Natural Gradients in Practice: Non-Conjugate Variational Inference in Gaussian Process Models

Hugh Salimbeni
Imperial College London, PROWLER.io

Stefanos Eleftheriadis
PROWLER.io

James Hensman
PROWLER.io

Abstract

The natural gradient method has been used effectively in conjugate Gaussian process models, but the non-conjugate case has been largely unexplored. We examine how natural gradients can be used in non-conjugate stochastic settings, together with hyperparameter learning. We conclude that the natural gradient can significantly improve performance in terms of wall-clock time. For ill-conditioned posteriors the benefit of the natural gradient method is especially pronounced, and we demonstrate a practical setting where ordinary gradients are unusable. We show how natural gradients can be computed efficiently and automatically in any parameterization, using automatic differentiation. Our code is integrated into the GPflow package.

1 Introduction

Minimizing the Kullback-Leibler (KL) divergence between an unknown and a tractable parametric distribution is the central task of variational inference. In the non-conjugate case, the prevalent approach is to optimize the objective using (stochastic) gradient descent or variants. Gradient descent based methods require careful tuning to work effectively and are prone to poor convergence when the Hessian at the solution is ill-conditioned (Boyd and Vandenberghe, 2004). Ill-conditioning is a problem especially encountered in kernel methods (Ma and Belkin, 2017). A further problem is that the step size is not dimensionless but its units are in the square of the parameters. The appropriate step size is tightly coupled with the parameterization, and no best step size can exist for all problems.
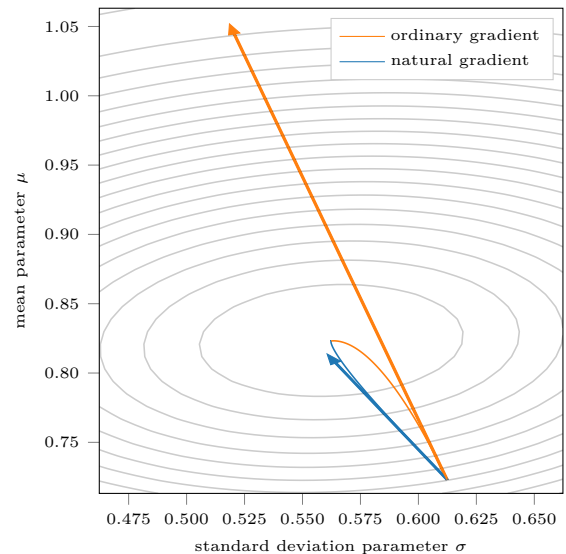
Figure 1: The natural gradient (blue arrow) is correctly scaled and points in a better direction than the ordinary gradient (orange arrow). The contours show the lower bound to a GP with a Bernoulli likelihood and variational posterior with a single Gaussian inducing point. The path followed by taking very small steps in the ordinary gradient (orange curve) ascends the contours. The path taking small steps in the natural gradient (blue curve) is independent of parameterization, and does not follow the contours.

The ordinary gradient turns out to be an unnatural direction to follow for variational inference since we are optimizing a *distribution*, rather than a set of parameters directly. One way to define the gradient is the direction that achieves maximum change subject to a perturbation within a small euclidean ball. To see why the euclidean distance is an unnatural metric for probability distributions, consider the two Gaussians $\mathcal{N}(0, 0.1)$ and $\mathcal{N}(0, 0.2)$, compared to $\mathcal{N}(0, 1000.1)$ and $\mathcal{N}(0, 1000.2)$. The former pair are different and the latter similar, yet in euclidean distance they are equally far apart in the mean and variance. Using the precision in place of the variance gives the opposite result, yet the distributions are unchanged. There is a fundamental mismatch between the ordinary gradient and the objective function: the gradient is dependent on

parameterization whereas the objective function is not.

Fortunately there is a way to solve the disparity: the natural gradient. The natural gradient can be defined as the direction that achieves maximum change in KL divergence. It is well known that paths following the natural gradient are invariant to reparameterization (see e.g., Martens, 2014), and that the natural gradient direction is the ordinary gradient rescaled by the inverse Fisher information matrix (Amari, 1998). Fig. 1 shows a two parameter example comparing the natural gradient to the ordinary gradient. In this case we see that the natural gradient points in a better direction than the ordinary gradient, and also has an appropriate scale.

To investigate whether the advantages suggested by Fig. 1 hold in practice, we consider several aspects in turn. We begin by comparing the different gradients across several different parameterizations (§5.1). To achieve this we first demonstrate how natural gradients can be calculated efficiently and without any cumbersome derivations. Through empirical investigation we show that the natural gradient is indeed a more effective direction to follow in all parameterizations, and also that there is an appropriate step size to use after an initial phase. Using these insights we propose a gradient descent algorithm for the common situation where the size of the dataset forces us to subsample the data, leading to stochastic gradients (§ 5.2). We then extend the approach to hyperparameter optimization using a double loop algorithm that outperforms the state-of-the-art Adam optimizer in wall-clock time (§5.3). Finally, we demonstrate a situation where natural gradients are essential for successful optimization, due to ill-conditioning (§5.4).

Below we summarise our contributions:

- We compare natural gradients to other state-of-the-art techniques in non-conjugate problems, explicitly comparing the influence of different parameterizations and step size on performance.

- We show how natural gradients in the exponential family can be computed efficiently and automatically in any parameterization, using automatic differentiation.

- We show that natural gradients can be used in conjunction with hyperparameter learning in the stochastic setting.

- We highlight a situation where the current approaches fail due to ill-conditioning, and show that natural gradients can solve this problem.

- We provide an implementation of our methods within the GPflow (Matthews et al., 2017) package.

2 Background

In this section we introduce the relevant background on the exponential family, variational inference, and optimisation approaches. We then define natural gradients and show how they take a simple form for the exponential family.

2.1 Preliminaries

We consider the problem of performing inference in a model of the form

$$p(\mathbf{y}, \mathbf{u}) = \left[\prod_{n=1}^{N} p(y_n | \mathbf{u}) \right] p(\mathbf{u}), \qquad (1)$$

where y_n are observed and $\mathbf{u}$ unobserved. Both the prior and likelihood may additionally depend on hyperparameters, but we have omitted these from the notation to reduce clutter. We assume that exact inference in (1) is intractable and make use of an approximate posterior $q(\mathbf{u}; \boldsymbol{\theta})$ in the exponential family. The exponential family is defined as

$$\log q(\mathbf{u}; \boldsymbol{\theta}) = \log h(\mathbf{u}) + \boldsymbol{\theta}^\top \mathbf{t}(\mathbf{u}) - A(\boldsymbol{\theta}), \qquad (2)$$

where $\mathbf{t}(\mathbf{u})$ is the sufficient statistics vector, $A(\boldsymbol{\theta})$ is the log normalizing constant and $h(\cdot)$ is a base measure. The parameterization used in (2) is known as the *natural* parameterization and $\boldsymbol{\theta}$ are the natural parameters. We can instead use an alternative smooth invertible parameterization $\boldsymbol{\xi} \equiv \boldsymbol{\xi}(\boldsymbol{\theta})$. We denote transformation between parameterizations through overloading notation, e.g., the inverse mapping back to the natural parameterization is $\boldsymbol{\theta}(\boldsymbol{\xi})$, and we also abbreviate $q(\mathbf{u}; \boldsymbol{\theta}(\boldsymbol{\xi}))$ as $q(\mathbf{u}; \boldsymbol{\xi})$. A parameterization of particular interest, known as the *expectation* parameterization, is defined as $\boldsymbol{\eta} \equiv \mathbb{E}_{q(\mathbf{u}; \boldsymbol{\theta})} \mathbf{t}(\mathbf{u})$. An important property of the exponential family is that the gradient of the log normalizer is equal to the expectation parameter: $\nabla_{\boldsymbol{\theta}} A(\boldsymbol{\theta}) = \boldsymbol{\eta}^\top$. This can be readily identified from differentiating (2) with respect to $\boldsymbol{\theta}$ and taking expectations.[1] Differentiating (2) again, it follows that the Hessian of the log density is a Jacobian:

$$\nabla_{\boldsymbol{\theta}}^2 \log q(\mathbf{u}; \boldsymbol{\theta}) = -\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\theta}}. \qquad (3)$$

Variational inference proceeds by minimizing the KL divergence from $q(\mathbf{u}; \boldsymbol{\xi})$ to the intractable posterior $p(\mathbf{u}|\mathbf{y})$, or equivalently maximizing the evidence lower bound (ELBO):

$$\mathcal{L}(\boldsymbol{\xi}) = \mathbb{E}_{q(\mathbf{u}; \boldsymbol{\xi})} \sum_{n=1}^{N} \log p(y_n | \mathbf{u}) - \mathrm{KL}[q(\mathbf{u}; \boldsymbol{\xi}) || p(\mathbf{u})]. \quad (4)$$

---

[1] Differentiating (2): $\nabla_{\boldsymbol{\theta}} \log q(\mathbf{u}; \boldsymbol{\theta}) = \mathbf{t}(\mathbf{u})^\top - \nabla_{\boldsymbol{\theta}} A(\boldsymbol{\theta})$. Taking expectations: $\mathbb{E}_{q(\mathbf{u}; \boldsymbol{\xi})} \nabla_{\boldsymbol{\theta}} \log q(\mathbf{u}; \boldsymbol{\theta}) = \boldsymbol{\eta}^\top - \nabla_{\boldsymbol{\theta}} A(\boldsymbol{\theta})$. The score has expectation zero, so the result follows.

Our fundamental problem is to minimize $-\mathcal{L}(\boldsymbol{\xi})$. All the approaches we consider find a sequence of parameters $\{\boldsymbol{\xi}_t\}_{t=0}^{T}$ using the iterative update

$$\boldsymbol{\xi}_{t+1} = \boldsymbol{\xi}_t - \gamma_t \mathbf{P}_t^{-1}\mathbf{g}_t, \quad \mathbf{g}_t = \nabla_{\boldsymbol{\xi}^\top}\mathcal{L}\big|_{\boldsymbol{\xi}=\boldsymbol{\xi}_t}, \quad (5)$$

where γ_t denotes the *step size* and $\mathbf{P}_t^{-1}\mathbf{g}_t$ the *direction*.

2.2 Optimization approaches

Gradient descent (GD). The simplest approach, known as gradient descent, is to set $\mathbf{P}$ to the identity matrix. The step size can be fixed, decayed, or found by a line search on each iteration.

Adam. A more sophisticated approach is to use a diagonal matrix for $\mathbf{P}$, with diagonal elements given by $(\sqrt{v_i} + \epsilon)^{-1}m_i$, where m_i and v_i are the bias corrected exponential moving averages of $[\mathbf{g}_t]_i$ and $([\mathbf{g}_t]_i)^2$. This approach is called Adam (Kingma and Ba, 2015).

LBFGS. One way of interpreting the update (5) is to identify the term $\mathbf{P}_t^{-1}\mathbf{g}_t$ as a minimizer of the local quadratic approximation $\mathcal{L}(\boldsymbol{\xi}_t + \boldsymbol{\delta}) \approx L(\boldsymbol{\xi}_t) + \mathbf{g}^\top\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^\top\mathbf{P}\boldsymbol{\delta}$. Under this interpretation, a natural choice for $\mathbf{P}$ is the Hessian so that the quadratic approximation coincides with the second order Taylor expansion. This is known as Newton's method. Due to the large computational expense of calculating and inverting the Hessian we do not consider it further. Instead, we use a low rank approximation to the Hessian computed from finite differences. Specifically we will compare to a common variant of this algorithm known as LBFGS (Byrd et al., 1995). This algorithm cannot be used in the stochastic setting as finite difference calculations are not robust to noise.

Natural gradient descent (NGD). Another way of interpreting the update (5) is to use the fact that the direction of steepest descent with respect to a norm $||\boldsymbol{\delta}||_\mathbf{A} = \boldsymbol{\delta}^T\mathbf{A}\boldsymbol{\delta}$ is given by $\mathbf{A}^{-1}\nabla_{\boldsymbol{\xi}^\top}\mathcal{L}$.[2] Identifying $\mathbf{P}$ with $\mathbf{A}$, the update (5) corresponds to the steepest descent with respect to the norm induced by the matrix $\mathbf{P}$. Gradient descent (where $\mathbf{P}$ is the identity and the induced metric is Euclidean) can therefore be seen as moving in the direction that maximizes the change in objective with respect to the euclidean norm of the parameters. The Euclidean norm is an unnatural way to compare two parameter vectors if the parameters correspond to *distributions*, however. If instead we consider the KL divergence between two distributions and take the small perturbation limit, we obtain $\mathrm{KL}[q(\mathbf{u};\boldsymbol{\xi}), q(\mathbf{u};\boldsymbol{\xi} + \boldsymbol{\delta})] = \frac{1}{2}\boldsymbol{\delta}^\top\left[\mathbb{E}_{q(\mathbf{u};\boldsymbol{\xi})}\nabla_{\boldsymbol{\xi}}^2\log q(\mathbf{u};\boldsymbol{\xi})\right]\boldsymbol{\delta} + \mathcal{O}(||\boldsymbol{\delta}||^3)$. Therefore, in a sufficiently small neighbourhood the KL divergence induces a quadratic norm with curvature given by the expected Hessian of the log density. This matrix is known as the Fisher information $\mathbf{F}_{\boldsymbol{\xi}}$,

$$\mathbf{F}_{\boldsymbol{\xi}} = -\mathbb{E}_{q(\mathbf{u};\boldsymbol{\xi})}\nabla_{\boldsymbol{\xi}}^2\log q(\mathbf{u};\boldsymbol{\xi}). \quad (6)$$

The direction of steepest descent with respect to this norm is called the natural gradient $\tilde{\nabla}_{\boldsymbol{\xi}}\mathcal{L}$, given by the gradient scaled by the inverse Fisher information: $\tilde{\nabla}_{\boldsymbol{\xi}}\mathcal{L} = (\nabla_{\boldsymbol{\xi}}\mathcal{L})\mathbf{F}_{\boldsymbol{\xi}}^{-1}$ (Amari, 1998).

For the exponential family the Fisher information takes a particularly simple form in the natural parameters. Using (3) we have that $\mathbf{F}_{\boldsymbol{\theta}} = \frac{\partial\boldsymbol{\eta}}{\partial\boldsymbol{\theta}}$. Using the chain rule, we see that the natural gradient in the natural parameters is given by $\tilde{\nabla}_{\boldsymbol{\theta}}\mathcal{L} = \frac{\partial\mathcal{L}}{\partial\boldsymbol{\eta}}$. This expression was used by Hensman et al. (2013) to compute natural gradients in the conjugate case.

To find the natural gradients in some other paramterization we can use the chain rule to obtain

$$\mathbf{F}_{\boldsymbol{\xi}} = \left(\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{\xi}}\right)^\top \frac{\partial\boldsymbol{\eta}}{\partial\boldsymbol{\theta}}\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{\xi}}. \quad (7)$$

This expression was used directly in (Malagò and Pistone, 2015) and (Sun et al., 2009) in a certain parameterization of the Gaussian. The calculation is extremely cumbersome and requires a careful recursive implementation. In the next section we show how to compute the natural gradient efficiently and automatically.

3 Efficient computation

Since all the parameterizations are invertible (and the inverse of the Jacobian is the Jacobian of the inverse), we have

$$\tilde{\nabla}_{\boldsymbol{\xi}}\mathcal{L} = \frac{\partial\mathcal{L}}{\partial\boldsymbol{\xi}}\left(\left(\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{\xi}}\right)^\top\frac{\partial\boldsymbol{\eta}}{\partial\boldsymbol{\theta}}\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{\xi}}\right)^{-1} \quad (8)$$

$$= \frac{\partial\mathcal{L}}{\partial\boldsymbol{\xi}}\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{\theta}}\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{\eta}}\left(\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{\theta}}\right)^\top. \quad (9)$$

Applying the chain rule and transposing, we obtain

$$\tilde{\nabla}_{\boldsymbol{\xi}^\top}\mathcal{L} = \frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{\theta}}\frac{\partial\mathcal{L}}{\partial\boldsymbol{\eta}^\top}. \quad (10)$$

We recognise (10) as a Jacobian-vector product, which is exactly what is computed in *forward-mode* differentiation. Forward-mode automatic differentiation libraries are perhaps less common than reverse-mode, but fortunately there is an elegant way to achieve forward-mode automatic differentiation using reverse-mode differentiation twice (Townsend et al., 2017). See the supplementary material for details on this trick. Importantly, this

---

[2]This can be seen by minimizing $\frac{1}{\epsilon}\mathcal{L}(\boldsymbol{\xi} + \boldsymbol{\delta})$ subject to the constraint that $||\boldsymbol{\delta}||_\mathbf{A} = \epsilon$ and letting $\epsilon \to 0$.

indirect computation only costs negligibly more than the forward pass $\boldsymbol{\xi}(\boldsymbol{\theta})$. The extra computation comes from the parameter conversion between $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$, which is $\mathcal{O}(M^3)$ for the (full rank) Gaussian for the six parameterizations we consider in the next section, where M is the dimension of $\mathbf{u}$. Note that direct inversion of the Fisher information for the Gaussian would be cubic in the number of parameters, i.e., $\mathcal{O}((M + M^2)^3)$. In practice, we find this increases the computation relative to the ordinary gradient by a factor of about 1.5. We emphasize that this approach requires no more code than the parameter transformation, so new parameterizations can be easily investigated.

4 Specific application: Sparse Gaussian processes

What we have described so far applies to any model and any exponential family variational posterior. We now present a specific example: a sparse Gaussian process (GP) model with a Gaussian variational posterior. For a comprehensive overview see (Matthews et al., 2016).

The model takes the form of (1). Each y_i is associated with a D-dimensional input $\mathbf{x}_i \in \mathbb{R}^D$. We place a GP prior on the unobserved variables $f(\mathbf{x}_i)$,

$$f(\cdot) \sim \mathcal{GP}(\mu, k), \qquad (11)$$

where μ and k are mean and covariance functions. That is, any collection of function values $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)$ are jointly Gaussian with mean $\mu(\mathbf{x}_i)$ and covariance $k(\mathbf{x}_i, \mathbf{x}_j)$, for $i, j = 1, \ldots, N$. Inference in this model scales cubically in N, and is intractable when the likelihood is not Gaussian, so we proceed with variational inference. We choose a Gaussian process for the posterior with the special property that it matches the prior conditioned on a number of inducing points $\mathbf{u} = [f(\mathbf{z}_i)]_{i=1}^M$. We use a directly parameterized Gaussian for $q(\mathbf{u})$. The posterior leads to the bound,

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{u})} \sum_{i=1}^N \log \tilde{p}(y_i|\mathbf{u}) - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})], \qquad (12)$$

where $\log \tilde{p}(y_i|\mathbf{u}) = \mathbb{E}_{q(f_i|\mathbf{u})} \log p(y_i|f_i)$. Since both expectations are over Gaussians they combine to a single expectation with mean and variance available in closed form. The univariate expectation of the likelihood can be found with Gauss-Hermite quadrature, or exactly in some cases. The bound (12) can be evaluated stochastically by evaluating a random subset of terms in the sum and scaling the KL term appropriately.

Parameterizations of the Gaussian. We now present the different parameterizations we will use of the Gaussian variational distribution $q(\mathbf{u})$. The Gaussian is a member of the exponential family with the sufficient statistic vector given by $\mathbf{t}(\mathbf{u}) = [\mathbf{u}, \mathrm{vec}(\mathbf{u}\mathbf{u}^\top)]$, so

that $\boldsymbol{\theta}^\top \mathbf{t}(\mathbf{u}) = \mathbf{u}^\top \boldsymbol{\theta}_1 + \mathbf{u}^\top \boldsymbol{\Theta}_2 \mathbf{u}$, where $\boldsymbol{\theta}_1$ is the first D elements of $\boldsymbol{\theta}$, and $\boldsymbol{\Theta}_2$ are remaining elements reshaped to a square matrix. We refer to this as the *unpacked form*. A common parameterization of the Gaussian is in terms of the mean ($\mathbf{m}$) and variance ($\mathbf{S}$). The unpacked natural parameters are given by $\mathbf{S}^{-1}\mathbf{m}, -\frac{1}{2}\mathbf{S}^{-1}$ and the expectation parameters by $\mathbf{m}, \mathbf{S} + \mathbf{m}\mathbf{m}^\top$. Converting between these parameterizations is straightforward and has complexity $\mathcal{O}(M^3)$.

We consider six parameterizations of the Gaussian. Perhaps the most commonly used in variational inference (e.g., Dai et al., 2015; Challis and Barber, 2011) is the mean and square root of the covariance: $\mathbf{m}, \mathbf{L}$, with $\mathbf{L}\mathbf{L}^T = \mathbf{S}$. We refer to this as the `mean/var-sqrt` parameterization. Another way to constrain the covariance to be positive definite is to use the matrix log of the covariance (e.g., Glasmachers et al., 2010) $\mathbf{m}, \mathbf{L}$, with $\exp(\mathbf{L}) = \mathbf{S}$, where exp here is the matrix exponential (the `mean/var-log` parameterization). We use additionally the unconstrained mean and variance parameters $\mathbf{m}, \mathbf{S}$ (the `mean/var` parameterization) and the unconstrained natural parameters $\boldsymbol{\theta}_1, \boldsymbol{\Theta}_2$ (the `natural` parameterization). For completeness we also constrain the natural parameters via the square root and log transformations. Since $\boldsymbol{\Theta}_2$ is negative definite we use $\boldsymbol{\theta}_1, \mathbf{L}$ with $\mathbf{L}\mathbf{L}^T = -\boldsymbol{\Theta}_2$ (the `natural-sqrt` parameterization), and, finally, $\boldsymbol{\theta}_1, \mathbf{L}$ with $\exp(\mathbf{L}) = -\boldsymbol{\Theta}_2$ (the `natural-log` parameterization).

5 Natural gradients in practice

In this section we investigate NGD for large step sizes. We aim to provide evidence to answer the following:

1. Is the natural gradient a good *direction*, irrespective of step size?

2. Can we easily choose an effective step size?

3. Are natural gradients useful when combined with hyperparameter optimization?

We use a running example of three common datasets with different likelihoods: energy efficiency (ENERGY, $N = 784, D = 8$) with a Gaussian likelihood, boston housing (BOSTON, $N = 506, D = 14$) with a student-t likelihood, and pima Indians diabetes (PIMA, $N = 784, D = 8$) with a Bernoulli likelihood. We use 100 inducing points initialized with k-means and the Matern ($\nu = {}^5/_2$) kernel. Details of hyperparameters and data preprocessing are in the supplementary material.

5.1 Deterministic case

To investigate the quality of direction, we apply NGD, GD and Adam each with a line search to find the γ
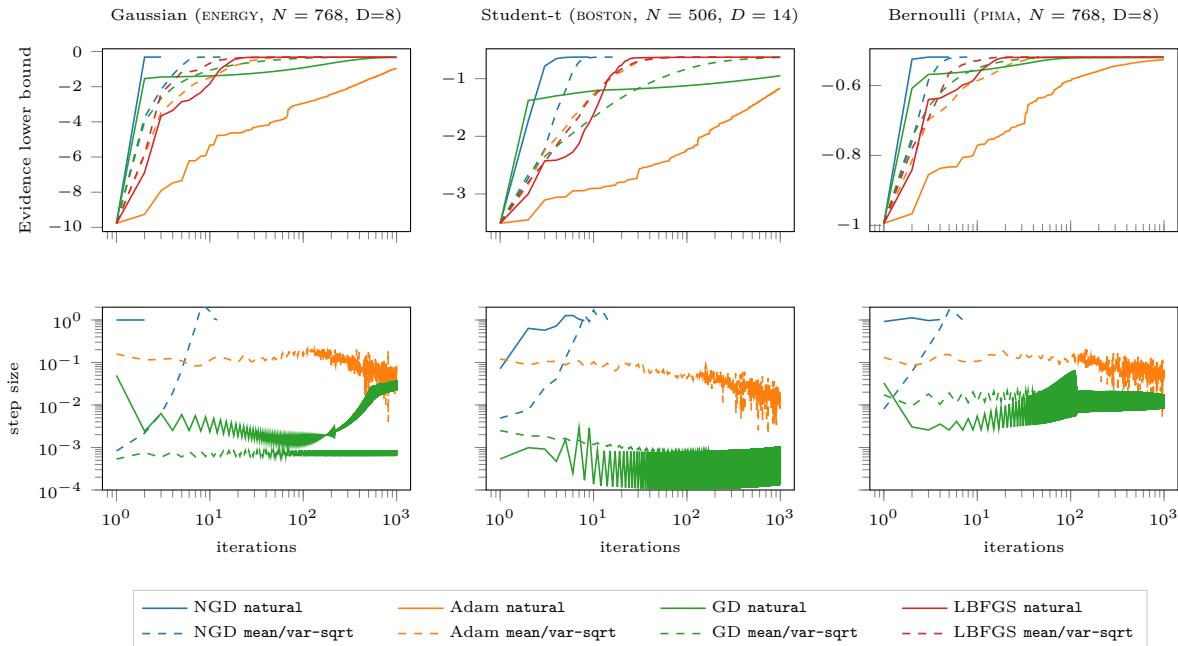
Figure 2: The natural gradient is a superior *direction* in both parameterizations, and the best step size increases during optimization to $\gamma \approx 1$. Upper row: optimization methods, all with a line search for the step size. Lower row: the step sizes used at each iteration. Additional figures in the supplementary material confirm these claims with four further parameterizations and multiple splits.

that achieves maximum value of the objective at each step. We run an exhaustive search for γ using the Brent (Brent, 1971) method until convergence. We compare also to LBFGS which includes a line search. Plots are shown in the supplementary material for experiments using five splits of 90% for each of the six parameterizations defined in §4. Fig. 2 shows a representative split with the `mean/var-sqrt` and `natural` parameterizations. For the case of the Gaussian likelihood we observe the optimal solution is found in a single step of $\gamma = 1$, as shown in Hensman et al. (2013). For the other parameterizations the initial natural gradient step size is a small value that is parameterization and likelihood dependent, but then increases to $\gamma = 1$. Once the step size has increased to near $\gamma = 1$ we observe extremely rapid convergence.

The natural gradient direction achieves faster convergence for all likelihoods and parameterizations (see supplementary material for the other 4 parameterizations). For all different parameterizations and likelihoods we see that the best step size for the natural gradient *increases* to $\gamma = 1$. This is in contrast to the ordinary gradient where the step size differs between likelihoods and generally needs to *decrease* as optimization progresses. For Adam the direction is elementwise rescaled and a value close to 0.1 seems appropriate for the constrained parameterizations, but for the unconstrained parameterizations Adam cannot make good progress.

In summary we have provided evidence that the natural gradient is indeed a better direction, and increasing the step size to $\gamma = 1$ is appropriate for fast convergence. We see also that the best combination of optimization method and parameterization is `natural` for NGD and `mean/var-sqrt` for GD and Adam. We will use these combinations for all subsequent experiments.

5.2 Stochastic natural gradients

We next consider the stochastic case where a line search is not possible. We introduce stochasticity by subsampling the data into minibatches of size 256. To find a reasonable γ for the Adam and GD methods we performed a search over $\{10^{-k}\}_{k=0}^{6}$. We used the largest rate that remained stable.

We now consider a strategy for γ. Our line search experiments suggest that γ should be gradually increased to some fixed value $\gamma \approx 1$. We therefore propose a simple schedule for NGD: (i) log-linearly increase γ from γ_{initial} to γ_{final} over K iterations; (ii) set $\gamma = \gamma_{\text{final}}$ for the remaining iterations. For the ENERGY, BOSTON and PIMA datasets we found that $\gamma_{\text{initial}} = 10^{-4}$, $\gamma_{\text{final}} = 10^{-1}$ and $K = 5$ were suitable values.

Fig. 3 shows the optimization of the ENERGY, BOSTON and PIMA datasets against wall-clock time with GD, Adam and NGD. We observe that NGD improves on Adam and GD after about 3×10^{-4} seconds (about 3
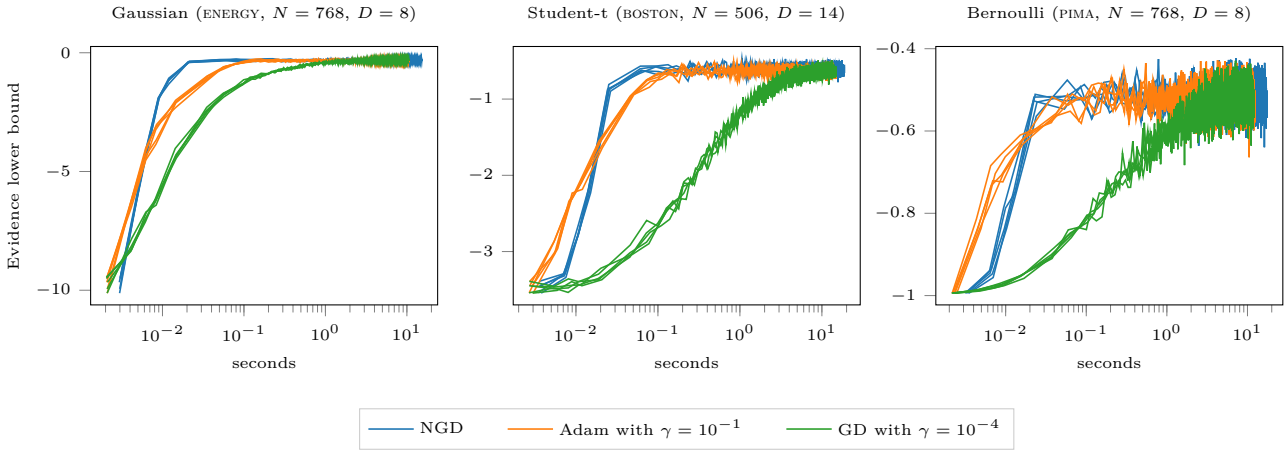
Figure 3: Stochastic optimization of the lower bound for fixed hyperparameters. The batch size is 256 and 5000 iterations are shown for five splits.
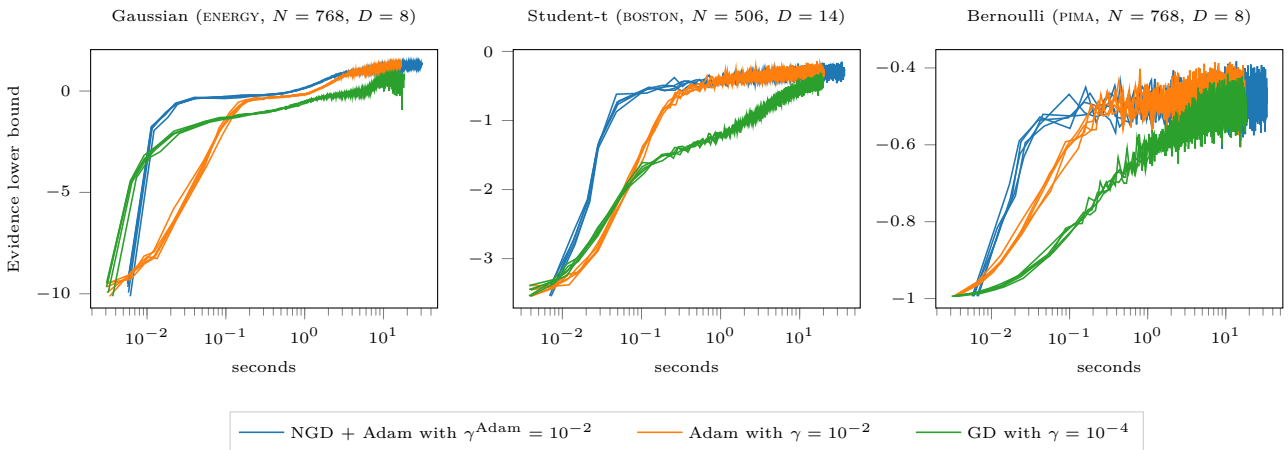

Figure 4: Joint optimization of the hyperparameters and the variational distribution. For natural gradients, a step of NGD on the variational parameters is alternated with a step of Adam on the hyperparameters. For Adam and GD the variational and hyperparameters are optimized together in a single objective. The batch size is 256 and 5000 iterations are shown for five splits.
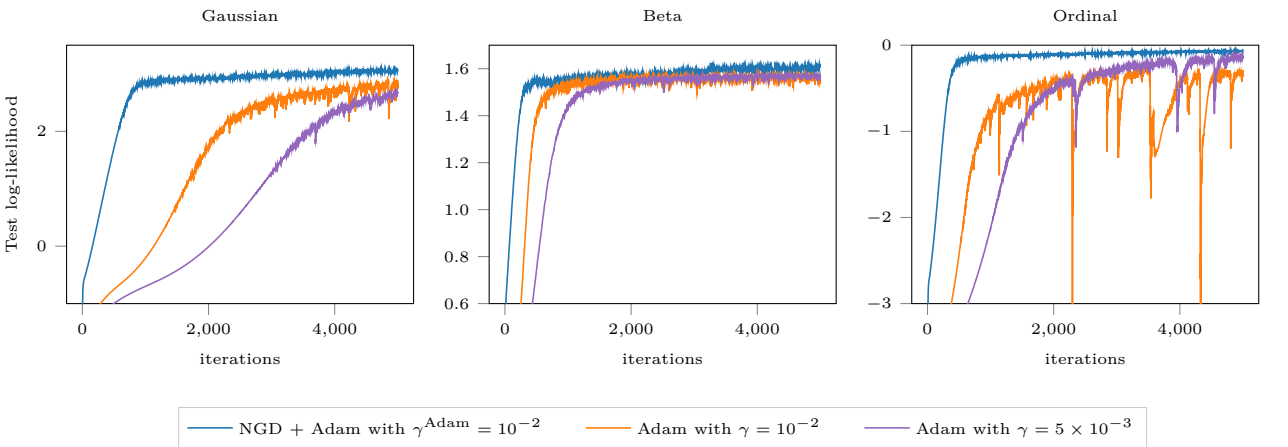

Figure 5: Optimization of the NAVAL dataset ($N = 11\text{K}$, $D = 16$), with three different likelihoods. The ill-conditioning of the variational distributions renders the optimization using ordinary gradients extremely difficult, even given a large number of iterations and different values for the Adam learning rate. The batch size is 256 and 5000 iterations are shown for a single split.

iterations). The advantages we see in the deterministic case appear to be realised in the stochastic setting.

5.3 Hyperparameters

An advantage of variational inference is that the ELBO can be optimized with respect to hyperparameters (we include also the inducing point inputs $\{\mathbf{z}_i\}_{i=1}^M$) as a proxy for the true marginal likelihood. Note that this is biased as the slack in the bound may depend on hyperparameter settings (Turner and Sahani, 2011). Nevertheless, it has been found to work well in practice, so a prevalent approach is to optimize the hyperparameters and variational parameters together in a single objective. We cannot use natural gradients directly for the hyperparameters as we do not have a probability distribution for them. Instead, we use an alternating scheme where we perform a step of Adam on the hyperparameters (with step size γ^{Adam}), followed by a step of NGD on the variational parameters (with step size γ) We refer to this hybrid method as NGD+Adam and apply this approach to the same three datasets, using the same schedule for γ as before. We compare to optimizing the variational distribution and hyperparmeters in a single objective using Adam and GD.

Fig. 4 shows the results of stochastic optimization of the variational distribution together with hyperparameters. We see that NGD+Adam outperforms the other three methods in terms of wall-clock time.

5.4 When natural gradients are *essential*

In this section we present a practical situation where natural gradients are *essential*. The previous experiments demonstrated settings where all approaches could find the same solution in a reasonable time. This is not always the case, however, and we present a setting where the natural gradient approach can find a better solution than any method using ordinary gradient.

In ill-conditioned settings ordinary gradients suffer from instability (Sun et al., 2009) and slow convergence. As the natural gradient is invariant to parameterization, NGD is not adversely effected by issues of conditioning. We consider the commonly used NAVAL dataset, which has target values uniformly distributed in 51 increments between 0.95 and 1. We use this dataset with three different likelihoods: a Gaussian likelihood (rescaling the values to zero mean and unit variance), a single-parameter Beta likelihood[3] (rescaling to [0, 1]) and an ordinal likelihood (rescaling to $0, 1, \ldots, 50$) (Chu and Ghahramani, 2005), with bins uniformly spaced between -2 and 2. For NGD+Adam use a schedule

---

[3]The usual α, β parameters are related by $\alpha = sm, \beta = s(1-m)$, with $m = \sigma(f)$ and $s > 0$ a hyperparameter.

with $\gamma_{\text{initial}} = 10^{-4}$, $\gamma_{\text{final}} = 10^{-1}$ and $K = 40$. We compare to the optimization of the lower bound with respect to hyperparameters and variational parameters using NGD+Adam and Adam.

Fig. 5 shows the optimization progress in terms of test log-likelihood after a large number of iterations. Note that ordinary gradient with Adam cannot achieve the optimal value, even after many iterations and with different step sizes.

5.5 Further Results

In this section we apply NGD+Adam to 4 larger datsets from UCI corpus, using a student-t likelihood, and also the MNIST for multiclass classification. In both settings we find that natural gradients either find the optimal solution more quickly, or enable a solution to be found that cannot be obtained using ordinary gradients alone.

Fig. 6 shows the optimization in the UCI datasets with student-t likelihood, using a minibatch size of 256 and with the same schedule for γ as in the NAVAL experiment. For the KIN8NM, POWER and YEAR datasets we observe significant improvement over Adam.

Fig. 7 shows the result of MNIST multiclass classification using the standard train/test split and a batch size of 1024. The schedule for γ increases log-linearly from 10^{-6} to 0.02 over 2000 iterations. We see that the natural gradient approach outperforms Adam in terms of test loglikelihood.

6 Related work

The first use of natural gradients for variational inference goes back to Sato (2001), where it was shown that for an exponential family conditionally conjugate model (i.e., a model where classical fixed point variational updates can be derived in closed form), the NGD corresponds exactly to the fixed point variational update if $\gamma = 1$. This observation leads to an online version of the fixed point algorithm. This idea was made more explicit in (Hoffman et al., 2013), where it was termed stochastic variational inference and applied to a range of problems. The first example of natural gradients used in the non-conjugate case with Gaussian variational distribution can be found in (Honkela et al., 2010). In this work, natural gradients are used for the variational mean. The inverse Fisher information for the mean has a particularly simple form (it is the precision), but the expression for the covariance is much more complicated and cumbersome to derive directly.

Natural evolution strategies (NES) (Sun et al., 2009) is closely related to variational inference. In NES a fitness function is optimized in expectation under a
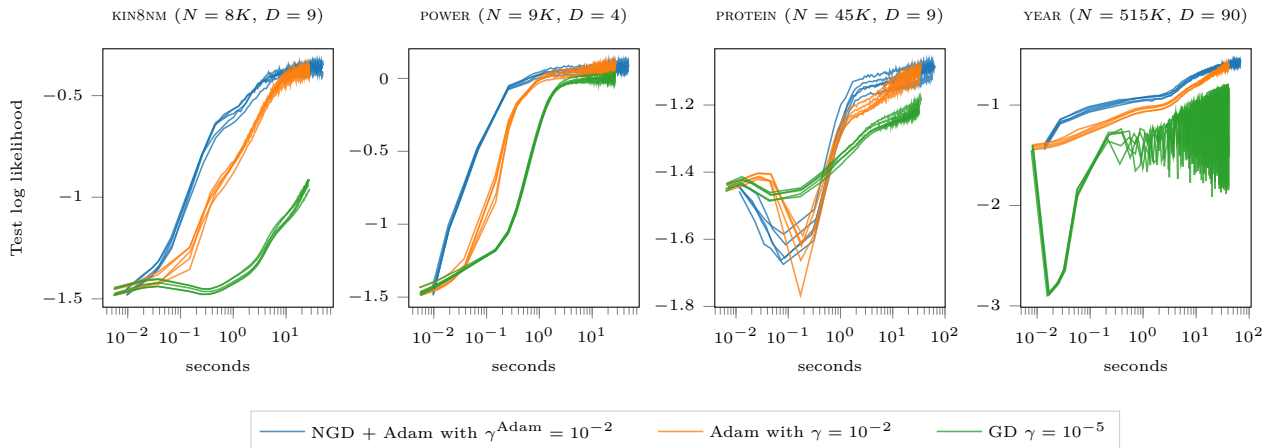
Figure 6: Optimization of hyperparameters and variational distributions for larger UCI datasets with a student-t likelihood.
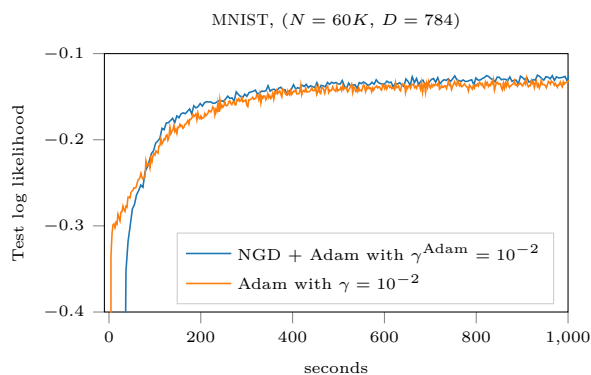

Figure 7: Optimization of the hyperparameters and variational distribution for the MNIST data, with a Robust-max multiclass likelihood. We see that after the first few initial iterations NGD+Adam outperforms Adam alone.

Gaussian. This converges to a zero entropy solution, so ordinary gradients cannot feasibly be used due to the problem of ill-conditioning. Natural gradients are therefore essential for a practical algorithm. In (Sun et al., 2009) the Fisher information for the `mean/var-sqrt` is calculated and inverted directly, which is inefficient. A similar result for the `mean/var` parameterization was presented in (Malagò and Pistone, 2015).

Recently, there have been several works employing natural gradients to approximations of non-conjugate components of a model. In the context of GPs, Khan et al. (2015, 2016) used a linearization of the non-conjugate terms and achieved impressive results. Johnson et al. (2016) use an auxiliary model to learn the approximate natural parameters with neural network likelihood, and then perform analytic updates on the conjugate approximation. Knowles and Minka (2011) use model-specific bounds to take approximate natural gradient steps in a variational message passing setting.

7 Discussion and conclusion

In all cases that we have investigated, we found that natural gradients accelerate convergence relative to methods using the ordinary gradient. In some cases the contrast is so severe that the ordinary gradient can require an unfeasibly large number of iterations to achieve the same results as the natural gradient. In practice, natural gradients are essential for finding a good solution in these situations. The drawbacks of the approach are that a schedule for γ must be specified. The success of the method relies on γ increasing to a reasonably large value (≈ 0.1) sufficiently quickly (< 1000 iterations). If γ needed to be kept small for much longer, then the advantage of the natural gradient method might be lost. Using a probabilistic line search (Mahsereci and Hennig, 2015) for NGD is a promising area for future research.

We have shown that natural gradients are useful for variational inference in non-conjugate sparse Gaussian process models. Natural gradients are particularly advantageous in problems where the ordinary gradient is crippled by the parameterization-dependent ill-conditioning. Such situations exist in practice. We have shown that natural gradients can be computed efficiently and with minimal effort using modern automatic differentiation techniques, and can be combined with modern optimizers such as Adam for hyperparameter learning. We compared six likelihoods and nine benchmark datasets, and found the natural gradient provided improvement in all cases.

Acknowledgements

References

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

Richard P. Brent. An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4):422–425, 1971.

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *Journal on Scientific Computing*, 16(5):1190–1208, 1995.

Edward Challis and David Barber. Concave Gaussian variational approximations for inference in large-scale Bayesian linear models. In *Int'l Conference on Artificial Intelligence and Statistics*, pages 199–207, 2011.

Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6(7):1019–1041, 2005.

Zhenwen Dai, Andreas Damianou, Javier González, and Neil D. Lawrence. Variational auto-encoded deep Gaussian processes. In *Int'l Conference on Learning Representations*, 2015.

Tobias Glasmachers, Tom Schaul, Sun Yi, Daan Wierstra, and Jürgen Schmidhuber. Exponential natural evolution strategies. In *Genetic and Evolutionary Computation*, pages 393–400, 2010.

James Hensman, Nicolò Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, pages 282–290, 2013.

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

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

Matthew Johnson, David Duvenaud, Alex Wiltschko, Ryan P. Adams, and Sandeep R. Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, pages 2946–2954, 2016.

Mohammad E. Khan, Pierre Baqué, François Fleuret, and Pascal Fua. Kullback-Leibler proximal variational inference. In *Advances in Neural Information Processing Systems*, pages 3402–3410, 2015.

Mohammad E. Khan, Reza Babanezhad, Wu Lin, Mark Schmidt, and Masashi Sugiyama. Faster stochastic variational inference using proximal-gradient methods with general divergence functions. In *Uncertainty in Artificial Intelligence*, pages 319–328, 2016.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int'l Conference on Learning Representations*, 2015.

David A. Knowles and Tom Minka. Non-conjugate variational message passing for multinomial and binary regression. In *Advances in Neural Information Processing Systems*, pages 1701–1709, 2011.

Siyuan Ma and Mikhail Belkin. Diving into the shallows: A computational perspective on large-scale shallow learning. In *Advances in Neural Information Processing Systems*, pages 3778–3787, 2017.

Maren Mahsereci and Philipp Hennig. Probabilistic line searches for stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 181–189, 2015.

Luigi Malagò and Giovanni Pistone. Information geometry of the Gaussian distribution in view of stochastic optimization. In *Foundations of Genetic Algorithms*, pages 150–162, 2015.

James Martens. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

Alexander G. de G. Matthews, James Hensman, Richard E. Turner, and Zoubin Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *Int'l Conference on Artificial Intelligence and Statistics*, pages 231–239, 2016.

Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18 (40):1–6, 2017.

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

Yi Sun, Daan Wierstra, Tom Schaul, and Juergen Schmidhuber. Efficient natural evolution strategies. In *Genetic and Evolutionary Computation*, pages 539–546, 2009.

Jamie Townsend, David Duvenaud, and Johnson Matthew. Autograd issue 175. `https://github.com/HIPS/autograd/pull/175`, 2017.

Richard E. Turner and Maneesh Sahani. Two problems with variational expectation maximisation for time-series models. *Bayesian Time Series Models*, pages 115–138, 2011.