}\right)$, where $\boldsymbol{\theta}^{(s)} \sim \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma)$. While sample averages of any size for both stochastic gradients are unbiased, the second one suffers from undesirable random walk behaviour when used for stochastic maximisation of the variational lower bound. Intuitively, this is because it doesn't utilise gradient information from the log joint density $f(\boldsymbol{\theta})$ that could allow to locate quickly a mode of the posterior distribution. Next we illustrate this using an example.

Suppose $f(\boldsymbol{\theta}) = \log\left(\text{const} \times \mathcal{N}(\boldsymbol{\theta}|\mathbf{m}, \Sigma)\right)$, i.e. the joint density is proportional to a multivariate Gaussian having the same covariance matrix with the variational distribution but different mean $\mathbf{m}$. For further simplification let us set $\Sigma = I$. The stochastic gradient used in DSVI becomes $(\mathbf{m} - \boldsymbol{\theta}^{(s)})$ while the alternative gradient is $f(\boldsymbol{\theta}^{(s)})(\boldsymbol{\theta}^{(s)} - \boldsymbol{\mu})$. Given that we initialise $\boldsymbol{\mu}$ far away from $\mathbf{m}$, the first gradient will allow updating $\boldsymbol{\mu}$ essentially via a deterministic transient phase where $\boldsymbol{\mu}$ rapidly moves towards its optimal value $\mathbf{m}$ as shown in Figure 1 (blue solid lines) for two different values of the learning rate (assumed constant during each run). Once the global maximum area is reached, DSVI diffuses around the global maximum with a variance that increases with the learning rate. In contrast, the alternative gradient exhibits random walk behaviour even in the transient phase (dashed green and red lines). Intuitively, this can be explained by the vector $\boldsymbol{\theta}^{(s)} - \boldsymbol{\mu}$ which determines the direction of movement. Clearly, this vector will point in any direction with the same probability and what really saves the algorithm from not diverging is that the random walk is drifted towards the global optimum area due to the penalty imposed by the scale $f(\boldsymbol{\theta}^{(s)})$.

The random walk behaviour and high variance of the alternative stochastic gradient is well-acknowledged by Paisley et al. (2012) who devised sophisticated control variate methods to improve convergence. Furthermore, the method of Paisley et al. (2012) can be applied to a more general class of inference problems than ours. However, for the problems our method is applicable to, we believe it should be preferred due to its efficiency and algorithmic simplicity.

## 3. Experiments

In this section, we apply the DSVI algorithm to different types of non-conjugate models. In Section 3.1 we consider standard concave Bayesian logistic regression, while in Section 3.2 and 3.3 we further elaborate on logistic regression by discussing how to deal with automatic variable selection and very large datasets. In Section 3.4 we consider DSVI for Gaussian process hyperparameter inference.

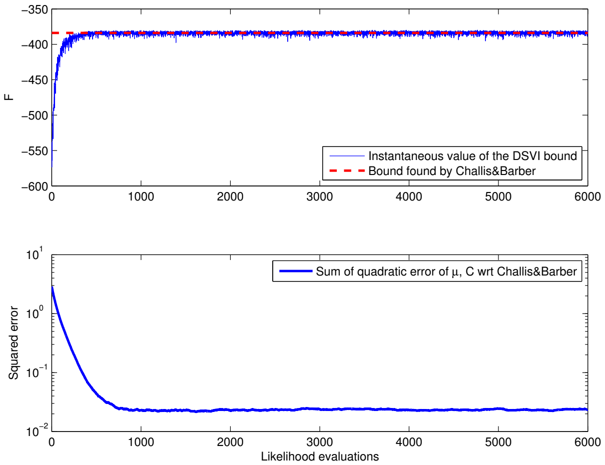

*Figure 2.* Top: Evolution of the instantaneous bound (see supplementary material for a definition) towards the reference value provided by Challis & Barber (2013). Bottom: Evolution of the squared error of the parameters, $||\boldsymbol{\mu}^{(t)} - \boldsymbol{\mu}^*||^2 + ||C^{(t)} - C^*||^2$.

### 3.1. Bayesian logistic regression

We first consider DSVI for standard Bayesian logistic regression. Given a dataset $\mathcal{D} \equiv \{\tilde{\mathbf{x}}_n, y_n\}_{n=1}^N$, where $\tilde{\mathbf{x}}_n \in \mathbb{R}^{\widetilde{D}}$ and $y_n \in \{-1, +1\}$, we model the probability of the outputs conditional on some weights $\boldsymbol{\theta}$ as $p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{n=1}^N s(y_n \mathbf{x}_n^\top \boldsymbol{\theta})$, where $s(a)$ is the logistic function and $\mathbf{x}_n = [1 \;\; \tilde{\mathbf{x}}_n^\top]^\top$ is the input augmented with an one to account for the bias. Using this likelihood and a fixed Gaussian prior on the weights $p(\boldsymbol{\theta}) = \mathcal{N}(0, I_D)$ (with $D = \widetilde{D} + 1$), we have fully specified the model and we can iterate Algorithm 1 for any given dataset. In this case, since the likelihood is log-concave, the complete functional (4) becomes concave so that convergence to the optimal solution is guaranteed. Results using this model are therefore bound to be identical to those using the method in (Challis & Barber, 2013), but obtained without the need of numerical quadrature and using instead simpler stochastic gradient ascent (which of course will need a larger number of iterations to attain convergence).

For the above simple setting and using the well-known `Pima` indians diabetes data set from the UCI repository, we show on Figure 2 the convergence of our method, using the result of running the code from (Challis & Barber, 2013) as a reference. For both methods we assumed a full scale matrix $C$ so that the complexity per iteration is linear with the number of data points $N$ and quadratic with the dimensionality $D$. Only 16 L-BFGS likelihood evaluations are required for the convergence of the reference method, whereas around 500 evaluations are needed for DSVI (no stochasticity over the data set was used for this experiment). However, the actual running time for DSVI was only around 3 times longer due to its simplicity.

### 3.2. Variable selection for logistic regression

In this section, we consider DSVI for variable selection in large scale Bayesian logistic regression where the input dimensionality $D$ can be of order of thousands or millions. For such cases, it is impractical to learn a correlated variational distribution with a full scale matrix $C$ and therefore we use a diagonal scale matrix so that the complexity becomes linear with $D$. As explained in Section 2, in such cases the variational approximation takes a factorised form, i.e. $q(\boldsymbol{\theta}|\boldsymbol{\mu}, \mathbf{c}) = \prod_{d=1}^{D} q_d(\theta_d|\mu_d, c_d)$. Next, based on the former factorised approximation, we introduce a variational inference algorithm specialised to variable selection.

The starting point of our method is the automatic relevance determination (ARD) idea, as used for instance in the relevance vector machine (Tipping, 2001). Specifically, the weights $\boldsymbol{\theta}$ are assigned a zero-mean Gaussian prior $p(\boldsymbol{\theta}) = \mathcal{N}(0, \boldsymbol{\Lambda})$ having a diagonal covariance matrix $\boldsymbol{\Lambda}$, i.e. $\boldsymbol{\Lambda} = \text{diag}(\ell_1^2, \ldots, \ell_D^2)$ with each $\ell_d^2$ representing the prior variance of $\theta_d$. We would like to select the hyperparameters $\boldsymbol{\Lambda}$ by maximising an approximation to the marginal likelihood which, under the variational framework, reduces to maximising the variational bound $\mathcal{F}(\boldsymbol{\mu}, \mathbf{c}, \boldsymbol{\Lambda})$ w.r.t. both the variational parameters $(\boldsymbol{\mu}, \mathbf{c})$ and the hyperparameters $\boldsymbol{\Lambda}$. A standard way to perform this maximisation is by using variational EM, where we alternate between updating $(\boldsymbol{\mu}, \mathbf{c})$ given $\boldsymbol{\Lambda}$ and updating $\boldsymbol{\Lambda}$ given $(\boldsymbol{\mu}, \mathbf{c})$. However, this scheme can exhibit slow convergence due to the high dependence between the variational parameters and the hyperparameters. Fortunately, as we will now show, the optimisation of $\mathcal{F}(\boldsymbol{\mu}, \mathbf{c}, \boldsymbol{\Lambda})$ w.r.t. $\boldsymbol{\Lambda}$ can be carried out analytically. This results in an elegant and simplified form for the final variational bound, the maximisation of which can exhibit faster convergence.

Firstly note that while DSVI is generally applicable to any non-conjugate model, more efficient algorithms could be obtained for cases in which the expectation (under the variational distribution) for some part of the log joint density can be performed analytically. An example of this is when the prior $p(\boldsymbol{\theta})$ is Gaussian, as in the above logistic regression model, where the joint density takes the form $g(\boldsymbol{\theta}) = \widetilde{g}(\boldsymbol{\theta})\mathcal{N}(0, \boldsymbol{\Lambda})$ with $\widetilde{g}(\boldsymbol{\theta}) = p(\mathbf{y}|\boldsymbol{\theta})$ denoting the likelihood. Then, the variational lower bound is explicitly written in the form

$$\mathcal{F}(\boldsymbol{\mu}, \mathbf{c}, \boldsymbol{\Lambda}) = \mathbb{E}_{\phi(\mathbf{z})}\left[\log \widetilde{g}(\mathbf{c} \circ \mathbf{z} + \boldsymbol{\mu})\right] + \frac{1}{2}\sum_{d=1}^{D} \log c_d^2 - \frac{1}{2}\sum_{d=1}^{D} \log \ell_d^2 - \frac{1}{2}\sum_{d=1}^{D} \frac{c_d^2 + \mu_d^2}{\ell_d^2} + \frac{D}{2}. \quad (13)$$

The maximum for each hyperparameter $\ell_d^2$ can be found analytically by setting the corresponding gradient to zero, which yields $(\ell_d^2)^* = c_d^2 + \mu_d^2$. By substituting these optimal values back into the lower bound we obtain

$$\mathcal{F}(\boldsymbol{\mu}, \mathbf{c}) = \mathbb{E}_{\phi(\mathbf{z})}\left[\log \widetilde{g}(\mathbf{c} \circ \mathbf{z} + \boldsymbol{\mu})\right] + \frac{1}{2}\sum_{d=1}^{D} \log \frac{c_d^2}{c_d^2 + \mu_d^2}. \quad (14)$$

This objective function has a rather simple form and it has the elegant property that it depends solely on the variational parameters $(\boldsymbol{\mu}, \mathbf{c})$. The second term in the sum can be thought of as regularisation term where each individual term $\log \frac{c_d^2}{c_d^2+\mu_d^2}$ encourages sparsity and, for instance, it can allow to shrink a variational mean parameter $\mu_d$ to zero whenever the corresponding input dimension is somehow redundant for solving the classification task. It is straightforward to apply DSVI to maximise the above variational objective function. All update equations and complete pseudo-code is described by Algorithm 2 in the supplementary material. Next we refer to this algorithm as DSVI-ARD.

We applied DSVI-ARD for binary classification in three cancer-related data sets[1] that are summarized in Table 1, in which the input variables are different gene expression measurements associated with patients and the output variable identify whether the patients have a certain type of cancer or not; see e.g. (Shevade & Keerthi, 2003). Notice that in all three datasets the number of training points is much smaller than the number of input dimensions. Using DSVI-ARD we solve these binary classification problems and report predictions in Table 2. For comparison purposes we also applied standard non-sparse Bayesian logistic regression with a fixed vague Gaussian prior over the parameters (denoted by CONCAV in Table 2). These results show that the ARD model is more consistent in avoiding overfitting, whereas CONCAV is not so consistent since, for instance, it overfits the `Leukemia` data set.

To visualize the ability to perform variable selection, the second row of Figure 3 displays the final values of the variational mean vector $\boldsymbol{\mu}$. Clearly, in all three datasets these mean vectors are highly sparse which shows that the proposed method is effective in identifying the features (genes) that are relevant for solving each classification task.

Finally, the learning rate sequences and annealing schedule when applying DSVI-ARD to all above problems was chosen as follows. The learning rate $\rho_t$ is initialised to $\rho_0 = 0.05/\#$training examples and scaled every 5000 iterations by a factor of 0.95. This learning rate is used to update $\boldsymbol{\mu}$, whereas $0.1\rho_t$ is used to update $\mathbf{c}$. A total of $10^5$ iterations was considered. The panels in the first row of Figure 3 show the evolution of averaged values for the lower bound over the iterations of the algorithm.

[1] Available from `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html`.

*Table 1.* Size and number of features of each cancer data set.

| Data set | #Train | #Test | $D$ |
|---|---|---|---|
| Colon | 42 | 20 | 2,000 |
| Leukemia | 38 | 34 | 7,129 |
| Breast | 38 | 4 | 7,129 |

*Table 2.* Train and test errors for the three cancer datasets and for each method: CONCAV is the original DSVI algorithm with a fixed prior, whereas ARD is the feature-selection version.

| Problem | Train Error | Test Error |
|---|---|---|
| Colon (ARD) | 0/42 | 1/20 |
| Colon (CONCAV) | 0/42 | 0/20 |
| Leukemia (ARD) | 0/38 | 3/34 |
| Leukemia (CONCAV) | 0/38 | 12/34 |
| Breast (ARD) | 0/38 | 2/4 |
| Breast (CONCAV) | 0/38 | 0/4 |

*Table 3.* Size and sparsity level of each large-scale data set.

| Data set | #Train | #Test | $D$ | #Nonzeros |
|---|---|---|---|---|
| a9a | 32,561 | 16,281 | 123 | 451,592 |
| rcv1 | 20,242 | 677,399 | 47,236 | 49,556,258 |
| Epsilon | 400,000 | 100,000 | 2,000 | 800,000,000 |

*Table 4.* Test error rates for DSVI-ARD and $\ell_1$-logistic regression on three large-scale data sets.

| Data set | DSVI ARD | Log. Reg. | $\lambda$ |
|---|---|---|---|
| a9a | 0.1507 | 0.1500 | 2 |
| rcv1 | 0.0414 | 0.0420 | 4 |
| Epsilon | 0.1014 | 0.1011 | 0.5 |

*Table 5.* Performance measures of GP regression where hyperparameters are selected by ML-II, DSVI or MCMC.

| Data set | | ML-II | DSVI | MCMC |
|---|---|---|---|---|
| Boston | (smse) | 0.0743 | 0.0709 | 0.0699 |
| | (nlpd) | 0.1783 | 0.1425 | 0.1317 |
| Bodyfat | (smse) | 0.1992 | 0.0726 | 0.0726 |
| | (nlpd) | -0.1284 | -2.0750 | -2.0746 |
| Pendulum | (smse) | 0.2727 | 0.2807 | 0.2801 |
| | (nlpd) | 0.4537 | 0.4465 | 0.4462 |

### 3.3. Large-scale data sets

In order to demonstrate the scalability of the proposed method, we run it on three well-known large-scale binary classification datasets a9a, rcv1, and Epsilon, whose details are listed on Table 3. Data set a9a is derived from "Adult" in UCI repository, rcv1 is an archive of manually categorised news stories from Reuters (we use the original train/test split), and Epsilon is an artificial data set from PASCAL's large-scale learning challenge 2008.

We use again the Bayesian logistic regression model with variable selection and we applied the DSVI-ARD algorithm described previously. For all problems, mini-batches of size 500 are used, so this process does not ever require the whole data set to be loaded in memory. We contrast our results with standard $\ell_1$-logistic regression, which exactly minimises the convex functional $L(\mathbf{w}) = ||\mathbf{w}||_1 - \lambda \sum_{n=1}^N \log s(y_n \mathbf{x}_n^\top \mathbf{w})$. Both methods are run on the exact same splits. The value of $\lambda$ was selected using 5-fold cross-validation. Results are reported on Table 4 and show the compromises made between both approaches.

The proposed approach scales well to very large data sets but it does not outperform $\ell_1$-logistic regression in these examples. This is expected, since the number of data points is so high that there is little benefit from using a Bayesian approach here. Note, however, the slight advantage obtained for rcv1, where there are a huge number of dimensions. Another benefit of DSVI-ARD is the low memory requirements (we needed a 32GB RAM computer to run the $\ell_1$-logistic regression, whereas a 4GB one was enough for DSVI-ARD). In contrast, logistic regression was more than 100 times faster in achieving convergence (using the highly optimised LIBLINEAR software).

### 3.4. Gaussian process hyperparameters

Gaussian processes (GPs) are non-parametric Bayesian models widely used to solve regression tasks. In a typical setting, a regression data set $\mathcal{D} \equiv \{\mathbf{x}_n, y_n\}_{n=1}^N$ with $\mathbf{x}_n \in \mathbb{R}^D$ and $y_n \in \mathbb{R}$ is modelled as $y_n = f(\mathbf{x}_n) + \varepsilon_n$, where $\varepsilon_n \sim \mathcal{N}(0, \sigma^2)$ and $f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}))$, for some kernel hyperparameters $\boldsymbol{\theta}$ and noise variance $\sigma^2$.

Point estimates for the hyperparameters are typically obtained by optimising the marginal likelihood of the GP using some gradient ascent procedure (Rasmussen & Williams, 2006). Here, we suggest to replace this procedure with stochastic gradient ascent optimisation of the lower bound that provides a posterior distribution over the hyperparameters. While the stochastic nature of the proposed method will probably imply that more marginal likelihood evaluations are required for convergence, this additional computational cost will make the model more resistant to overfitting and provide a posterior over the hyperparameters at a fraction of the cost of full MCMC.

Using a GP with kernel $k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 e^{-\frac{1}{2}\sum_{d=1}^D \frac{(x_d - x'_d)^2}{\ell_d^2}}$, we place vague independent normal priors over the hyperparameters in log space and compute the posterior and predictive densities for three data sets: Boston, Bodyfat, and Pendulum. Obviously, for this model, no stochasticity over the data set is used. Boston is a UCI data set related to housing values in Boston, Bodyfat requires predicting the percentage of body fat from several body measurements and in Pendulum the change in angular veloc-

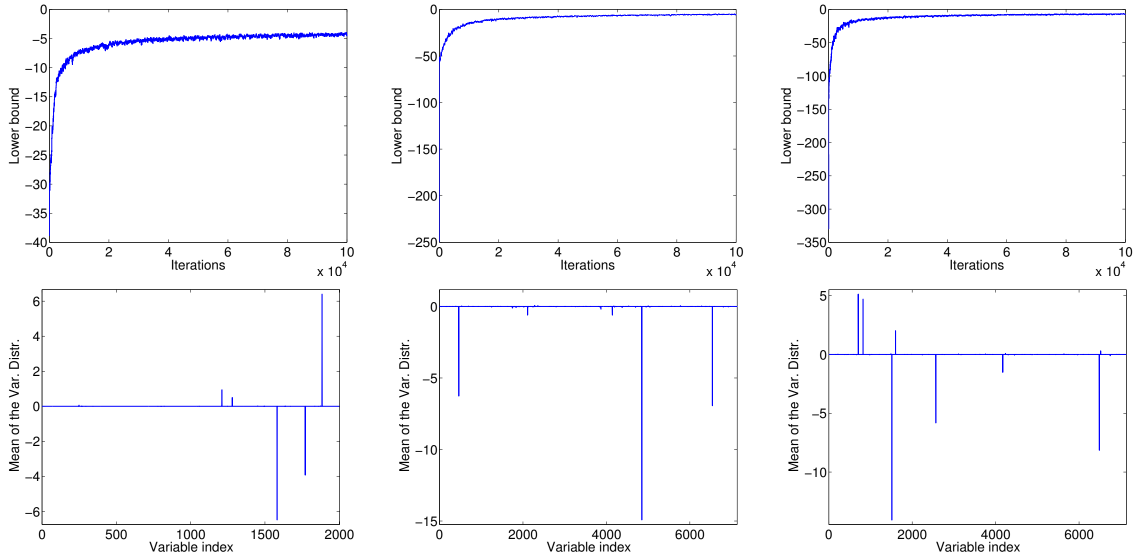

*Figure 3.* Top: Rolling-window average (see supplementary material) of the instantaneous lower bound values. Bottom: Final value of the approximate mean vectors $\boldsymbol{\mu}$. First column corresponds to `Colon`, second to `Leukemia` and third to `Breast` dataset.

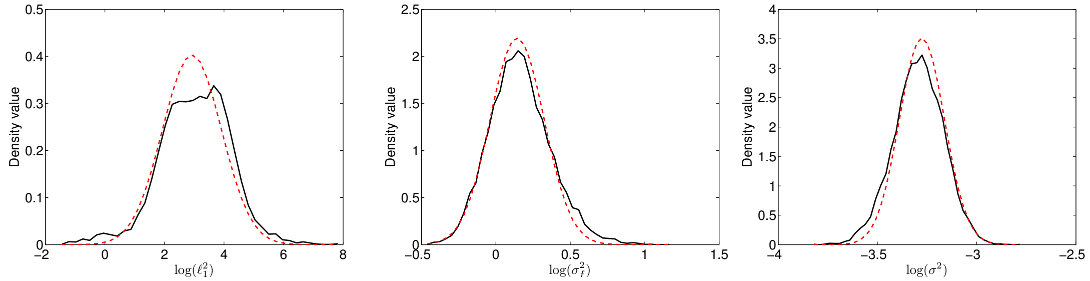

*Figure 4.* Marginal variational Gaussian distributions for some hyperparameters in `Boston` dataset (shown as dashed red lines). The black solid lines show the ground-truth empirical estimates for these marginals obtained by MCMC.

ity of a simulated mechanical pendulum must be predicted.

Figure 4 displays variational posterior marginal distributions for three of the hyperparameters in the Boston housing dataset together with the corresponding empirical marginals obtained by long MCMC runs. Clearly, the variational marginals match very closely the MCMC estimates; see the supplementary material for a complete set of such figures for all hyperparameters in all three regression datasets. Furthermore, negative log-predictive densities (nlpd) as well as standardised mean square errors (smse) in test data are shown in Table 5 for maximum marginal likelihood model selection (ML-II, the standard for GPs), DSVI and MCMC. As the table shows, ML-II, which is the most widely used method for hyperparameter selection in GPs, overfits the `Bodyfat` data set. DSVI and MCMC do not show this problem, yielding much better test performance. To provide an intuition of the computational effort associated to each of these methods, note that on these experiments, on average ML-II took 40 seconds, DSVI 30 minutes and MCMC 20 hours. Further details on all above GP regression experiments, including the learning rates used, are given in the supplementary material.

## 4. Discussion and future work

We have presented a stochastic variational inference algorithm that utilises gradients of the joint probability density and it is based on double stochasticity (by both subsampling training data and simulating from the variational density) to deal with non-conjugate models and big datasets. We have shown that the method can be applied to a number of diverge cases achieving competitive results. Further work should be concerned with speeding the stochastic approximation algorithm as well as fitting more complex variational distributions such as mixture models.

### Acknowledgments

MKT greatly acknowledges support from "Research Funding at AUEB for Excellence and Extroversion, Action 1: 2012-2014". MLG gratefully acknowledges support from Spanish CICYT TIN2011-24533.

## References

Barber, D. and Bishop, C. M. Ensemble learning in Bayesian neural networks. In Jordan, M., Kearns, M., and Solla, S. (eds.), *Neural networks and machine learning*, pp. 215–237, Berlin, 1998.

Bottou, Léon. Online Algorithms and Stochastic Approximations. In *Online Learning and Neural Networks*. Cambridge University Press, 1998.

Bottou, Léon and Bousquet, Olivier. The tradeoffs of large scale learning. In *NIPS*, volume 20, pp. 161–168, 2008.

Challis, Edward and Barber, David. Concave gaussian variational approximations for inference in large-scale bayesian linear models. In *AISTATS*, pp. 199–207, 2011.

Challis, Edward and Barber, David. Gaussian kullback-leibler approximate inference. *J. Mach. Learn. Res.*, 14 (1):2239–2286, January 2013.

Hoffman, Matthew D., Blei, David M., and Bach, Francis R. Online learning for latent dirichlet allocation. In *NIPS*, pp. 856–864, 2010.

Hoffman, Matthew D., Blei, David M., Wang, Chong, and Paisley, John William. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1): 1303–1347, 2013.

Honkela, A., Raiko, T., Kuusela, M., Tornio, M., and Karhunen, J. Approximate Riemannian conjugate gradient learning for fixed-form variational Bayes. *Journal of Machine Learning Research*, 11:3235–3268, 2011.

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

Kingma, Diederik P. and Welling, Max. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

Mnih, Andriy and Gregor, Karol. Neural variational inference and learning in belief networks. In *The 31st International Conference on Machine Learning (ICML 2014)*, 2014.

Neal, Radford M. and Hinton, Geoffrey E. A view of the em algorithm that justifies incremental, sparse, and other variants. In Jordan, Michael I. (ed.), *Learning in Graphical Models*, pp. 355–368. 1999.

Opper, M. and Archambeau, C. The variational Gaussian approximation revisited. *Neural Computation*, 21(3): 786–792, 2009.

Paisley, John William, Blei, David M., and Jordan, Michael I. Variational bayesian inference with stochastic search. In *ICML*, 2012.

Ranganath, Rajesh, Gerrish, Sean, and Blei, David. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 814822, 2014.

Rasmussen, C.E. and Williams, C.K.I. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, 2006.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *The 31st International Conference on Machine Learning (ICML 2014)*, 2014.

Robbins, Herbert and Monro, Sutton. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Robert, Christian P. and Casella, George. *Monte Carlo Statistical Methods*. Springer-Verlag, 1 edition, August 1999.

Seeger, Matthias. Bayesian model selection for support vector machines, gaussian processes and other kernel classifiers. In *NIPS 12*, pp. 603–609, 1999.

Shevade, Shirish Krishnaj and Keerthi, S. Sathiya. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19(17):2246–2253, 2003.

Staines, Joe and Barber, David. Variational optimization. Technical report, 2012.

Tipping, Michael E. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

Wainwright, Martin J. and Jordan, Michael I. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, January 2008.