Efficient Gaussian Process Classification Using Pólya-Gamma Data Augmentation

Florian Wenzel [*1,2] **Théo Galy-Fajou** [*3] **Christan Donner** [3] **Marius Kloft** [1,2] **Manfred Opper** [3]

Abstract

We propose an efficient stochastic variational approach to GP classification building on Pólya-Gamma data augmentation and inducing points, which is based on closed-form updates of natural gradients. We evaluate the algorithm on real-world datasets containing up to 11 million data points and demonstrate that it is up to three orders of magnitude faster than the state-of-the-art while being competitive in terms of prediction performance.

1. Introduction

Gaussian processes (GPs) (Rasmussen and Williams, 2005) provide a popular Bayesian non-linear non-parametric method for regression and classification. Because of their ability of accurately adapting to data and thus achieving high prediction accuracy, GPs are a standard method in several application areas, including geospatial predictive modeling (Stein, 2012) and robotics (Dragiev et al., 2011).

However, recent trends in data availability in the sciences and technology have made it necessary to develop algorithms capable of processing massive data (John Walker, 2014). New application domains of machine learning—such as personalized medicine (Collins and Varmus, 2015), business intelligence (Chen et al., 2012), and smart urbanism (Kitchin, 2014)—emerged, which come with vast amounts of data.

Currently, GP classification has limited applicability to big data. Naive inference typically scales cubic in the number of data points, and exact computation of posterior and marginal likelihood is intractable.

Nevertheless, the combination of so-called sparse Gaussian process techniques with approximate inference methods, such as expectation propagation (EP) or the variational approach, have enabled GP classification for datasets containing millions of data points (Hensman and Matthews, 2015; Dezfouli and Bonilla, 2015; Hernández-Lobato and Hernández-Lobato, 2016).

While these results are already impressive, we will show in this paper that a speedup of up to three orders magnitudes can be achieved. Our approach is based on replacing the ordinary (stochastic) gradients for optimizing the variational objective function by more efficient *natural gradients*, which recently have been successfully used in a variety of variational inference problems (e.g., Honkela et al., 2010; Wenzel et al., 2017; Jähnichen et al., 2018).

Unfortunately, an efficient computation of the natural gradient for the GP classification problem is not straight forward. The use of the probit link function in Hensman and Matthews (2015); Dezfouli and Bonilla (2015); Hernández-Lobato and Hernández-Lobato (2016) leads to expectations in the variational objective functions that can only be computed by numerical quadrature, thus, preventing efficient optimization.

We derive a natural-gradient approach to variational inference in GP classification based on the *logit* link. We exploit that the corresponding likelihood has an auxiliary variable representation as a continuous mixture of Gaussians involving Pólya-Gamma random variables (Polson et al., 2013). The variational inference updates are based on natural gradients and can be computed in closed-form. Our main contributions are as follows:

- We present an efficient Gaussian process classification model using a logit link function. Our approach is based on Pólya-Gamma data augmentation and inducing points for Gaussian process inference.
- We derive an efficient inference algorithm based on stochastic variational inference and natural gradients. All natural gradient updates are given in closed-form and do not rely on numerical quadrature methods or sampling approaches. Natural gradients have the advantage that they provide effective second-order optimization updates.

[*]Equal contribution [1]Department of Computer Science, Humboldt-Universität zu Berlin, Germany [2]Department of Computer Science, TU Kaiserslautern, Germany [3]Department of Computer Science, Technical University of Berlin, Germany. Correspondence to: Florian Wenzel <wenzelfl@hu-berlin.de>.

- In our experiments, we demonstrate that our approach drastically improves speed up to three orders of magnitude while being competitive in terms of prediction performance. We apply our method to massive real-world datasets up to 11 million points and demonstrate superior scalability.

The paper is organized as follows. In section 2 we discuss related work. In section 3 we introduce our novel scalable GP classification model and in section 4 we present an efficient variational inference algorithm. Section 5 concludes with experiments. Our code is available via Github[1].

2. Background and Related Work

Gaussian process classification Hensman and Matthews (2015) consider Gaussian process classification with a probit inverse link function and suggest a variational model that builds on inducing points. Their approach suffers from the drawback that the updates are not computed in closed-form preventing the application of efficient optimization techniques.

Hernández-Lobato and Hernández-Lobato (2016) follow an expectation propagation approach based on inducing points and have a similar computational cost as Hensman and Matthews (2015).

Dezfouli and Bonilla (2015) propose a general automated variational inference approach for sparse GP models with non-conjugate likelihood. Since they follow a black box approach and do not exploit model specific properties they do not employ efficient optimization techniques.

Naish-Guzman and Holden (2007) also use the concept of inducing points but do not allow for stochastic optimization and cannot be applied on big datasets.

Pólya-Gamma data augmentation Polson et al. (2013) introduced the idea of data augmentation in logistic models using the class of Pólya-Gamma distributions. This allows for exact inference via Gibbs sampling or approximate variational inference schemes (Scott and Sun, 2013).

Linderman et al. (2015) extend this idea to multinomial models and discuss the application for Gaussian processes with multinomial observations but their approach does not scale to big datasets and they do not consider the concept of inducing points.

3. Model

The logit GP Classification model is defined as follows. Let $X = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \in \mathbb{R}^{d \times n}$ be the d-dimensional training points with labels $\boldsymbol{y} = (y_1, \dots, y_n) \in \{-1, 1\}^n$. The likelihood of the labels is

$$p(\boldsymbol{y}|\boldsymbol{f}, X) = \prod_{i=1}^{n} \sigma(y_i f(\boldsymbol{x}_i)), \tag{1}$$

where $\sigma(z) = (1 + \exp(-z))^{-1}$ is the logit link function and f is the latent decision function. We place a GP prior over f and obtain the joint distribution of the labels and the latent GP

$$p(\boldsymbol{y}, \boldsymbol{f}|X) = p(\boldsymbol{y}|\boldsymbol{f}, X)p(\boldsymbol{f}|X), \tag{2}$$

where $p(\boldsymbol{f}|X) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, K_{nn})$ and K_{nn} denotes the kernel matrix evaluated at the training points X. For the sake of clarity we omit the conditioning on X in the following.

3.1. Pólya-Gamma data augmentation

Due to the analytically inconvenient form of the likelihood function, inference for logit GP classification is a challenging problem. We aim to remedy this issue by considering an augmented representation of the original model. Later we will see that the augmented model is indeed advantageous as it leads to efficient closed-form updates in our variational inference scheme.

Polson et al. (2013) introduced the class of Pólya-Gamma random variables and proposed a data augmentation strategy for inference in models with binomial likelihoods. The augmented model has the appealing property that the likelihood of the latent function $\boldsymbol{f}$ is proportional to a Gaussian density when conditioned on the augmented Pólya-Gamma variables. This allows for Gibbs sampling methods, where model parameters and Pólya-Gamma variables can be sampled alternately from the posterior (Polson et al., 2013). Alternatively, the augmentation scheme can be utilized to derive an efficient approximate inference algorithm in the variational inference framework, which will be pursued here.

The Pólya-Gamma distribution is defined as follows. The random variable $\omega \sim \mathrm{PG}(b, 0)$, $b > 0$ is defined by the moment generating function

$$\mathbb{E}_{\mathrm{PG}(\omega|\, b,0)}[\exp(-\omega t)] = \frac{1}{\cosh^b(\sqrt{t/2})}. \tag{3}$$

It can be shown that this is the Laplace transform of an infinite convolution of gamma distributions. The definition is related to our problem by the fact that the logit link can be written in a form that involves the cosh function, namely $\sigma(z_i) = \exp(\frac{1}{2}z_i)(2\cosh(\frac{z_i}{2}))^{-1}$. In the following we derive a representation of the logit link in terms of Pólya-Gamma variables.

First, we define the general $\mathrm{PG}(b, c)$ class which is derived by an exponential tilting of the $\mathrm{PG}(b, 0)$ density, it is given

[1] https://amor.cms.hu-berlin.de/~wenzelfl/x-gpc.html

by

$$\mathrm{PG}(\omega|\, b, c) \propto \exp(-\frac{c^2}{2}\omega)\mathrm{PG}(\omega|\, b, 0).$$

From the moment generating function (3) the first moment can be directly computed

$$\mathbb{E}_{PG(\omega|b,c)}[\omega] = \frac{b}{2c}\tanh\left(\frac{c}{2}\right).$$

Surprisingly, for the subsequently presented variational algorithm these properties suffice and the full representation of the Pólya-Gamma density $\mathrm{PG}(\omega|b, c)$ is not required.

We now adapt the data augmentation strategy based on Pólya-Gamma variables for the GP classification model. To do this we write the non-conjugate logistic likelihood function (1) in terms of Pólya-Gamma variables

$$\begin{aligned}\sigma(z_i) &= (1+\exp(-z_i))^{-1} \\ &= \frac{\exp(\frac{1}{2}z_i)}{2\cosh(\frac{z_i}{2})} \\ &= \frac{1}{2}\int \exp\left(\frac{z_i}{2} - \frac{z_i^2}{2}\omega_i\right) p(\omega_i)\mathrm{d}\omega_i, \end{aligned} \tag{4}$$

where $p(\omega_i) = \mathrm{PG}(\omega_i|1, 0)$ and by making use of (3). For more details see Polson et al. (2013). Using this identity and substituting $z_i = y_i f(x_i)$ we augment the joint density (2) with Pólya-Gamma variables

$$\begin{aligned}p(\boldsymbol{y}, \boldsymbol{\omega}, \boldsymbol{f}) &= p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{\omega})p(\boldsymbol{f})p(\boldsymbol{\omega}) \\ &\propto \exp\left(\frac{1}{2}\boldsymbol{y}^\top\boldsymbol{f} - \frac{1}{2}\boldsymbol{f}^\top\Omega\boldsymbol{f}\right) p(\boldsymbol{f})p(\boldsymbol{\omega}), \end{aligned} \tag{5}$$

where $\Omega = \mathrm{diag}(\boldsymbol{\omega})$ is the diagonal matrix of the Pólya-Gamma variables $\{\omega_i\}$. In contrast to the original model (2) the augmented model is conditionally conjugate forming the basis for deriving closed-form updates in section 4.

Interestingly, employing a structured mean-field variational inference approach (cf. section 4) to the plain Pólya-Gamma augmented model (5) leads to the same bound for GP classification derived by Gibbs and MacKay (2000). This is an interesting new perspective on this bound since they do not employ a data augmentation approach. A proof is provided in appendix A.5. Our approach goes beyond Gibbs and MacKay (2000) by including a sparse GP prior (section 3.2), proposing a scalable inference algorithm based on natural gradients (section 4) and introducing a perturbative correction which builds on the Pólya-Gamma representation (section 4.2).

3.2. Sparse Gaussian process

Inference in GP models typically has the computational complexity $\mathcal{O}(n^3)$. We obtain a scalable approximation of our model and focus on inducing point methods (Snelson and Ghahramani, 2006). We follow a similar approach as in Hensman and Matthews (2015) and reduce the complexity to $\mathcal{O}(m^3)$, where m is number of inducing points.

We augment the latent GP f with m additional input-output pairs $(Z_1, u_1), \ldots, (Z_m, u_m)$, termed as *inducing inputs* and *inducing variables*. The function values of the GP $\boldsymbol{f}$ and the inducing variables $\boldsymbol{u} = (u_1, \ldots, u_m)$ are connected via

$$\begin{aligned}p(\boldsymbol{f}|\boldsymbol{u}) &= \mathcal{N}\left(\boldsymbol{f}|K_{nm}K_{mm}^{-1}\boldsymbol{u}, \widetilde{K}\right) \\ p(\boldsymbol{u}) &= \mathcal{N}\left(\boldsymbol{u}|0, K_{mm}\right), \end{aligned} \tag{6}$$

where K_{mm} is the kernel matrix resulting from evaluating the kernel function between all inducing inputs, K_{nm} is the cross-kernel matrix between inducing inputs and training points and $\widetilde{K} = K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}$. Including the inducing points in our model gives the augmented joint distribution

$$p(\boldsymbol{y}, \boldsymbol{\omega}, \boldsymbol{f}, \boldsymbol{u}) = p(\boldsymbol{y}|\boldsymbol{\omega}, \boldsymbol{f})p(\boldsymbol{\omega})p(\boldsymbol{f}|\boldsymbol{u})p(\boldsymbol{u}) \tag{7}$$

Note that the original model (2) can be recovered by marginalizing $\boldsymbol{\omega}$ and $\boldsymbol{u}$.

4. Inference

The goal of Bayesian inference is to compute the posterior of the latent model variables. Because this problem is intractable for the model at hand, we employ variational inference to map the inference problem to a feasible optimization problem. We first chose a family of tractable variational distributions and select the best candidate by minimizing the Kullback-Leibler divergence between the posterior and the variational distribution. This is equivalent to optimizing a lower bound on the marginal likelihood, known as evidence lower bound (ELBO) (Jordan et al., 1999; Wainwright and Jordan, 2008).

In the following we develop a stochastic variational inference (SVI) algorithm that enables stochastic optimization of the variational bound using only mini-batches of the data in each iteration.

4.1. Variational approximation

We aim to approximate the posterior of the inducing points $p(\boldsymbol{u}|\boldsymbol{y})$ and apply the methodology of variational inference to the marginal joint distribution $p(y, \omega, u) = p(\boldsymbol{y}|\boldsymbol{\omega}, \boldsymbol{u})p(\boldsymbol{\omega})p(\boldsymbol{u})$. Following a similar approach as Hensman and Matthews (2015), we apply Jensen's inequality to obtain a tractable lower bound on the log-likelihood of the labels

$$\begin{aligned}\log p(\boldsymbol{y}|\boldsymbol{\omega}, \boldsymbol{u}) &= \log \mathbb{E}_{p(f|u)}[p(\boldsymbol{y}|\boldsymbol{\omega}, f)] \\ &\geq \mathbb{E}_{p(f|u)}[\log p(\boldsymbol{y}|\boldsymbol{\omega}, f)]. \end{aligned} \tag{8}$$

By this inequality we construct a variational lower bound on the evidence

$$\begin{aligned}\log p(\boldsymbol{y}) &\geq \mathbb{E}_{q(\boldsymbol{u},\boldsymbol{\omega})}[\log p(\boldsymbol{y}|\boldsymbol{u},\boldsymbol{\omega})] \\ &\quad - \mathrm{KL}\left(q(\boldsymbol{u},\boldsymbol{\omega})||p(\boldsymbol{u},\boldsymbol{\omega})\right) \\ &\geq \mathbb{E}_{p(\boldsymbol{f}|\boldsymbol{u})q(\boldsymbol{u})q(\boldsymbol{\omega})}[\log p(\boldsymbol{y}|\boldsymbol{\omega},\boldsymbol{f})] \\ &\quad - \mathrm{KL}\left(q(\boldsymbol{u},\boldsymbol{\omega})||p(\boldsymbol{u},\boldsymbol{\omega})\right) \\ &=: \mathcal{L},\end{aligned}$$

where the first inequality is the usual evidence lower bound (ELBO) in variational inference and the second inequality is due to (8).

We follow a structured mean-field approach (Wainwright and Jordan, 2008) and assume independence between the inducing variables u and Pólya-Gamma variables ω, yielding a variational distribution of the form $q(u,\omega) = q(u)q(\omega)$. Setting the functional derivative of $\mathcal{L}$ w.r.t. $q(u)$ and $q(\omega)$ to zero, respectively, results in the following consistency condition for the maximum,

$$q(\boldsymbol{u},\boldsymbol{\omega}) = q(\boldsymbol{u})\prod_i q(\omega_i), \tag{9}$$

with $q(\omega_i) = \mathrm{PG}(\omega_i|1,c_i)$ and $q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}|\boldsymbol{\mu},\Sigma)$. Remarkably, we do not have to use the full Pólya-Gamma class $\mathrm{PG}(\omega_i|b_i,c_i)$, but instead consider the restricted class $b_i = 1$ since it already contains the optimal distribution.

We use (9) as variational family which is parameterized by the variational parameters $\{\boldsymbol{\mu},\Sigma,\boldsymbol{c}\}$ and obtain a closed-form expression of the variational bound

$$\begin{aligned}&\mathcal{L}(\boldsymbol{c},\boldsymbol{\mu},\Sigma) \\ &= \mathbb{E}_{p(\boldsymbol{f}|\boldsymbol{u})q(\boldsymbol{u})q(\boldsymbol{\omega})}[\log p(\boldsymbol{y}|\boldsymbol{\omega},\boldsymbol{f})] \\ &\quad - \mathrm{KL}\left(q(\boldsymbol{u},\boldsymbol{\omega})||p(\boldsymbol{u},\boldsymbol{\omega})\right) \\ &\overset{\mathsf{c}}{=} \frac{1}{2}\Bigg(\log|\Sigma| - \log|K_{mm}|) - \mathrm{tr}(K_{mm}^{-1}\Sigma) - \boldsymbol{\mu}^\top K_{mm}^{-1}\boldsymbol{\mu} \\ &\quad + \sum_i \Big\{ y_i\boldsymbol{\kappa}_i\boldsymbol{\mu} - \theta_i\left(\widetilde{K}_{ii} - \boldsymbol{\kappa}_i\Sigma\boldsymbol{\kappa}_i^\top - \boldsymbol{\mu}^\top\boldsymbol{\kappa}_i^\top\boldsymbol{\kappa}_i\boldsymbol{\mu}\right) \\ &\quad + c_i^2\theta_i - 2\log\cosh\frac{c_i}{2}\Big\}\Bigg),\end{aligned} \tag{10}$$

where $\theta_i = \frac{1}{2c_i}\tanh\left(\frac{c_i}{2}\right)$ and $\boldsymbol{\kappa}_i = K_{im}K_{mm}^{-1}$. Remarkably, all intractable terms involving expectations of $\log\mathrm{PG}(\omega_i|1,0)$ cancel out. Details are provided in appendix A.2.

4.2. Stochastic variational inference

We employ stochastic variational inference (SVI) (Hoffman et al., 2013) to optimize the variational bound (10) using stochastic optimization.

Since we have the variational objective in closed-form we are able to compute the natural gradients in closed-form as well. Using the natural gradient over the standard Euclidean gradient is favorable since natural gradients are invariant to reparameterization of the variational family (Amari and Nagaoka, 2007; Martens, 2017) and provide effective second-order optimization updates (Amari, 1998; Hoffman et al., 2013).

This is in contrast to the model of Hensman and Matthews (2015), where the global updates cannot be computed in a closed-form and one relies on less efficient Euclidean gradient updates that are computed using numerical quadrature methods.

Closed-form updates Our algorithm alternates between updates of the local variational parameters $\boldsymbol{c}$ and global parameters $\boldsymbol{\mu}$ and Σ. In each iteration we update the parameters based on a mini-batch of the data $\mathcal{S} \subset \{1,...,n\}$ of size $s = |\mathcal{S}|$.

We update the *local parameters* $\boldsymbol{c}_\mathcal{S}$ in the mini-batch $\mathcal{S}$ by employing coordinate ascent. To this end, we fix the global parameters and analytically compute the unique maximum of (10) w.r.t. the local parameters, leading to the updates

$$c_i = \sqrt{\widetilde{K}_{ii} + \boldsymbol{\kappa}_i\Sigma\boldsymbol{\kappa}_i^\top + \boldsymbol{\mu}^\top\boldsymbol{\kappa}_i^\top\boldsymbol{\kappa}_i\boldsymbol{\mu}} \tag{11}$$

for $i \in \mathcal{S}$.

We update the *global parameters* by employing stochastic optimization of the variational bound (10). The optimization is based on stochastic estimates of the natural gradients of the global parameters. This is the standard Euclidean gradient multiplied by the inverse Fisher information matrix. We use the natural parameterization of the variational Gaussian distribution, i.e., the parameters $\boldsymbol{\eta}_1 := \Sigma^{-1}\boldsymbol{\mu}$ and $\eta_2 = -\frac{1}{2}\Sigma^{-1}$. Using the natural parameters results in simpler and more effective updates. The natural gradients based on the mini-batch $\mathcal{S}$ are given by

$$\begin{aligned}\widetilde{\nabla}_{\boldsymbol{\eta}_1}\mathcal{L}_\mathcal{S} &= \frac{n}{2s}\boldsymbol{\kappa}_\mathcal{S}^\top\boldsymbol{y}_\mathcal{S} - \boldsymbol{\eta}_1 \\ \widetilde{\nabla}_{\eta_2}\mathcal{L}_\mathcal{S} &= -\frac{1}{2}\left(K_{mm}^{-1} + \frac{n}{s}\boldsymbol{\kappa}_\mathcal{S}^\top\Theta_\mathcal{S}\boldsymbol{\kappa}_\mathcal{S}\right) - \eta_2,\end{aligned} \tag{12}$$

where $\Theta = \mathrm{diag}(\boldsymbol{\theta})$ and $\theta_i = \frac{1}{2c_i}\tanh\left(\frac{c_i}{2}\right)$. The factor $\frac{n}{s}$ is due to the rescaling of the mini-batches. The global parameters are updated according to a stochastic natural gradient ascent scheme. We employ the adaptive learning rate method described by Ranganath et al. (2013).

The natural gradient updates always lead to a positive definite covariance matrix[2] and in contrast to Hensman and Matthews (2015) our implementation does not require any assurance for positive-definiteness of the variational covariance matrix Σ.

[2]This follows directly since K_{mm} and Θ are positive definite.

Details for the derivation of the updates can be found in appendix A.3. The complexity of each iteration in the inference scheme is $\mathcal{O}(m^3)$, due to the inversion of the matrix η_2.

On the quality of the approximation In other applications of variational inference to GP classification, one tries to approximate the posterior directly by a Gaussian $q^*(f)$ which minimizes the Kullback-Leibler divergence between the variational distribution and the true posterior (Hensman and Matthews, 2015). On the other hand, in our paper, we apply variational inference to the augmented model, looking for the best distribution that factorizes in the Pólya-Gamma variables ω_i and the original function f. This approach also yields a Gaussian approximation $q(f)$ as a factor in the optimal density. Of course $q(f)$ will be different from the optimal $q^*(f)$. We could however argue that asymptotically, in the limit of a large number of data, the predictions given by both densities may not be too different, as the posterior uncertainty for both densities should become small (Opper and Archambeau, 2009).

It would be interesting to see how the ELBOs of the two variational approaches, which both give a lower bound on the likelihood of the data, differ. Unfortunately, such a computation would require the knowledge of the optimal $q^*(f)$. However, we can obtain some estimate of this difference when we assume that we use the *same* Gaussian density $q(f)$ for both bounds as an approximation. In this case, we obtain

$$\mathcal{L}_{\text{orig}} - \mathcal{L}_{\text{augmented}} = \mathbb{E}_{q(f)}[\text{KL}\left(q(\omega)||p(\omega|f,y)\right)].$$

This lower bound on the gap is small if on average the variational approximation $q(\omega)$ is close to the posterior $p(\omega|f,y)$. For the sake of simplicity we consider here the non-sparse case, i.e. the inducing points equal the training points ($f = u$). However, it is straight-forward to extend the results also to the sparse case.

We also investigate the question if we can improve our approximation after convergence, such that it it will be closer to the true posterior. We propose a correction method based on perturbation theory (Opper et al., 2015). The idea is to view the variational bound as a first-order term from a perturbation of the true log evidence and to derive a power series of corrections. We derive a second-order correction which is a Gaussian distribution with corrected mean and covariance and is presented in appendix A.6. The quality of the original approximation and the corrected version is investigated in experiment 5.1.

Predictions The approximate posterior of the GP values and inducing variables is given by $q(\boldsymbol{f}, \boldsymbol{u}) = p(\boldsymbol{f}|\boldsymbol{u})q(\boldsymbol{u})$, where $q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}|\boldsymbol{\mu}, \Sigma)$ denotes the optimal variational distribution. To predict the latent function values f_* at a test point x_* we substitute our approximate posterior into the standard predictive distribution

$$\begin{aligned} p(f_*|y) &= \int p(f_*|\boldsymbol{f},\boldsymbol{u})p(\boldsymbol{f},\boldsymbol{u}|y)\mathrm{d}\boldsymbol{f}\mathrm{d}\boldsymbol{u} \\ &\approx \int p(f_*|\boldsymbol{f},\boldsymbol{u})p(\boldsymbol{f}|\boldsymbol{u})q(\boldsymbol{u})\mathrm{d}\boldsymbol{f}\mathrm{d}\boldsymbol{u} \\ &= \int p(f_*|\boldsymbol{u})q(\boldsymbol{u})\mathrm{d}\boldsymbol{u} \\ &= \mathcal{N}\left(f^*|\mu_*,\ \sigma_*^2\right), \end{aligned} \tag{13}$$

where the prediction mean is $\mu_* = K_{*m}K_{mm}^{-1}\boldsymbol{\mu}$ and the variance $\sigma_*^2 = K_{**} + K_{*m}K_{mm}^{-1}(\Sigma K_{mm}^{-1} - I)K_{m*}$. The matrix K_{*m} denotes the kernel matrix between the test point and the inducing points and K_{**} the kernel value of the test point. The distribution of the test labels is easily computed by applying the logit link function to (13),

$$p(y_* = 1|y) = \int \sigma(f_*)p(f_*|y)\mathrm{d}f_*. \tag{14}$$

This integral is analytically intractable but can be computed numerically by quadrature methods. This is adequate and fast since the integral is only one-dimensional.

Computing the mean and the variance of the predictive distribution has complexity $\mathcal{O}(m)$ and $\mathcal{O}(m^2)$, respectively.

Optimization of the hyperparameters and inducing point locations We select the optimal kernel hyperparameters by maximizing the marginal likelihood $p(y|h)$, where h denotes the set of hyperparameters and the inducing point locations (this approach is called empirical Bayes (Maritz and Lwin, 1989)). We follow an approximate approach and optimize the fitted variational lower bound $\mathcal{L}(h)$ (10) as a function of h by alternating between optimization steps w.r.t. the variational parameters and the hyperparameters (Mandt et al., 2016).

5. Experiments

We compare our proposed method, efficient Gaussian process classification (**X-GPC**), with the state-of-the-art method **SVGPC** by Hensman and Matthews (2015), provided in the highly optimized package GPflow[3] (Matthews et al., 2017), which builds on TensorFlow (Abadi et al., 2015). Both methods are applied to real-world datasets containing up to 11 million datapoints.

We do not include the method proposed in Hernández-Lobato and Hernández-Lobato (2016) in our experiments since no public code was available. The authors note that "the performance of the stochastic variants of [their approach] and [SVGPC] in terms of the test error or the aver-

[3]We use GPflow version 0.4.0.

age negative log likelihood with respect to the running time is very similar".

In all experiments a squared exponential covariance function with a common length scale parameter for each dimension, an amplitude parameter and an additive noise parameter is used. The kernel hyperparameters and inducing point locations are initialized to the same values and optimized using Adam (Kingma and Ba, 2014). All algorithms are run on a single CPU. We experiment on 11 datasets from the OpenML website and the UCI repository ranging from 768 to 11 million datapoints.

In the first experiment (section 5.1), we examine the quality of the approximation provided by our method X-GPC and the proposed perturbative correction. In section 5.2, we evaluate the prediction performance and run time of X-GPC and SVGPC on several real-world datasets. We investigate the relation of prediction performance vs. the training time in section 5.3. In section 5.4, we examine the sensitivity of the models to the number of inducing points.

5.1. Quality of the approximation

We empirically examine the quality of the variational approximation provided by our method and its perturbative correction (see section 4.2). We compare the approximations to the true posterior obtained by employing an asymptotically correct Gibbs sampler (Polson and Scott, 2011; Linderman et al., 2015). For both approximations, we compare the posterior mean and variance as well as the prediction probabilities with the ground truth. In figure 1 we plot the approximated values vs. the ground truth. In table 1 we report the mean absolute difference between the approximations and the ground truth as well as the difference of the prediction performance using the approximate predictive distribution and true predictive distribution.

Since the Gibbs sampler does not scale to large datasets we experiment on the two small datasets: German and Diabetis. We find that our original solution is very close to the true posterior and can be marginally improved by the perturbative correction. While the correction may be beneficial for the datasets considered in section 5.2, we found the difference to be insignificant. We therefore only include the original (non-corrected) method in the following experiments.

5.2. Numerical comparison

We evaluate the prediction performance and run time of our method X-GPC and the state-of-the-art SVGPC. We perform a 10-fold cross-validation and report average prediction errors with one standard deviation. For datasets with more than 1 million points, we limit the test set to 100,000 points. We report the average prediction error, the negative test log-likelihood (14) and the run time along with one standard

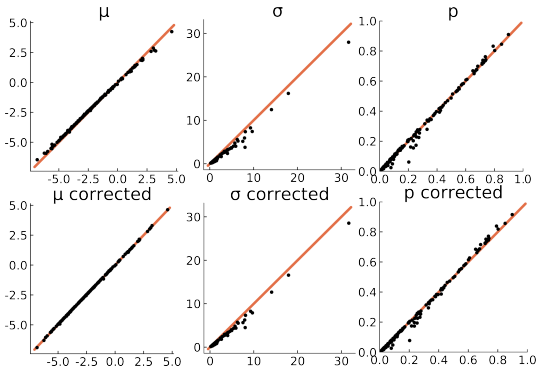

Figure 1. Test marginals of the Diabetis dataset. Each plot shows the MCMC ground truth on the x-axis and the estimated value of our model with and without the perturbative correction, respectively, on the y-axis.

Diabetis	Δ mean	Δ variance	pred. error	NLL
X-GPC	0.103	0.426	0.156	0.347
Corrected	0.033	0.348	0.156	0.347
Ground Truth	0	0	0.156	0.346
German				
X-GPC	0.237	2.28	0.38	0.666
Corrected	0.104	1.94	0.38	0.667
Ground Truth	0	0	0.38	0.662

Table 1. We compare the original X-GPC solution and the proposed perturbative correction with the ground truth on the datasets Diabetis and German. We report the mean absolute difference between the estimated values and the ground truth (Δ mean and Δ variance), along with the prediction error and the average negative log predictive likelihood. Lower values indicate better performance.

deviation. For all datasets, we use 100 inducing points. We use a mini-batch size of 100 points, except for the two small datasets Diabetes and German, where we use the full-batch versions of both methods.

For X-GPC we found that the following simple convergence criterion on the global parameters leads to good results: a sliding window average being smaller than a threshold. Unfortunately, the implementation of SVGPC in GPflow does not include a convergence criterion. We found that the trajectories of the global parameters of SVGPC tend to be noisy, and using a convergence criterion on the global parameters often leads to poor results. To have a fair comparison, we therefore monitor the convergence of the prediction performance on a hold-out set and use a sliding window average of size 5 and threshold 10^{-4} as convergence criterion.

The results are shown in table 2. We observe that X-GPC is about one to three orders of magnitude faster than SVPGC on all datasets: we obtain speed ups ranging from a factor of 10.8 on Shuttle to a factor of 506 on Diabetes. The prediction error is similar for both methods but we observe that

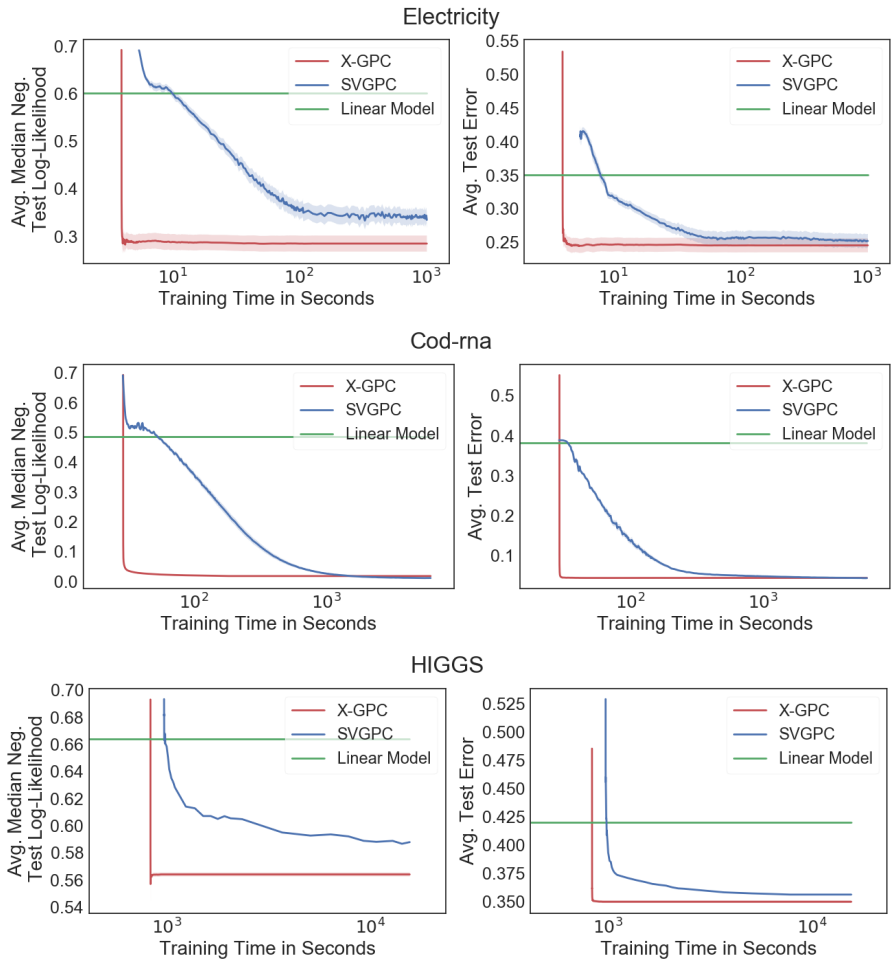

Figure 2. Average median of the negative test log-likelihood and average test prediction error as a function of training time (seconds in a $\log_{10}$ scale) on the datasets Electricity (45,312 points), Cod RNA (343,564 points) and Higgs (11 million points).

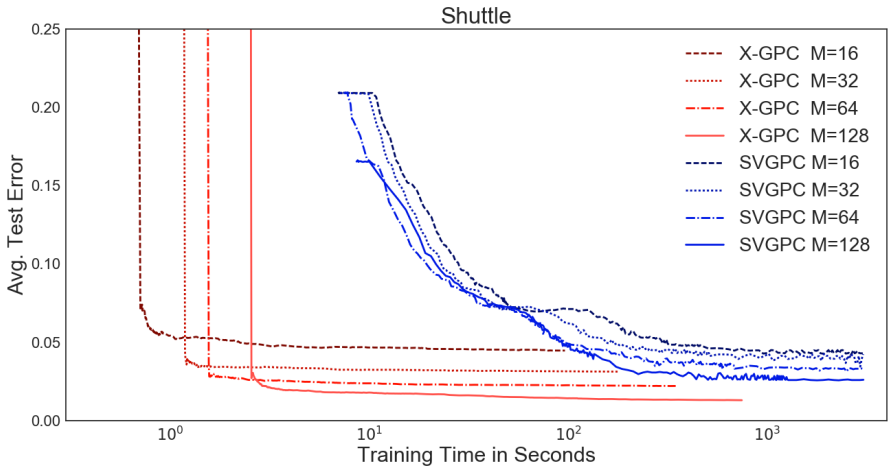

Figure 3. Prediction error as function of training time (on a $\log_{10}$ scale). Different numbers of inducing points are considered, $m = 16, 32, 64, 128$. Best seen in color.

Dataset	n / d		**X-GPC**	SVGPC
aXa	36974 123	Error NLL Time	0.17 ± 0.07 0.16 ± 0.10 8.7 ± 0.9	0.17 ± 0.07 0.18 ± 0.12 571 ± 2.2
Bank Market.	45211 43	Error NLL Time	0.11 ± 0.09 0.10 ± 0.10 7.4 ± 1.8	0.11 ± 0.09 0.10 ± 0.09 609 ± 2.7
Click Predict.	399482 12	Error NLL Time	0.17 ± 0.00 0.17 ± 0.01 35 ± 2.7	0.17 ± 0.00 0.24 ± 0.01 1256 ± 191
Cod RNA	343564 8	Error NLL Time	0.04 ± 0.00 0.02 ± 0.00 134 ± 15	0.05 ± 0.00 0.01 ± 0.00 3002 ± 122
Cov Type	581012 54	Error NLL Time	0.32 ± 0.06 0.53 ± 0.06 29 ± 9.66	0.32 ± 0.05 0.54 ± 0.07 1004 ± 51
Diabetis	768 8	Error NLL Time	0.23 ± 0.07 0.31 ± 0.12 0.8 ± 0.1	0.23 ± 0.07 0.33 ± 0.11 405 ± 59
Electricity	45312 8	Error NLL Time	0.25 ± 0.07 0.29 ± 0.05 4.6 ± 1.2	0.25 ± 0.06 0.34 ± 0.05 888 ± 1.9
German	1000 20	Error NLL Time	0.25 ± 0.13 0.40 ± 0.19 1.03 ± 0.2	0.25 ± 0.13 0.39 ± 0.18 319 ± 15
Higgs	11M 22	Error NLL Time	0.36 ± 0.00 0.05 ± 0.01 14.1 ± 5.7	0.35 ± 0.00 0.03 ± 0.02 1019 ± 5.9
Shuttle	58000 9	Error NLL Time	0.02 ± 0.00 0.01 ± 0.00 139 ± 6.6	0.03 ± 0.00 0.00 ± 0.00 1501 ± 92
SUSY	5M 18	Error NLL Time	0.20 ± 0.01 0.21 ± 0.01 523 ± 25	0.21 ± 0.01 0.27 ± 0.01 10366 ± 360

Table 2. Average test prediction error, negative test log-likelihood (NLL) and time in seconds along with one standard deviation.

X-GPC improves the test log-likelihood on some datasets (Click Prediction, Electricity, Susy). This means that the confidence levels in the predictions are better calibrated for X-GPC, i.e., when predicting a wrong label, SVGPC tends to be more confident than X-GPC.

5.3. Performance as a function of time

Since X-GPC and SVGPC are based on an optimization scheme, there is a trade-off between the run time of the algorithm and the prediction performance. We profile each method and monitor the negative test log-likelihood and prediction error on a hold-out test set as a function of time. We do not plot the variational bounds of X-GPC and SVGPC since they are not directly comparable due to the difference of the likelihood functions (logit vs. probit link). Again we employ a 10-fold cross validation and plot the median negative test log-likelihood and the average prediction error values along with the standard errors. We do not plot the variational bound since it differs for X-GPC and SVPGC and thus is not directly comparable. We adopt the same kernel setting as in section 5.2 and use 100 inducing points. As a benchmark, we fit a linear model and use the logistic regression implementation in scikit-learn (Pedregosa et al., 2011), which is based on LIBLINEAR (Fan et al., 2008).

The results are displayed in figure 2 for selected datasets, while the results for the remaining datasets are deferred to appendix A.1. X-GPC is already close to the optimum after a few iterations due to its efficient natural gradient updates. The test log-likelihood and the prediction error converges around one to three orders of magnitude faster for X-GPC than for SVGPC.

Being based on Tensorflow, SVGPC has a slightly longer initialization time than X-GPC. However, the main speed-up is due to the more efficient updates performed by X-GPC.

5.4. Inducing points

We examine the effect of different numbers of inducing points on the prediction performance and run time. For both methods we compare different numbers of inducing points: $M = 16, 32, 64, 128$. For each setting, we perform a 10-fold cross validation on the Shuttle dataset and plot the mean prediction error as function of time.

The results are displayed in figure 3. We observe for both methods that the higher the number of inducing points, the better the prediction performance, but the longer the run time. Throughout all settings of inducing points our method is consistently faster of around one to three orders of magnitude. The final prediction error is similar for both methods for $M = 16$, whereas X-GPC significantly improves performance for $M = 32, 64, 128$.

6. Conclusions

We proposed an efficient Gaussian process classification method that builds on Pólya-Gamma data augmentation and inducing points. The experimental evaluations shows that our method is up to three orders of magnitude faster than the state-of-the-art approach while being competitive in terms of prediction performance. Speed improvements are due to the Pólya-Gamma data augmentation approach that enables efficient second order optimization.

The presented work shows how data augmentation can speed up variational approximation of GPs. Our analysis may pave the way for using data augmentation to derive efficient stochastic variational algorithms also for variational Bayesian models other than GPs. Furthermore, future work may aim at extending the approach to multi-class and multi-label classification.

ACKNOWLEDGEMENTS

We thank Stephan Mandt, James Hensman and Scott W. Linderman for fruitful discussions. This work was partly funded by the German Research Foundation (DFG) awards KL 2698/2-1 and GRK1589/2 and the by the Federal Ministry of Science and Education (BMBF) awards 031L0023A and 031B0187B.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Amari, S. (1998). Natural grad. works efficiently in learning. *Neural Computation*.

Amari, S. and Nagaoka, H. (2007). *Methods of Information Geometry*. American Mathematical Society.

Chen, H., Chiang, R. H., and Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS quarterly*, 36(4).

Collins, F. S. and Varmus, H. (2015). A new initiative on precision medicine. *New England Journal of Medicine*, 372(9):793–795.

Dezfouli, A. and Bonilla, E. V. (2015). Scalable inference for gaussian process models with black-box likelihoods. In *Advances in Neural Information Processing Systems 28*, pages 1414–1422.

Dragiev, S., Toussaint, M., and Gienger, M. (2011). Gaussian process implicit surfaces for shape estimation and grasping. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2845–2850.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Gibbs, M. N. and MacKay, D. J. C. (2000). Variational Gaussian process classifiers. *IEEE Transactions on Neural Networks*, 11(6):1458–1464.

Hensman, J. and Matthews, A. (2015). Scalable Variational Gaussian Process Classification. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics,*.

Hernández-Lobato, D. and Hernández-Lobato, J. M. (2016). Scalable gaussian process classification via expectation propagation. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 168–176.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic Variational Inference. *Journal of Machine Learning Research*.

Honkela, A., Raiko, T., Kuusela, M., Tornio, M., and Karhunen, J. (2010). Approximate riemannian conjugate gradient learning for fixed-form variational bayes. *Journal of Machine Learning Research*, 11:3235–3268.

Jähnichen, P., Wenzel, F., Kloft, M., and Mandt, S. (2018). Scalable generalized dynamic topic models. In *International Conference on Artificial Intelligence and Statistics*.

John Walker, S. (2014). *Big data: A revolution that will transform how we live, work, and think*. Taylor & Francis.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An Introduction to Variational Methods for Graphical Models. *Machine Learning*.

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

Kitchin, R. (2014). The real-time city? big data and smart urbanism. *GeoJournal*, 79(1):1–14.

Linderman, S. W., Johnson, M. J., and Adams, R. P. (2015). Dependent multinomial models made easy: Stick-breaking with the polya-gamma augmentation. In *Advances in Neural Information Processing Systems 28*, pages 3456–3464.

Mandt, S., Hoffman, M., and Blei, D. (2016). A Variational Analysis of Stochastic Gradient Algorithms. *International Conference on Machine Learning*.

Maritz, J. and Lwin, T. (1989). Empirical Bayes Methods with Applications. *Monographs on Statistics and Applied Probability.*

Martens, J. (2017). New insights and perspectives on the natural gradient method. *Arxiv Preprint*.

Matthews, A. G. d. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017). GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6.

Naish-Guzman, A. and Holden, S. B. (2007). The generalized FITC approximation. In *Advances in Neural Information Processing Systems 20*, pages 1057–1064.

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

Opper, M., Fraccaro, M., Paquet, U., Susemihl, A., and Winther, O. (2015). Perturbation theory for variational inference. *NIPS WS*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Polson, N. G., Scott, J. G., and Windle, J. (2013). Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349.

Polson, N. G. and Scott, S. L. (2011). Data augmentation for support vector machines. *Bayesian Anal.*

Ranganath, R., Wang, C., Blei, D. M., and Xing, E. P. (2013). An Adaptive Learning Rate for Stochastic Variational Inference. *International Conference on Machine Learning*.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Scott, J. G. and Sun, L. (2013). Expectation-maximization for logistic regression. *arXiv preprint arXiv:1306.0040*.

Snelson, E. and Ghahramani, Z. (2006). Sparse GPs using Pseudo-inputs. *NIPS*.

Stein, M. L. (2012). *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, pages 1–305.

Wenzel, F., Galy-Fajou, T., Deutsch, M., and Kloft, M. (2017). Bayesian nonlinear support vector machines for big data. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*.

A. Appendix

A.1. Additional performance plots

We show all time vs. prediction performance plots for the datasets presented in table 2 in section section 5.2 which could not be included in the main paper due to space limitations.

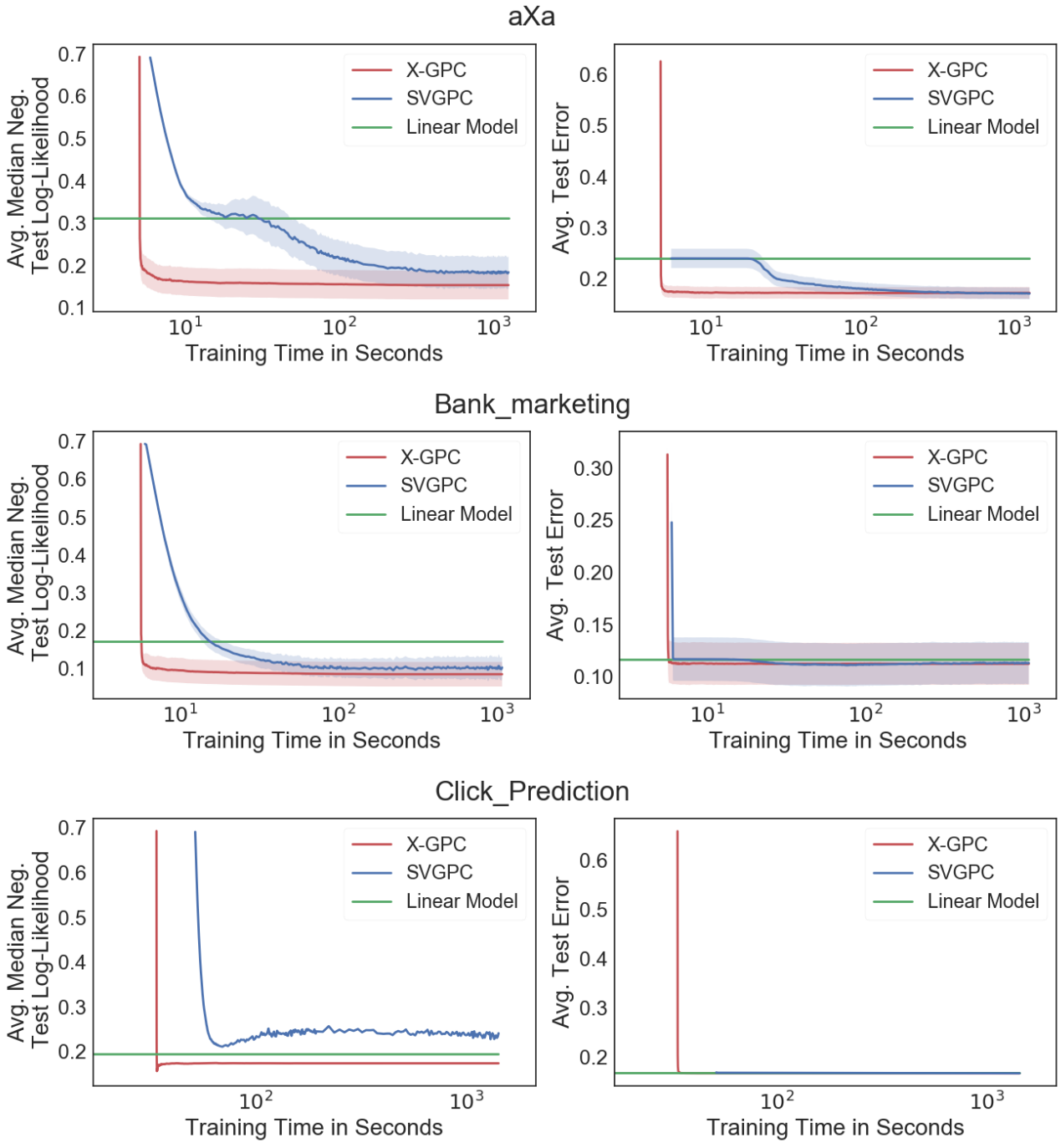

Figure 4. Average median of the negative test log-likelihood and average test prediction error as function of training time measured in seconds (on a $\log_{10}$ scale).

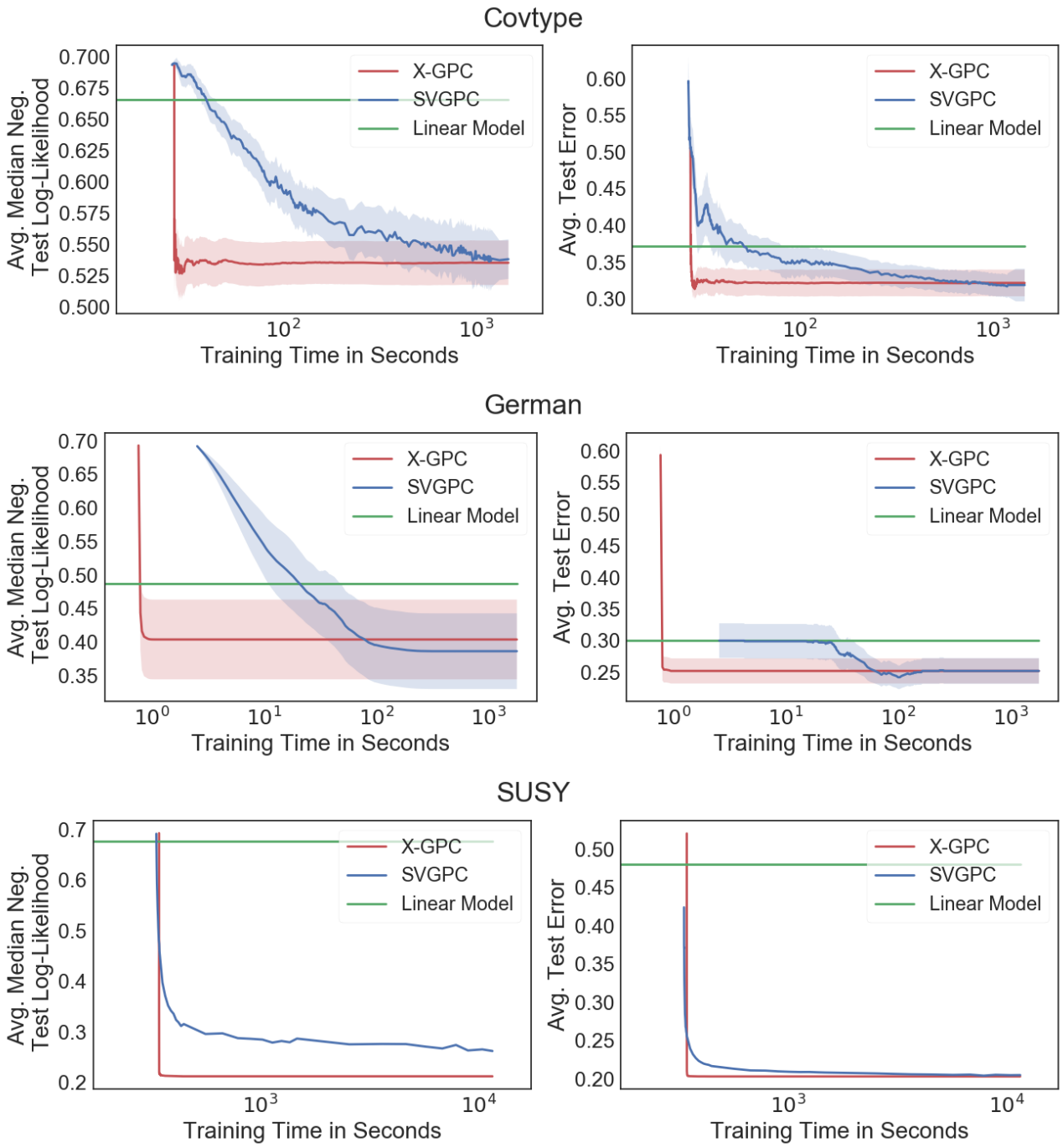

Figure 5. Average median of the negative test log-likelihood and average test prediction error as function of training time measured in seconds (on a $\log_{10}$ scale).

A.2. Variational bound

We provide details of the derivation of the variational bound (10) which is defined as

$$\mathcal{L}(\boldsymbol{c}, \boldsymbol{\mu}, \Sigma) = \mathbb{E}_{p(\boldsymbol{f}|\boldsymbol{u})q(\boldsymbol{u})q(\boldsymbol{\omega})}[\log p(\boldsymbol{y}|\boldsymbol{\omega}, \boldsymbol{f})] - \mathrm{KL}\left(q(\boldsymbol{u}, \boldsymbol{\omega})||p(\boldsymbol{u}, \boldsymbol{\omega})\right),$$

and the family of variational distributions

$$q(\boldsymbol{u}, \boldsymbol{\omega}) = q(\boldsymbol{u}) \prod_i q(\omega_i) = \mathcal{N}(\boldsymbol{u}|\boldsymbol{\mu}, \Sigma) \prod_i \mathrm{PG}(\omega_i|1, c_i).$$

Considering the likelihood term we obtain

$$\begin{aligned}\mathbb{E}_{p(f|u)}[\log p(\boldsymbol{y}|\boldsymbol{\omega}, f)] &\stackrel{\mathsf{c}}{=} \frac{1}{2}\mathbb{E}_{p(f|u)}\left[\boldsymbol{y}^\top \boldsymbol{f} - \boldsymbol{f}^\top \Omega \boldsymbol{f}\right] \\ &= \frac{1}{2}\left(\boldsymbol{y}^\top K_{nm}K_{mm}^{-1}\boldsymbol{u} - \mathrm{tr}(\Omega\widetilde{K}) - \boldsymbol{u}^\top K_{mm}^{-1}K_{mn}\Omega K_{nm}K_{mm}^{-1}\boldsymbol{u}\right).\end{aligned}$$

Computing the expectations w.r.t. to variational distributions gives

$$\begin{aligned}&\mathbb{E}_{p(\boldsymbol{f}|\boldsymbol{u})q(\boldsymbol{u})q(\boldsymbol{\omega})}[\log p(\boldsymbol{y}|\boldsymbol{\omega}, \boldsymbol{f})] \\ &= \frac{1}{2}\mathbb{E}_{q(\boldsymbol{u})q(\boldsymbol{\omega})}\left[\boldsymbol{y}^\top K_{nm}K_{mm}^{-1}\boldsymbol{u} - \mathrm{tr}(\Omega\widetilde{K}) - \boldsymbol{u}^\top K_{mm}^{-1}K_{mn}\Omega K_{nm}K_{mm}^{-1}\boldsymbol{u}\right] \\ &= \frac{1}{2}\mathbb{E}_{q(\boldsymbol{u})}\left[\boldsymbol{y}^\top K_{nm}K_{mm}^{-1}\boldsymbol{u} - \mathrm{tr}(\Theta\widetilde{K}) - \boldsymbol{u}^\top K_{mm}^{-1}K_{mn}\Theta K_{nm}K_{mm}^{-1}\boldsymbol{u}\right] \\ &= \frac{1}{2}\left[\boldsymbol{y}^\top K_{nm}K_{mm}^{-1}\boldsymbol{\mu} - \mathrm{tr}(\Theta\widetilde{K}) - \mathrm{tr}(K_{mm}^{-1}K_{mn}\Theta K_{nm}K_{mm}^{-1}\Sigma) - \boldsymbol{\mu}^\top K_{mm}^{-1}K_{mn}\Theta K_{nm}K_{mm}^{-1}\boldsymbol{\mu}\right] \\ &= \frac{1}{2}\sum_i \left(y_i\boldsymbol{\kappa}_i\boldsymbol{\mu} - \theta_i\widetilde{K}_{ii} - \theta_i\boldsymbol{\kappa}_i\Sigma\boldsymbol{\kappa}_i^\top - \theta_i\boldsymbol{\mu}^\top\boldsymbol{\kappa}_i^\top\boldsymbol{\kappa}_i\boldsymbol{\mu}\right),\end{aligned}$$

where $\theta_i = \mathbb{E}_{p(\omega_i)}[\omega_i] = \frac{1}{2c_i}\tanh\left(\frac{c_i}{2}\right)$, $\Theta = \mathrm{diag}(\boldsymbol{\theta})$ and $\boldsymbol{\kappa}_i = K_{im}K_{mm}^{-1}$.

The Kullback-Leibler divergence between the Gaussian distributions $q(\boldsymbol{u})$ and $p(\boldsymbol{u})$ is easily computed

$$\mathrm{KL}(q(\boldsymbol{u}))||p(\boldsymbol{u})) \stackrel{\mathsf{c}}{=} \frac{1}{2}\left(\mathrm{tr}\left(K_{mm}^{-1}\Sigma\right) + \boldsymbol{\mu}^\top K_{mm}^{-1}\boldsymbol{\mu} - \log|\Sigma| + \log|K_{mm}|\right).$$

The Kullback-Leibler divergence regarding the Pólya-Gamma also can be computed in closed-form. Have $q(\omega_i) = \cosh\left(\frac{c_i}{2}\right)\exp\left(-\frac{c_i^2}{2}\omega_i\right)\mathrm{PG}(\omega_i|1,0)$ and $p(\omega_i) = \mathrm{PG}(\omega_i|1,0)$ we obtain

$$\begin{aligned}\mathrm{KL}(q(\boldsymbol{\omega}))||p(\boldsymbol{\omega})) &= \mathbb{E}_{q(\boldsymbol{\omega})}\left[\log q(\boldsymbol{\omega}) - \log p(\boldsymbol{\omega})\right] \\ &= \sum_i \left(\mathbb{E}_{q(\omega_i)}\left[\log\left(\cosh\left(\frac{c_i}{2}\right)\exp\left(-\frac{c_i^2}{2}\omega_i\right)\mathrm{PG}(\omega_i|1,0)\right)\right] - \mathbb{E}_{q(\omega_i)}\left[\log\mathrm{PG}(\omega_i|1,0)\right]\right) \\ &= \sum_i \left(\log\cosh\frac{c_i}{2} - \frac{c_i}{4}\tanh\left(\frac{c_i}{2}\right) + \mathbb{E}_{q(\omega_i)}\left[\log\mathrm{PG}(\omega_i|1,0)\right] - \mathbb{E}_{q(\omega_i)}\left[\log\mathrm{PG}(\omega_i|1,0)\right]\right) \\ &= \sum_i \left(\log\cosh\frac{c_i}{2} - \frac{c_i}{4}\tanh\left(\frac{c_i}{2}\right)\right).\end{aligned}$$

Remarkably, the intractable expectations cancel out which would not have been the case if we assumed $\mathrm{PG}(\omega_i|b_i, c_i)$ as variational family. In section 4.1 we have shown that the restricted family $b_i = 1$ contains the optimal distribution.

Summing all terms results in the final lower bound

$$\begin{aligned}\mathcal{L}(\boldsymbol{c}, \boldsymbol{\mu}, \Sigma) \stackrel{\mathsf{c}}{=} \frac{1}{2}\Bigg(&\log|\Sigma| - \log|K_{mm}|) - \mathrm{tr}(K_{mm}^{-1}\Sigma) - \boldsymbol{\mu}^\top K_{mm}^{-1}\boldsymbol{\mu} + \\ &\sum_i \left\{y_i\boldsymbol{\kappa}_i\boldsymbol{\mu} - \theta_i\widetilde{K}_{ii} - \theta_i\boldsymbol{\kappa}_i\Sigma\boldsymbol{\kappa}_i^\top - \theta_i\boldsymbol{\mu}^\top\boldsymbol{\kappa}_i^\top\boldsymbol{\kappa}_i\boldsymbol{\mu} + c_i^2\theta_i - 2\log\cosh\frac{c_i}{2}\right\}\Bigg).\end{aligned}$$

A.3. Variational updates

Local parameters The derivative of the variational bound (10) w.r.t. the local parameter c_i is

$$\begin{aligned}
\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}c_i} &= \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}c_i}\left\{\theta_i\left(-\widetilde{K}_{ii} - \boldsymbol{\kappa}_i\Sigma\boldsymbol{\kappa}_i^\top - \boldsymbol{\mu}^\top\boldsymbol{\kappa}_i^\top\boldsymbol{\kappa}_i\boldsymbol{\mu} + c_i^2\right) - 2\log\cosh\frac{c_i}{2}\right\} \\
&= \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}c_i}\left\{\frac{1}{2c_i}\tanh\left(\frac{c_i}{2}\right)\left(-\widetilde{K}_{ii} - \boldsymbol{\kappa}_i\Sigma\boldsymbol{\kappa}_i^\top - \boldsymbol{\mu}^\top\boldsymbol{\kappa}_i^\top\boldsymbol{\kappa}_i\boldsymbol{\mu} + c_i^2\right) - 2\log\cosh\frac{c_i}{2}\right\} \\
&= \frac{\mathrm{d}}{\mathrm{d}c_i}\left\{\frac{1}{4c_i}\tanh\left(\frac{c_i}{2}\right)\left(\underbrace{-\widetilde{K}_{ii} - \boldsymbol{\kappa}_i\Sigma\boldsymbol{\kappa}_i^\top - \boldsymbol{\mu}^\top\boldsymbol{\kappa}_i^\top\boldsymbol{\kappa}_i\boldsymbol{\mu}}_{:=-A_i}\right) + \frac{c_i}{4}\tanh\left(\frac{c_i}{2}\right) - \log\cosh\frac{c_i}{2}\right\} \\
&= \left(\frac{A_i}{4c_i^2} - \frac{1}{4}\right)\tanh\left(\frac{c_i}{2}\right) - \frac{1}{2}\left(\frac{A_i}{4c_i} - \frac{c_i}{4}\right)\left(1 - \tanh^2(\frac{c_i}{2})\right) \\
&= U(c_i)\left(\frac{c_i}{2}\left(1 - \tanh^2(\frac{c_i}{2})\right) - \tanh\left(\frac{c_i}{2}\right)\right),
\end{aligned}$$

where $U(c_i) = \frac{\Sigma_{ii}+\mu_i^2}{4c_i^2} - \frac{1}{4}$.

The gradient equals zero in two case. First, in the case $U(c_i) = 0$ which leads to[4]

$$c_i = \sqrt{\widetilde{K}_{ii} + \boldsymbol{\kappa}_i\Sigma\boldsymbol{\kappa}_i^\top + \boldsymbol{\mu}^\top\boldsymbol{\kappa}_i^\top\boldsymbol{\kappa}_i\boldsymbol{\mu}},$$

which is always valid since $\boldsymbol{\kappa}$, Σ and $\widetilde{K}$ are definite positive matrices. The second consists of the right hand side of the product being zero which leads to $c_i = 0$. The second derivative reveals that the first case always corresponds to a maximum and the second case to a minimum.

Global parameters We first compute the Euclidean gradients of the variational bound (10) w.r.t. the global parameters $\boldsymbol{\mu}$ and Σ. We obtain

$$\begin{aligned}
\frac{d\mathcal{L}}{d\boldsymbol{\mu}} &= \frac{1}{2}\frac{d}{d\boldsymbol{\mu}}\left(-\boldsymbol{\mu}^\top K_{mm}^{-1}\boldsymbol{\mu} + \boldsymbol{y}^\top\boldsymbol{\kappa}\boldsymbol{\mu} - \boldsymbol{\mu}^\top\boldsymbol{\kappa}^\top\Theta\boldsymbol{\kappa}\boldsymbol{\mu}\right) \\
&= \frac{1}{2}\left(-2K_{mm}^{-1}\boldsymbol{\mu} + \boldsymbol{\kappa}^\top\boldsymbol{y} - 2\boldsymbol{\kappa}^\top\Theta\boldsymbol{\kappa}\boldsymbol{\mu}\right) \\
&= -\left(K_{mm}^{-1} + \boldsymbol{\kappa}^\top\Theta\boldsymbol{\kappa}\right)\boldsymbol{\mu} + \frac{1}{2}\boldsymbol{\kappa}^\top\boldsymbol{y},
\end{aligned} \tag{15}$$

and

$$\begin{aligned}
\frac{d\mathcal{L}}{d\Sigma} &= \frac{1}{2}\frac{d}{d\Sigma}\left(\log|\,\Sigma\,| - \mathrm{tr}(K_{mm}^{-1}\Sigma) - \mathrm{tr}(\boldsymbol{\kappa}^\top\Theta\boldsymbol{\kappa}\Sigma)\right) \\
&= \frac{1}{2}\left(\Sigma^{-1} - K_{mm}^{-1} - \boldsymbol{\kappa}^\top\Theta\boldsymbol{\kappa}\right).
\end{aligned} \tag{16}$$

We now compute the natural gradients w.r.t. natural parameterization of the variational Gaussian distribution, i.e the parameters $\boldsymbol{\eta}_1 := \Sigma^{-1}\boldsymbol{\mu}$ and $\eta_2 = -\frac{1}{2}\Sigma^{-1}$. For a Gaussian distribution, properties of the Fisher information matrix expose the simplification that the natural gradient w.r.t. the natural parameters can be expressed in terms of the Euclidean gradient w.r.t. the mean and covariance parameters. It holds that

$$\widetilde{\nabla}_{(\boldsymbol{\eta}_1,\eta_2)}\mathcal{L}(\eta) = \left(\nabla_\mu\mathcal{L}(\eta) - 2\nabla_\Sigma\mathcal{L}(\eta)\mu,\ \nabla_\Sigma\mathcal{L}(\eta)\right), \tag{17}$$

where $\widetilde{\nabla}$ denotes the natural gradient and ∇ the Euclidean gradient. Substituting the Euclidean gradients (16) and (15) in to equation (17) we obtain the natural gradients

$$\begin{aligned}
\widetilde{\nabla}_{\eta_2}\mathcal{L} &= \frac{1}{2}\left(-2\eta_2 - K_{mm}^{-1} - \boldsymbol{\kappa}^\top\Theta\boldsymbol{\kappa}\right) \\
&= -\eta_2 - \frac{1}{2}\left(K_{mm}^{-1} + \boldsymbol{\kappa}^\top\Theta\boldsymbol{\kappa}\right)
\end{aligned}$$

[4] We omit the negative solution since $\mathrm{PG}(b, c) = \mathrm{PG}(b, -c)$.

and

$$\begin{aligned}\widetilde{\nabla}_{\boldsymbol{\eta}_1}\mathcal{L} &= -\left(K_{mm}^{-1}+\boldsymbol{\kappa}^\top\Theta\boldsymbol{\kappa}\right)\left(-\frac{1}{2}\eta_2^{-1}\boldsymbol{\eta}_1\right)+\frac{1}{2}\boldsymbol{\kappa}^\top\boldsymbol{y}-2\left(-\eta_2-\frac{1}{2}\left(K_{mm}^{-1}+\boldsymbol{\kappa}^\top\Theta\boldsymbol{\kappa}\right)\right)\left(-\frac{1}{2}\eta_2^{-1}\boldsymbol{\eta}_1\right)\\ &= \frac{1}{2}\boldsymbol{\kappa}^\top\boldsymbol{y}-\boldsymbol{\eta}_1.\end{aligned}$$

A.4. Natural gradient and coordinate ascent updates

If the full conditional distributions and the corresponding variational distribution belong to the same exponential family it is known in variational inference that "we can compute the natural gradient by computing the coordinate updates in parallel and subtracting the current setting of the parameter" (Hoffman et al., 2013). In our setting it is not clear if this relation holds since we do not consider the classic ELBO but a lower bound on it due to (8). Interestingly, the lower bound (8) does not break this property and our natural gradient updates correspond to coordinate ascent updates as we show in the following. Setting the Euclidean gradients and (15) to zero and using the natural parameterization gives

$$\eta_2 = -\frac{1}{2}\Sigma^{-1} = -\frac{1}{2}\left(K_{mm}^{-1}-\boldsymbol{\kappa}^\top\Theta\boldsymbol{\kappa}\right). \tag{18}$$

Setting (16) to zero yields

$$\boldsymbol{\mu} = \frac{1}{2}\left(K_{mm}^{-1}+\boldsymbol{\kappa}^\top\Theta\boldsymbol{\kappa}\right)^{-1}\boldsymbol{\kappa}^\top\boldsymbol{y}.$$

Substituting the update from above (18) and using natural parameterization results in

$$\boldsymbol{\eta}_1 = \frac{1}{2}\boldsymbol{\kappa}^\top\boldsymbol{y}.$$

This shows that using learning rate one in our natural gradient ascent scheme corresponds to employing coordinate ascent updates in the Euclidean parameter space.

A.5. Variational bound by Gibbs and MacKay

When using the full GP representation in our model and not the sparse approximation, the bound in our model is equal to the bound used by Gibbs and MacKay (2000). We provide a proof in the following.

Applying our variational inference approach to the joint distribution (5) gives the variational bound

$$\begin{aligned}\log p(\boldsymbol{y}|\,\boldsymbol{f}) &\geq \mathbb{E}_{q(\omega)}\left[\log p(\boldsymbol{y}|\,\boldsymbol{f},\boldsymbol{\omega})\right]-\mathrm{KL}(q(\boldsymbol{\omega})|p(\boldsymbol{\omega}))\\ &= \mathbb{E}_{q(\omega)}\left[\frac{1}{2}\boldsymbol{y}^\top\boldsymbol{f}-\frac{1}{2}\boldsymbol{f}^\top\Omega\boldsymbol{f}\right]-n\log(2)-\mathrm{KL}(q(\boldsymbol{\omega})|p(\boldsymbol{\omega}))\\ &= \frac{1}{2}\boldsymbol{y}^\top\boldsymbol{f}-\frac{1}{2}\boldsymbol{f}^\top\Theta\boldsymbol{f}-n\log(2)+\sum_{i=1}^{n}\left(\frac{c_i^2}{2}\theta_i-\log\cosh(c_i/2)\right).\end{aligned}$$

(Gibbs and MacKay, 2000) employ the following inequality on logit link

$$\sigma(z)\geq\sigma(c)\exp\left(\frac{z-c}{2}-\frac{\sigma(c)-1/2}{2c}(z^2-c^2)\right).$$

Using this bound in the setting of GP classification yields the following lower bound on the log-likelihood,

$$\begin{aligned}
\log p(\boldsymbol{y}|\,\boldsymbol{f}) =& \sum_{i=1}^{n} \log \sigma(y_i f_i) \\
\geq& \sum_{i=1}^{n} \left(\log \sigma(c_i) + \frac{y_i f_i - c_i}{2} - \frac{\sigma(c_i) - 1/2}{2c_i}((y_i f_i)^2 - c_i^2) \right) \\
=& \sum_{i=1}^{n} \left(-\log\cosh(c_i/2) - \log(2) + \frac{y_i f_i}{2} - \frac{\sigma(c_i) - 1/2}{2c_i}(f_i^2 - c_i^2) \right) \\
=& \sum_{i=1}^{n} \left(-\log\cosh(c_i/2) - \log(2) + \frac{y_i f_i}{2} - \frac{1}{4c_i}\tanh(c_i/2)(f_i^2 - c_i^2) \right) \\
=& \sum_{i=1}^{n} \left(-\log\cosh(c_i/2) - \log(2) + \frac{y_i f_i}{2} - \frac{1}{2}\theta_i(f_i^2 - c_i^2) \right) \\
=& \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{f} - \frac{1}{2}\boldsymbol{f}^\top \Theta \boldsymbol{f} - n\log(2) + \sum_{i=1}^{n} \left(\frac{c_i^2}{2}\theta_i - \log\cosh(c_i/2) \right),
\end{aligned}$$

where we made use of the fact that $\sigma(x) - 1/2 = \tanh(x/2)/2$. This concludes the proof.

A.6. Perturbative correction

Based on the perturbative theory proposed by Opper et al. (2015), we compute correction terms for the parameters of the latent GP f. The correction terms for the mean and the variance of the latent variables are :

$$\begin{aligned}
f_{\text{corr}} &= f_q + \sum_i \text{Cov}_q(f, f_i)\Sigma_{ii}\mu_i \text{Var}_q(\omega_i) \\
\sigma_{\text{corr}} &= \sigma_q + f_q^2 - f_{\text{corr}}^2 + \frac{1}{2}\sum_i \text{Cov}_q^2(f, f_i)\left(\mu_i - \Sigma_{ii}\right)\text{Var}_q(\omega_i)
\end{aligned}$$

Where f_i is the latent function of the training point x_i, $f_q = \mathbb{E}_q[f]$, $\sigma_q = \mathbb{E}_q[\sigma]$, $\mu_i = \mathbb{E}_q[f_i]$ and $\Sigma_{ii} = \text{Var}_q(f_i)$. The variance of ω is given by :

$$\text{Var}_q(\omega) = \frac{1}{2c^2}\left(\frac{\tanh^2\left(\frac{c}{2}\right) - 1}{2} + \frac{\tanh\left(\frac{c}{2}\right)}{c} \right)$$

The perturbative theory builds on the difference $V = H - H_q$ between the true probability measure $p(f,\omega) = \frac{1}{Z}\mu(f,\omega)e^{-H(f,\omega)}$ and our variational approximation $q(f,\omega) = \frac{1}{Z_q}\mu(f,\omega)e^{-H_q(f,\omega)}$. We define $p_\lambda(f,\omega) = \frac{1}{Z_\lambda}\mu(f,\omega)e^{-H_\lambda(f,\omega)}$, where $H_\lambda = H_q + \lambda V$, as our corrected distribution. Expectation of functions given this corrected distribution can be achieved through a Taylor expansion and setting λ to 1. For our model we used a second order expansion resulting in :

$$\begin{aligned}
\mathbb{E}_\lambda[f(x)] &\approx \mathbb{E}_q[f(x)] + \frac{1}{2}\text{Cov}\left(f(x), V^2\right) \\
&= \mathbb{E}_q[(f(x)] + \frac{1}{8}\left(\sum_i \left\{ \text{Cov}(f(x), f_i^4) - 2\text{Cov}((f(x), f_i^2)\mathbb{E}_q\left[f_i^2\right] \right\} \text{Var}_q(\omega_i) \right)
\end{aligned}$$