
Multi-Class Gaussian Process Classification Made Conjugate: Efficient Inference via Data Augmentation

Théo Galy-Fajou *

TU Berlin
Germany

Florian Wenzel *

TU Kaiserslautern
Germany

Christian Donner

TU Berlin
Germany

Manfred Opper

TU Berlin
Germany

Abstract

We propose a new scalable multi-class Gaussian process classification approach building on a novel modified softmax likelihood function. The new likelihood has two benefits: it leads to well-calibrated uncertainty estimates and allows for an efficient latent variable augmentation. The augmented model has the advantage that it is conditionally conjugate leading to a fast variational inference method via block coordinate ascent updates. Previous approaches suffered from a trade-off between uncertainty calibration and speed. Our experiments show that our method leads to well-calibrated uncertainty estimates and competitive predictive performance while being up to two orders faster than the state of the art.

1 INTRODUCTION

In real-world decision making systems, it is important that classification methods do not only provide accurate predictions, but also indicate when they are likely to be incorrect. Calibrated confidence estimates are important in many application domains such as self driving cars (Bojarski et al., 2016), medical diagnosis (Caruana et al., 2015) and speech recognition (Xiong et al., 2016).

In multi-class classification tasks, modern deep neural networks achieve state-of-the-art accuracies but often suffer from bad calibration (Guo et al., 2017). Gaussian process (GP) models provide an attractive alternative approach to multi-class classification problems.

Due to the Bayesian treatment of uncertainty, GPs have the advantage of leading to well-calibrated uncertainty estimates (Williams and Barber, 1998; Rasmussen and Williams, 2005). Furthermore, GP models become more

expressive as the number of data points grows and allow for incorporating prior knowledge by using different kernel functions. However, inference in multi-class GP classification models is challenging.

In the easier setting of *binary* classification, GPs can be applied to big datasets using variational inference methods (Hensman and Matthews, 2015; Wenzel et al., 2019). This is possible because the expectation of generic log-likelihoods in the variational objective (the so-called ELBO) over the variational distribution (typically a Gaussian) reduces to univariate integrals which can be performed in an efficient way by using numerical quadrature methods. The optimization of the variational objective can then be achieved by stochastic gradient methods involving mini-batches. A further speedup of such methods is possible by the application of natural gradient techniques (Salimbeni et al., 2018).

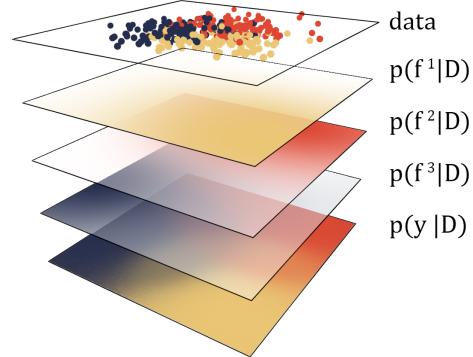


Figure 1: In a GP multi-class classification model, each class density is modeled by an individual GP $p(f^c|D)$. For predictions $p(y|D)$, the latent GPs are marginalized out.

The *multi-class* problem is more complicated because it involves not only one latent GP, but one GP for each class. In the common multi-class likelihoods, as e.g. the

*Equal contribution. Contact: galy-fajou@tu-berlin.de.

softmax function, the GPs are coupled. This leads to complicated multivariate integrals which make a direct application of variational inference techniques intractable. Previous inference methods for the softmax model rely on approximations and do not scale (Williams and Barber, 1998; Chai, 2012).

To tackle this issue, Hernández-Lobato et al. (2011) propose an alternative to the softmax, the *robust-max* likelihood. This likelihood simplifies the problem by focusing mainly on the maximal latent GP and discarding information of the other less likely classes. The model is robust against outliers and often yields good classification accuracy. However, it sacrifices the gradual response of the traditional softmax for an all-or-nothing criterion leading to bad uncertainty quantification.

In problems with well separated classes and a few outliers, the robust-max likelihood is an excellent choice, while in problems with overlapping classes a gradual classification criterion is more desirable (Xiong et al., 2010). In this work, we introduce a novel likelihood, the *logistic softmax* likelihood, which combines the best of both worlds. It has a gradual classification criterion similar to the traditional softmax, but on the other hand also enables fast inference.

We propose an augmentation approach that renders the model conditionally conjugate. Inference in the augmented model is much easier. We derive a fast *variational inference* algorithm based on closed-form updates. Our inference approach is faster and more stable than the state of the art since it uses efficient block coordinate ascent updates and does not rely on sampling.

Alternatively, the conditionally conjugate form of the augmented model directly leads to another inference strategy. If we are willing to pay more computation time, we obtain *exact samples* from the true posterior by a Gibbs sampling scheme. Our main contributions are as follows:

- We introduce a new multi-class GP classification model building on a modification of the softmax likelihood function. By applying a variable augmentation approach, we render the model conditionally conjugate.
- We propose an efficient stochastic variational inference scheme which is based on block coordinate ascent updates. Unlike in previous work, all updates are given in closed-form and do not rely on numerical quadrature methods or sampling.
- Our method scales to datasets with many data points and a large number of classes. The experiments show that our method is faster than the state-of-the-art while leading to competitive prediction performance.

- We solve the calibration issue of the robust-max likelihood as our model leads to much better uncertainty quantification.

The paper is structured as follows. Section 2 introduces the problem of multi-class GP classification and reviews related work. In Section 3 we introduce the new model and present a data augmentation strategy that renders the model conditionally conjugate. In Section 4 we present an efficient inference algorithm. We show experimental results in Section 5. Finally, Section 6 concludes and lays out future research directions. Our code is included in a Julia package¹.

2 BACKGROUND AND RELATED WORK

We begin our review by introducing the multi-class GP classification model. Related work can be grouped into approaches that consider alternative likelihood functions or apply data augmentation strategies.

Multi-class GP classification. We consider a dataset of N data points $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ with labels $\mathbf{y} = (y_1, \dots, y_N)$, where $y_i \in \{1, \dots, C\}$ and C is the total number of classes. The multi-class GP classification model consists of a latent GP prior for each class $\mathbf{f} = (f^1, \dots, f^C)$, where $f^c \sim \text{GP}(0, k^c)$ and k^c is the corresponding kernel function. The labels are modeled by a categorical likelihood

$$p(y_i = k | \mathbf{x}_i, \mathbf{f}_i) = g^k(\mathbf{f}(\mathbf{x}_i)), \quad (1)$$

where $g^k(f)$ is a function that maps the real vector of the GP values to a probability vector.

The most common way to form a categorical likelihood is through the softmax transformation

$$p(y_i = k | \mathbf{f}_i) = \frac{\exp(f_i^k)}{\sum_{c=1}^C \exp(f_i^c)}, \quad (2)$$

where we use the shorthand $f_i^c = f^c(\mathbf{x}_i)$ and for the sake of clarity we omit the conditioning on x_i .

There have been several early works addressing multi-class GP classification with a softmax likelihood (Williams and Barber, 1998; Kim and Ghahramani, 2006; Chai, 2012; Riihimäki et al., 2013). Nevertheless, these methods do not scale well with the number of data points. Izmailov et al. (2018) use tensor train decomposition to use high numbers of inducing points but do not provide efficient closed-form updates.

¹<https://github.com/theogf/AugmentedGaussianProcesses.jl>

The robust-max likelihood. Recently, there have been advances to scale multi-class GP classification to big datasets by changing the likelihood. Hernández-Lobato et al. (2011) propose the *robust-max* likelihood

$$p(y = k | \mathbf{f}) = (1 - \epsilon) \prod_{c \neq y}^C \Theta(f^k - f^c) + \frac{\epsilon}{C}, \quad (3)$$

where ϵ is the probability of a labeling error, and Θ is the Heaviside function. This likelihood simplifies the problem as it leads to a decoupling of the latent GPs.

Originally, the authors propose an expectation propagation (EP) based approach which only scales to small datasets. Hensman et al. (2015) and Salimbeni et al. (2018) scale this model to big datasets employing a variational inference approach but rely on numerical quadrature. As we show later, this likelihood has the big disadvantage of leading to poor confidence calibration.

The Heaviside likelihood. Villacampa-Calvo and Hernández-Lobato (2017) build on the Heaviside likelihood

$$p(y = k | \mathbf{f}) = \prod_{c \neq y}^C \Theta(f^k - f^c), \quad (4)$$

where Θ is again the Heaviside function. The authors propose a scalable expectation propagation approach but have to make approximations on the likelihood. The inference is still slow and the applicability to big datasets is limited.

Data augmentation. Other approaches consider probabilistic data augmentation. Wenzel et al. (2019) propose an augmentation approach for binary GP classification leading to a conditionally conjugate model, but are limited to the binary classification setting. Linderman et al. (2015) consider data augmentation for multinomial likelihoods but focus on sampling. The approach has the disadvantage of breaking the symmetry between the classes and is limited to small datasets. Polson et al. (2013) propose conditionally conjugate Pólya-Gamma augmentation for the softmax likelihood (extended by Češnovar and Štrumbelj (2017) to GPU support) which is suitable for sampling but cannot be used for obtaining an efficient variational inference algorithm since the ELBO is intractable. Girolami and Rogers (2006) propose an augmentation strategy to multinomial probit regression but does not scale. Ruiz et al. (2018) propose an augmentation approach for enabling subsampling of classes for parametric models with categorical likelihoods. The approach is limited to parametric models and cannot be applied to GP models.

3 MULTI-CLASS GAUSSIAN PROCESS CLASSIFICATION

We formulate a multi-class GP classification model which leads to well calibrated confidences and is amenable to fast inference. We define a new likelihood function, termed the *logistic-softmax*, which shares the good prediction properties of the softmax. But in addition, it has the advantage that it allows for a data augmentation approach which renders the model conditionally conjugate. The augmented posterior can then be efficiently approximated by a structured mean-field variational inference method resulting in a fast algorithm with closed-form updates.

3.1 THE LOGISTIC-SOFTMAX GP MODEL

We consider the multi-class GP classification model as described in eq. 1. Different functions g for mapping real vectors to probability vectors that have been considered in literature include the softmax (eq. 2), the multinomial probit (Albert and Chib, 1993), the robust-max likelihood (eq. 3) and the Heaviside likelihood (eq. 4).

In this work, we propose the *logistic-softmax*:

$$p(y_i = k | \mathbf{f}_i) = \frac{\sigma(f_i^k)}{\sum_{c=1}^C \sigma(f_i^c)}, \quad (5)$$

where $\sigma(z) = (1 + \exp(-z))^{-1}$ is the logistic function. Our likelihood is a modified version of the softmax likelihood which replaces the inner exponential functions by logistic functions. Alternatively, it can be interpreted as the standard softmax applied to a non-linearly transformed GP, i.e. $p(y_i | \mathbf{f}_i) = \text{softmax}(\log \sigma(\mathbf{f}_i))$. The likelihood reduces to the binary logistic likelihood for $C = 2$.

In the following section we derive a three steps augmentation scheme, where we (i) decouple the GP latent variables f_i^k in the denominator by the introduction of a set of auxiliary λ -variables, (ii) further simplify the model likelihood by introducing Poisson random variables, and finally (iii) use a Pólya–Gamma representation of the sigmoid function (Polson et al., 2013) to achieve the desired conjugate representation of the model.

3.2 CONJUGATE AUGMENTATION

We expand the logistic-softmax likelihood (5) by three data augmentation steps leading to a conditionally conjugate model. The final model is displayed in Figure 2. In the following we present the augmentations.

Augmentation 1: Gamma augmentation. To remedy the intractable normalizer term we make use of the integral

identity $\frac{1}{z} = \int_0^\infty \exp(-\lambda z) d\lambda$ and express the likelihood (5) as

$$\begin{aligned} p(y_i = k | \mathbf{f}_i) &= \frac{\sigma(f_i^k)}{\sum_{c=1}^C \sigma(f_i^c)} \\ &= \sigma(f_i^k) \int_0^\infty \exp\left(-\lambda_i \sum_{c=1}^C \sigma(f_i^c)\right) d\lambda_i. \end{aligned}$$

This augmentation is well known in the Gibbs sampling community to deal with intractable normalization constants (see e.g. Walker (2011)) but is not often used in the setting of variational inference. By interpreting λ_i as an additional latent variable we obtain the augmented likelihood

$$p(y_i = k | \mathbf{f}_i, \lambda_i) = \sigma(f_i^k) \prod_{c=1}^C \exp(-\lambda_i \sigma(f_i^c)), \quad (6)$$

and we impose the improper prior $p(\lambda_i) \propto \mathbb{1}_{[0, \infty)}(\lambda_i)$. The improper prior is not problematic since it leads to a proper complete conditional distribution as we will see in the end of the section.

Augmentation 2: Poisson augmentation. We rewrite the exponential factors in (6) based on the moment generation function of the Poisson distribution $\text{Po}(\cdot | \lambda)$ which is

$$\exp(\lambda(z - 1)) = \sum_{n=0}^{\infty} z^n \text{Po}(n | \lambda).$$

Using $z = \sigma(-f)$ and the fact that $\sigma(f) = 1 - \sigma(-f)$ we rewrite the exponential factors as

$$\begin{aligned} \exp(-\lambda_i \sigma(f_i^c)) &= \exp(\lambda_i(\sigma(-f_i^c) - 1)) \\ &= \sum_{n_i^c=0}^{\infty} (\sigma(-f_i^c))^{n_i^c} \text{Po}(n_i^c | \lambda_i), \end{aligned}$$

which leads to the augmented likelihood

$$p(y_i = k | \mathbf{f}_i, \lambda_i, \mathbf{n}_i) = \sigma(f_i^k) \prod_{c=1}^C (\sigma(-f_i^c))^{n_i^c}, \quad (7)$$

where $\mathbf{n}_i = (n_i^1, \dots, n_i^C)$ and the augmented Poisson variables are distributed as $p(n_i^c | \lambda_i) = \text{Po}(n_i^c | \lambda_i)$, see e.g. Donner and Opper (2017, 2018). Note that this augmentation is only possible since the transformation on f_i^c is bounded, hence the need for a modified likelihood.

Augmentation 3: Pólya-Gamma augmentation. In the last augmentation step, we aim for a Gaussian representation of the sigmoid function. The Pólya-Gamma

representation (Polson et al., 2013) allows for rewriting the sigmoid function as a scale mixture of Gaussians

$$\sigma(z)^n = \int_0^\infty 2^{-n} \exp\left(\frac{nz}{2} - \frac{z^2}{2}\omega\right) \text{PG}(\omega | n, 0), \quad (8)$$

where $\text{PG}(\omega | n, b)$ is a Pólya-Gamma distribution. Pólya-Gamma variables are well suited for augmentations since the moments are known analytically and an efficient sampler exists (Polson et al., 2013). By applying this augmentation to (7) we obtain

$$\begin{aligned} p(y_i = k | \mathbf{f}_i, \lambda_i, \mathbf{n}_i, \boldsymbol{\omega}_i) &= \\ \prod_{c=1}^C 2^{-(y_i'^c + n_i^c)} \exp\left(\frac{(y_i'^c - n_i^c)f_i^c}{2} - \frac{(f_i^c)^2}{2}\omega_i^c\right), \end{aligned} \quad (9)$$

where $\boldsymbol{\omega}_i = (\omega_i^1, \dots, \omega_i^C)$ are Pólya-Gamma variables with distributions

$$p(\boldsymbol{\omega}_i | \mathbf{n}_i, y_i) = \prod_{c=1}^C \text{PG}(\omega_i^c | y_i'^c + n_i^c, 0),$$

where \mathbf{y}' is an $N \times C$ -dimensional one-hot encoding of the labels, i.e. $y_i'^c$ is 1 if $y_i = c$, and 0 otherwise. Details are deferred to appendix A.1.

Realizing that (9) has a Gaussian form with respect to \mathbf{f}_i we achieved our goal of a conjugate representation of the latent GPs. As we will show in the next paragraph the model is also conditionally conjugate for the augmented variables.

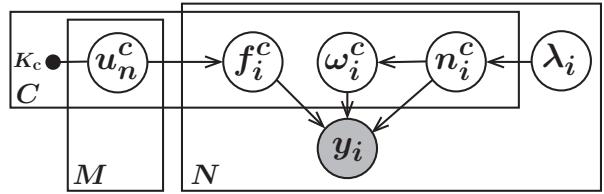


Figure 2: The final augmented model as presented in Section 3.2. Shaded circles represent observable variables, empty circles latent variables and dots hyperparameters.

The final model. The effort of the augmentations finally pays off as the final augmented model is now tractable and the complete conditional distributions are given in closed-form.

The complete conditionals of the GPs \mathbf{f}^c are

$$p(\mathbf{f}^c | \mathbf{y}, \boldsymbol{\omega}^c, \mathbf{n}^c) = \mathcal{N}\left(\mathbf{f}^c | \frac{1}{2} A^c (\mathbf{y}'^c - \mathbf{n}^c), A^c\right),$$

where the conditional covariance matrix is given by $A^c = (\text{diag}(\boldsymbol{\omega}^c) + K_c^{-1})^{-1}$ and K_c is the kernel matrix of the

GP \mathbf{f}^c . For the conditional distribution of λ we get

$$p(\lambda_i | \mathbf{n}_i) = \text{Ga}\left(\lambda_i | 1 + \sum_{c=1}^C n_i^c, C\right),$$

where $\text{Ga}(\cdot | a, b)$ denotes a gamma distribution with shape parameter a and rate parameters b . The improper prior on λ_i does not impose an issue since the complete conditional distribution is proper.

For the Poisson variables \mathbf{n} , we get

$$p(n_i^c | f_i^c, \lambda_i) = \text{Po}(n_i^c | \lambda_i \sigma(f_i^c)),$$

Finally, for the Pólya-Gamma variables ω the complete conditional distributions are

$$p(\omega_i^c | n_i^c, f_i^c, y_i) = \text{PG}(\omega_i^c | y_i^c + n_i^c, |f_i^c|).$$

4 INFERENCE

We derive a variational approximation of the posterior of the augmented model (9). In the following we develop an efficient stochastic variational inference (SVI) algorithm that is based on closed-form block coordinate ascent updates. Our method allows both for subsampling of data points and of outcomes (classes) scaling to datasets with a large number of data points and a large number of classes.

4.1 VARIATIONAL APPROXIMATION

To scale our model to big datasets, we approximate the latent GPs \mathbf{f}^c by *sparse GPs* building on *inducing points*. For each GP \mathbf{f}^c , we introduce M inducing points \mathbf{u}^c and connect the GP values with the inducing points via the joint prior distribution $p(\mathbf{f}^c, \mathbf{u}^c)$ given in [Titsias \(2009\)](#). Details on variational sparse GP approximations can be found in [Titsias \(2009\)](#); [Hensman et al. \(2013\)](#).

We approximate the posterior distribution of the latent sparse GPs \mathbf{u} and the augmented variables $\lambda, \mathbf{n}, \omega$ by assuming the following structure of the variational distribution $q(\mathbf{u}, \lambda, \mathbf{n}, \omega) = q(\mathbf{u}, \lambda)q(\mathbf{n}, \omega)$. Note that the only assumption on the variational posterior is the decoupling of two groups of variables. Since our model is conditionally conjugate, the family of the optimal variational distribution can be easily determined by averaging the complete conditionals in log-space ([Blei et al., 2017](#)). From the above decoupling assumption, it follows that the optimal variational posterior has a factorizing form $q(\mathbf{u}, \lambda, \mathbf{n}, \omega) = q(\mathbf{u})q(\lambda)q(\omega, \mathbf{n})$ and the factors are

$$\begin{aligned} q(\mathbf{u}) &= \prod_c \mathcal{N}(\mathbf{u}^c | \boldsymbol{\mu}^c, \Sigma^c), \quad q(\lambda) = \prod_i \text{Ga}(\lambda_i | \alpha_i, \beta_i), \\ q(\omega, \mathbf{n}) &= \prod_{i,c} \text{PG}(\omega_i^c | y_i^c + n_i^c, b_i^c) \text{Po}(n_i^c | \gamma_i^c), \end{aligned}$$

where $\boldsymbol{\mu}^c, \Sigma^c, \alpha_i, \beta_i, b_i^c, \gamma_i^c$, for all $i \in \{1, \dots, N\}$ and $c \in \{1, \dots, C\}$ are the *variational parameters*. The variational parameters are optimized by a coordinate ascent scheme outlined in Section 4.2. Finally, the approximate posterior of the sparse GPs $q^*(\mathbf{u})$ can be used to obtain an approximate posterior of the original latent GPs \mathbf{f} by $q^*(\mathbf{f}) := \int p(\mathbf{f} | \mathbf{u})q(\mathbf{u})d\mathbf{u}$ which is given in closed-form (see e.g., [Hensman and Matthews, 2015](#)).

4.2 VARIATIONAL INFERENCE

Building on the conditionally conjugate representation of our model deriving efficient variational parameter updates is straightforward. We implement the classic SVI algorithm described by [Hoffman et al. \(2013\)](#), which builds on block coordinate ascent updates. We iteratively optimize each factor of the variational distribution, while holding the others fixed. The variational parameters of each factor are directly set to the optimal value given the other parameters.

We compute the block coordinate ascent (CAVI) updates in closed-form by averaging the parameters of each complete conditional in log space ([Blei et al., 2017](#)) and details are deferred to appendix A.2. When using minibatches of the data, each global variational parameter (i.e. $\boldsymbol{\mu}^c$ and Σ^c) is updated using a convex combination of the old parameter and the CAVI update, which corresponds to a natural gradient ascent scheme ([Hoffman et al., 2013](#)). Remarkably, the negative ELBO in our augmented model is convex in the global parameters (see appendix A.5 for the proof). Therefore, our algorithm is ensured to converge to the global optimum ([Hoffman et al., 2013](#)). The inference algorithm is summarized in Alg. 1 and its complexity is $\mathcal{O}(CM^3)$.

Extreme classification. When the number of possible outcomes (classes) C is very large, using probabilistic multi-class models becomes generally computationally expensive as the likelihood (categorical distribution) scales linearly with the number of classes. Using large categorical distributions is a challenging problem ([Ruiz et al., 2018; Titsias, 2016](#)).

With a slight modification, our method can deal with an extreme classification setting (large number of classes). In our augmentation, the GPs in the normalizer term are decoupled and allow for subsampling of the classes. This reduces the complexity to $\mathcal{O}(M^3)$, i.e. being independent of the number of classes. We provide details in appendix A.3. This approach is especially useful when using shared hyperparameters among the class specific latent GPs.

Algorithm 1 Conjugate multi-class Gaussian process classification

```

1: Input: data  $\mathbf{X}, \mathbf{y}$ , minibatch size  $|\mathcal{S}|$ 
2: Output: variational posterior GPs  $p(\mathbf{u}^c | \mu^c, \Sigma^c)$ 
3: Set the learning rate schedules  $\rho_t, \rho_t^h$  appropriately
4: Initialize all variational parameters and hyperparameters
5: Select  $M$  inducing points locations (e.g. kMeans)
6: for iteration  $t = 1, 2, \dots$  do
7:   # Sample minibatch:
8:   Sample a minibatch of the data  $\mathcal{S} \subset \{1, \dots, N\}$ 
9:   # Local variational updates
10:  for  $i \in \mathcal{S}$  do
11:    Update  $(\alpha_i, \gamma_i)$  (Eq. 12,13)
12:    for each class  $c$  do
13:      Update  $b_i^c$  (Eq. 14)
14:    end for
15:  end for
16:  # Global variational GP updates
17:  for each class  $c$  do
18:     $\mu^c \leftarrow (1 - \rho_t)\mu^c + \rho_t\hat{\mu}^c$  (Eq. 15)
19:     $\Sigma^c \leftarrow (1 - \rho_t)\Sigma^c + \rho_t\hat{\Sigma}^c$  (Eq. 16)
20:  end for
21:  # Hyperparameter updates
22:  Gradient step  $h \leftarrow h + \rho_t^h \nabla_h \mathcal{L}$ 
23: end for

```

Predictions. The posterior distribution of the latent function $p(f_*^c | x_*, \mathbf{y})$ at a new test point x_* is approximated by

$$q(f_*^c | x^*, \mathbf{y}) = \int p(f_*^c | \mathbf{u}^c) q(\mathbf{u}^c) d\mathbf{u} = \mathcal{N}\left(f_*^c | \mu_*^c, \sigma_*^{2c}\right),$$

where the mean is $\mu_*^c = K_{*\mathbf{m}}^c K_{mm}^{-1} \mu^c$ and the variance $\sigma_*^{2c} = K_{**}^c + K_{*\mathbf{m}}^c K_{mm}^{-1} (\Sigma^c K_{mm}^{-1} - I) K_{*\mathbf{m}}^c$. The matrix $K_{*\mathbf{m}}$ denotes the kernel matrix between the test point and the inducing points and K_{**} the kernel value of the test point. The final approximate predictive distribution of a test label is

$$p(y = k | x_*, \mathbf{y}) \approx \int p(y = k | \mathbf{f}_*) \prod_{c=1}^C q(f_*^c | x^*, \mathbf{y}) d\mathbf{f}_*,$$

where $p(y = k | \mathbf{f}_*)$ is the logistic-softmax likelihood. This is a C -dimensional analytically intractable integral. We approximate it by Monte Carlo integration. For faster convergence, the random samples can be replaced by Quasi-Monte Carlo sequences (Owen, 1998; Buchholz et al., 2018). Finally, a point is classified by the highest predictive likelihood, $y_i^* = \arg \max_{c \in C} p(y_i = c | \mathbf{f}_*)$.

Optimization of the hyperparameters. We select the optimal kernel hyperparameters by maximizing the

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

4.3 GIBBS SAMPLING

Since our augmented model is conditionally conjugate we can directly derive a Gibbs sampling scheme. In order to sample from the *exact posterior*, we alternate between drawing a sample from each complete conditional distributions. The augmented variables are naturally marginalized out and asymptotically, the latent GP samples will be from the true posterior.

5 EXPERIMENTS

In this section we empirically answer the following questions:

- What is the effect of using the softmax, logistic-softmax, robust-max and Heaviside likelihood on predictive performance and calibration quality? (Section 5.1)
- How does the augmentation affect the predictive performance? (Section 5.2)
- How does our method perform compared to other state-of-the-art GP based multi-class classification methods? (Section 5.4)

In all experiments we use a squared exponential covariance function with automatic relevance determination (ARD): $k(\mathbf{x}, \mathbf{x}') = \eta \exp\left(-\sum_{d=1}^D \frac{(x_d - x'_d)^2}{2l_d^2}\right)$, where we set the initial variance η to 1 and the length scales l are initialized to the median of the pairwise distance matrix of the data. The hyperparameters are optimized using Adam (Kingma and Ba, 2015). We use a collection of datasets from the LIBSVM repository². Every dataset has been normalized to mean 0 and variance 1. For each method, we use 200 inducing points, unless stated otherwise. The initial inducing points locations are determined by the kmeans++ algorithm (Arthur and Vassilvitskii, 2007). We find that fixing the locations while training gives good results. We use a mini-batch size of 200 and all experiments are performed on a single CPU.

²<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html>

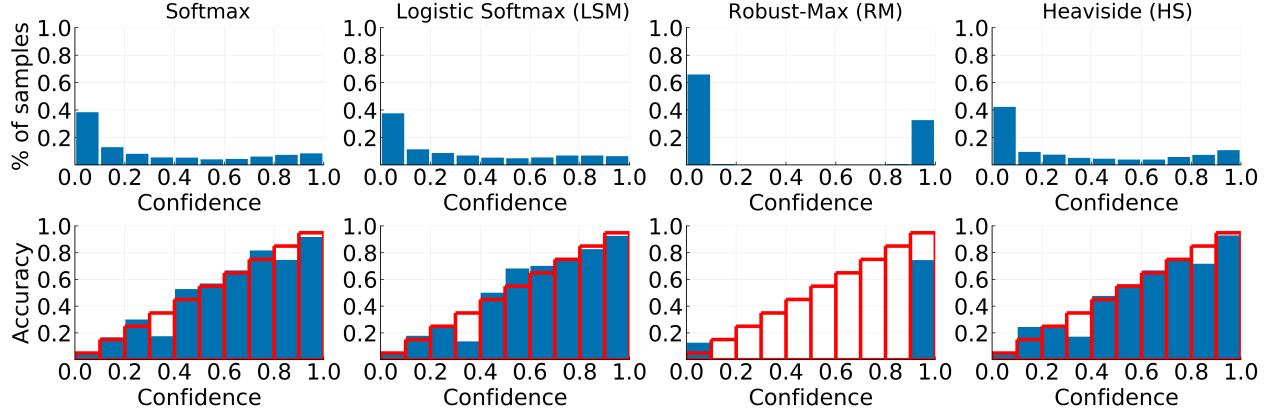


Figure 3: Likelihood comparison: Confidence histograms (top) and reliability diagrams (bottom) for four different likelihood models. The robust-max model always predicts with probability either close to one or close to zero leading to a poor confidence calibration.

5.1 LIKELIHOODS COMPARISON

We begin the experiments by investigating the effect of using different likelihood functions. We compare our novel logistic-softmax (eq. 5), the softmax (eq. 2), the robust-max (eq. 3) and the Heaviside likelihood (eq. 4). For each model we employ variational inference to obtain an approximate posterior. In this experiment, no augmentation is used and the gradients are estimated by sampling.

To investigate uncertainty calibration, we create seven different toy datasets of 500 points with three classes. The data is generated from a mixture of Gaussians model with different variances σ^2 . For $\sigma^2 = 0$, the classes are sharply separated and for $\sigma^2 = 1$, the classes highly overlap and are almost indistinguishable.

See appendix A.4 for a visualization of the decision boundaries of the different methods. In Figure 4 we plot test error, negative log-likelihood and calibration error as function of the noise in the data. The (expected) calibration error is a summary statistic of calibration and is computed by the expectation between confidence and accuracy in the reliability diagram (c.f. Guo et al., 2017).

For datasets where the classes are sharply separated (small σ^2), all models perform similarly. But for datasets where classes overlap (high σ^2), the robust-max performs poorly due to bad uncertainty calibration.

In Figure 3 we show the confidence histograms and reliability diagrams for one dataset ($\sigma^2 = 0.5$). The diagrams are generated according to Naeini et al. (2015); Guo et al. (2017) – the reliability diagram displays the accuracy as function of confidence (a perfectly calibrated model would produce the identity function) and the confidence histogram shows the empirical distribution of the prediction confidence. The robust-max model fails to provide

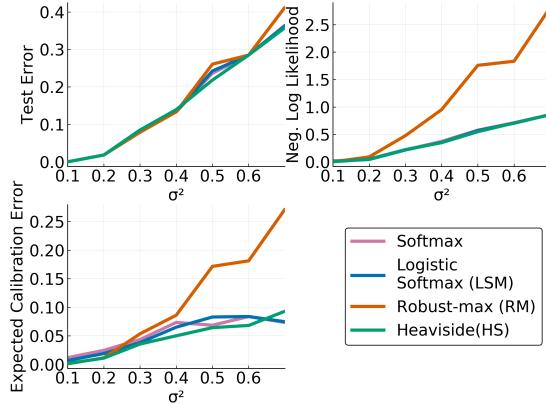


Figure 4: Likelihood comparison: The test error, negative log-likelihood and calibration error are plotted as function of the noise (σ^2) in the generated dataset. For highly overlapping classes (large σ^2), the robust-max likelihood yields poor calibration and bad log-likelihood values.

sensitive uncertainty estimates and only predicts with either probability close to zero or close to one. The softmax, logistic-softmax and Heaviside likelihood yield similar predictive performance and confidence calibration. However, as the following experiments show, our approach is much faster than the softmax and Heaviside model. It is the only scalable approach that leads to well calibrated confidences and the logistic-softmax can be used as an efficient replacement of the standard softmax.

5.2 EFFECT OF THE AUGMENTATION

We investigate the effect of the augmentation of the logistic-softmax model and its variational approximation. To this end we compare three different inference methods (1) variational inference for our augmented model (*Aug-*

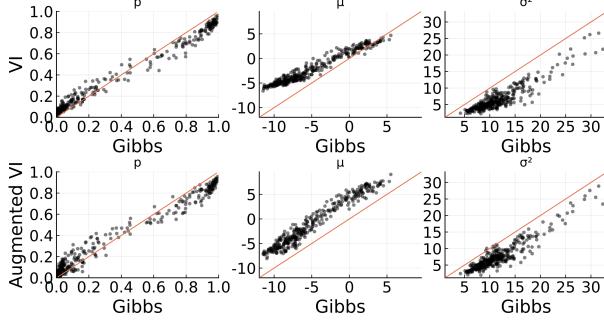


Figure 5: Effect of the augmentation: Comparison of the predictive marginals (p), posterior mean (μ) and posterior variance (σ^2) on a test set. Each plot shows the ground truth of the Gibbs sampler on the x-axis. On the y-axis the estimated values by variational inference without augmentation VI (top) and augmented variational inference $Augmented\ VI$ are shown (bottom). Our efficient augmented VI method produces values very close to the less efficient VI method. Both methods slightly overestimate the mean (μ) and underestimate the variance (σ^2). However, for both methods the final predictions (p) are close to the ground truth.

mented VI), (2) variational inference without augmentation (approximating the posterior of the original model from section 3.1 using a variational Gaussian), where the gradients are computed via sampling (VI) and (3) Gibbs sampling (*Gibbs*), c.f. Section 4.3. After burn-in, the samples from the Gibbs sampler serve as ground truth since they come from the exact posterior. In this experiment we do not use the inducing point approximation and all hyperparameters are fixed. We apply all three methods on the dataset Wine (3 classes) and compare the predictive likelihood (p) and the mean (μ) and variance (σ^2) of the latent GPs on a test set. We compare each entry of the three-dimensional vectors p, μ, σ^2 with the ground truth and display the results for all classes $c = 1, 2, 3$ combined in Figure 5.

Variational inference in the augmented model results in an approximate posterior which is very close to the variational inference solution in the original model. Both methods lead to a similar slight approximation error of the posterior mean μ and variance σ^2 and give predictive marginals p close to the ground truth. The Gibbs sampling approach has a final prediction accuracy of 0.98, whereby both variational inference methods have a final accuracy of 0.96. We find that the augmentation approach can be used as a scalable alternative to standard variational inference.

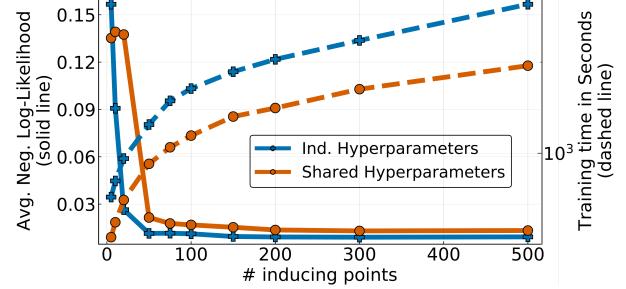


Figure 6: Inducing points and hyperparameters: The trade-off between predictive performance and run time is shown. Two versions of our method are used: individual hyperparameters for each GP (blue) and shared hyperparameters (orange). On the left y-axis we plot the negative log-likelihood (solid line) and on the right y-axis the training time (dashed line) as function of the number of inducing points.

5.3 HYPERPARAMETERS

In this experiment we answer two questions. What is the effect of the number of inducing points and what is the difference between using shared hyperparameters and individual hyperparameters for each latent GP? We train our model on the Shuttle dataset (58,000 points, 9 classes) for 200 epochs. We vary the number of inducing points from 5 to 400, and set the GP hyperparameters to be either shared or independent among classes.

In Figure 6 we display the trade-off between predictive performance and training time. We plot the negative log-likelihood (solid lines, y-axis left) and training time (dashed lines, y-axis right) as a function of the number of inducing points. If the number of inducing points is increased, the negative log-likelihood goes down and, oppositely, the training time goes up. We find that using only 200 inducing points already leads to near optimal predictive performance. Using independent hyperparameters over shared hyperparameters does not lead to a significant improvement of the predictive performance but implies a higher computational cost, especially for datasets with a large number of classes.

5.4 NUMERICAL COMPARISON

Finally, we evaluate the predictive performance and convergence speed of our method against other state-of-the-art multi-class GP classification approaches. We compare our logistic-softmax likelihood with augmentation based approach (**LSM**) against two competitors. First, the robust-max likelihood model (**RM**) by Hensman and Matthews (2015) which is provided in the package GPFlow (De G. Matthews et al., 2017) and trained by the natural

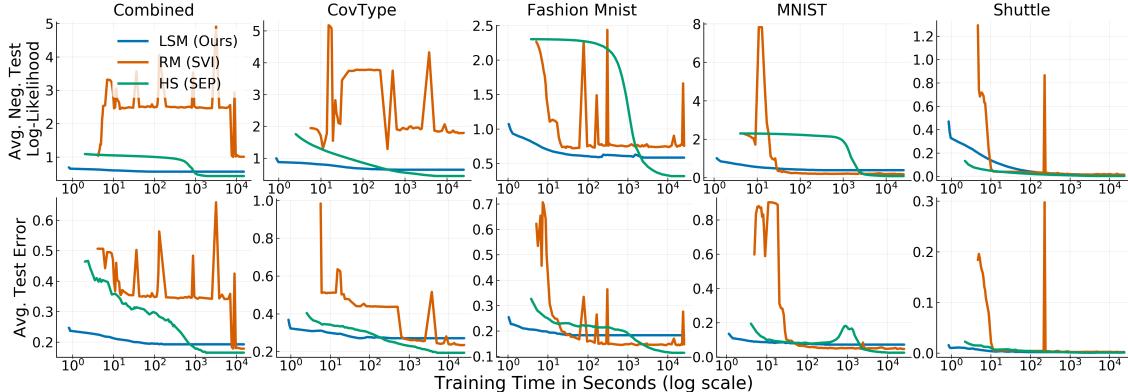


Figure 7: *Numerical comparison:* Prediction error and negative log-likelihood as a function of training time (seconds on a \log_{10} scale). Our method (LSM) converges one to two orders of magnitudes faster than the Heaviside model (HS) and is around 10 times faster than the robust-max model (RM). RM yields poor negative log-likelihood values due to poor uncertainty calibration.

gradient method of Salimbeni et al. (2018) and second, the Heaviside likelihood model (HS) trained by a scalable EP method (Villacampa-Calvo and Hernández-Lobato, 2017). For all methods, the hyperparameters are initialized to the same values, and are optimized using Adam. We compare the methods on five different multi-class benchmark datasets: Combined (98,528 points, 50 features, 3 classes), CovType (581,000 points, 54 features, 7 classes), Fashion-MNIST (70,000 points, 784 features, 10 classes), MNIST (70,000 points, 784 features, 10 classes) and Shuttle (58,000 points, 9 features, 7 classes).

In Figure 7 we plot the test error and negative log-likelihood as functions of the training time for each dataset. We find that our method (LSM) is one to two orders of magnitude faster than the EP based method for the Heaviside model (HS) and around ten times faster than the SVI based method for the robust-max model (RM).

Furthermore, our method consistently beats RM in terms of negative log-likelihood due to the better calibrated uncertainty quantification. Only on the MNIST dataset RM reaches a slightly better log-likelihood. This dataset is easily separable and therefore, suits well to the robust-max likelihood assumptions. On most datasets, the EP based method (HS) leads to slightly better predictive log-likelihood values, but is demanding a much longer training time. In contrast to the log-likelihood, the pure prediction error is not very sensitive to uncertainty calibration. All three methods achieve similar prediction errors whereby HS is a bit better on some datasets.

Moreover, the optimization curves in Figure 7 show that our inference method is much more stable than the SVI approach for the RM model. This is due to our efficient coordinate ascent updates which are given in closed-form. The RM approach suffers from additional noise injected

by approximating its gradients.

To summarize, our method is a good choice for fast inference on big datasets. It is particularly well fitted for datasets with overlapping classes where well calibrated uncertainty quantification is important. Due to the closed-form updates our method is more stable than the competitors.

6 CONCLUSION

We proposed an efficient Gaussian process multi-class classification method that builds on data augmentation. The augmented model is conditionally conjugate allowing for fast and stable variational inference based on closed-form updates. The experiments show that our approach leads to better confidence calibration than recent scalable multi-class GP classification methods. Additionally, we achieve competitive prediction performance while being faster than state-of-the-art. For small problems the proposed Gibbs sampler can be used which provides samples from the exact posterior.

The presented work shows how data augmentation can speed up inference in GP based models. Our approach may pave the way to similar augmentation strategies for other Bayesian models. Future work may aim at extending our approach to Bayesian neural networks (BNNs). Inference in BNNs is a hard problem. Exchanging the common softmax link functions with our proposed logistic-softmax may lead to a conditionally conjugate augmentation approach for BNNs. Typically, Gaussian priors are used for the weights of the network. In the augmented model the posterior of the weights would be given in closed-form. This might lead to an efficient inference algorithm.

Acknowledgements

We thank Stephan Mandt, Robert Bamler and Marius Kloft for discussions and feedback on the manuscript. We also thank Simon Danisch for helping with implementation details in Julia. 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, 01IS18051A.

Bibliography

- Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679.
- Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.
- Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*.
- Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. (2016). End to end learning for self-driving cars. *CoRR*, abs/1604.07316.
- Buchholz, A., Wenzel, F., and Mandt, S. (2018). Quasi-monte carlo variational inference. In *International Conference on Machine Learning*, pages 667–676.
- Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. (2015). Intelligible models for health-care: Predicting pneumonia risk and hospital 30-day readmission. In *KDD*, pages 1721–1730. ACM.
- Češnovar, R. and Štrumbelj, E. (2017). Bayesian lasso and multinomial logistic regression on gpu. *PLOS ONE*, 12(6):1–17.
- Chai, K. M. A. (2012). Variational multinomial logit gaussian process. *Journal of Machine Learning Research*, 13:1745–1808.
- De G. Matthews, A. 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. *J. Mach. Learn. Res.*, 18(1):1299–1304.
- Donner, C. and Opper, M. (2017). The inverse Ising problem in continuous time: A latent variable approach. *Physical Review E*, 96(6):062104.
- Donner, C. and Opper, M. (2018). Efficient Bayesian Inference for a Gaussian Process Density Model. *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 1–10.
- Girolami, M. and Rogers, S. (2006). Variational bayesian multinomial probit regression with gaussian process priors. *Neural Computation*, 18(8):1790–1817.
- Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.
- Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence*.
- Hensman, J. and Matthews, A. (2015). Scalable Variational Gaussian Process Classification. *AISTATS*.
- Hensman, J., Matthews, A., Filippone, M., and Ghahramani, Z. (2015). MCMC for variationally sparse gaussian processes. *NIPS*.
- Hernández-Lobato, D., Hernández-Lobato, J. M., and Dupont, P. (2011). Robust multi-class gaussian process classification. In *Advances in neural information processing systems*, pages 280–288.
- Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic Variational Inference. *JMLR*.
- Izmailov, P., Novikov, A., and Kropotov, D. (2018). Scalable gaussian processes with billions of inducing inputs via tensor train decomposition. *AISTATS*.
- Kim, H.-C. and Ghahramani, Z. (2006). Bayesian gaussian process classification with the em-ep algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(12):1948–1959.
- Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations*.
- Linderman, S. W., Johnson, M. J., and Adams, R. P. (2015). Dependent multinomial models made easy: Stick-breaking with the polya-gamma augmentation. *NIPS*.
- Mandt, S., Hoffman, M., and Blei, D. (2016). A Variational Analysis of Stochastic Gradient Algorithms. *ICML*.
- Maritz, J. and Lwin, T. (1989). Empirical Bayes Methods with Applications. *Monographs on Statistics and Applied Probability*.
- Naeini, M. P., Cooper, G., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian

- binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- Owen, A. (1998). Monte Carlo extension of quasi-Monte Carlo. *1998 Winter Simulation Conference. Proceedings (Cat. No.98CH36274)*, 1(1):571–577.
- 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.
- Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.
- Riihimäki, J., Jylänki, P., and Vehtari, A. (2013). Nested expectation propagation for gaussian process classification. *J. Mach. Learn. Res.*, 14(1):75–109.
- Ruiz, F. J. R., Titsias, M. K., Dieng, A. B., and Blei, D. M. (2018). Augment and reduce: Stochastic inference for large categorical distributions. *ICML*.
- Salimbeni, H., Eleftheriadis, S., and Hensman, J. (2018). Natural gradients in practice: Non-conjugate variational inference in gaussian process models. *AISTATS*.
- Titsias, M. (2016). One-vs-each approximation to softmax for scalable estimation of probabilities. In *Advances in Neural Information Processing Systems*, pages 4161–4169.
- Titsias, M. K. (2009). Variational learning of inducing variables in sparse gaussian processes. In *In Artificial Intelligence and Statistics 12*, pages 567–574.
- Villacampa-Calvo, C. and Hernández-Lobato, D. (2017). Scalable multi-class gaussian process classification using expectation propagation. *ICML*.
- Walker, S. G. (2011). Posterior sampling when the normalizing constant is unknown. *Communications in Statistics–Simulation and Computation®*, 40(5):784–792.
- Wenzel, F., Galy-Fajou, T., Donner, C., Kloft, M., and Opfer, M. (2019). Efficient gaussian process classification using polya-gamma data augmentation. *AAAI*.
- Williams, C. K. I. and Barber, D. (1998). Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1342–1351.
- Xiong, H., Wu, J., and Liu, L. (2010). Classification with classoverlapping: A systematic study. In *Proceedings of the 1st International Conference on E-Business Intelligence (ICEBI2010)*,. Atlantis Press.
- Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., and Zweig, G. (2016). Achieving human parity in conversational speech recognition. *CoRR*, abs/1610.05256.

A APPENDIX

A.1 Reparametrization of the Pólya-Gamma variables

By applying the augmentation of the sigmoid (8) to the augmented likelihood (7), we obtain the Pólya-Gamma augmented likelihood

$$p(y_i = k | \mathbf{f}_i, \lambda_i, \mathbf{n}_i, \tilde{\omega}_i, \boldsymbol{\omega}_i) = \frac{1}{2} \exp\left(\frac{f_i^k}{2} - \frac{(f_i^k)^2}{2}\tilde{\omega}_i\right) \times \prod_{c=1}^C 2^{-n_i^c} \exp\left(-\frac{n_i^c f_i^c}{2} - \frac{(f_i^c)^2}{2}\omega_i^c\right), \quad (10)$$

where we impose the prior distributions

$$\begin{aligned} p(\tilde{\omega}_i) &= \text{PG}(1, 0) \\ p(\boldsymbol{\omega}_i | \mathbf{n}_i) &= \prod_c \text{PG}(\omega_i^c | n_i^c, 0). \end{aligned}$$

We simplify this expression by combining all terms corresponding to the index k . To this end, we use a one hot-encoding of $\mathbf{y} \in \{0, \dots, C\}^N$ as $\mathbf{y}' \in \{0, 1\}^{C \times N}$,

$$y'_i^c = \begin{cases} 1 & \text{for } y_i = c \\ 0 & \text{otherwise.} \end{cases}.$$

Building on the identity $\omega_1 + \omega_2 = \omega_3$ with $\omega_1 \sim \text{PG}(b_1, c)$, $\omega_2 \sim \text{PG}(b_2, c)$ and $\omega_3 \sim \text{PG}(b_1 + b_2, c)$, we rewrite equation (10) as

$$p(y_i = k | \mathbf{f}_i, \lambda_i, \mathbf{n}_i, \boldsymbol{\omega}_i) = \prod_{c=1}^C 2^{-(y'_i^c + n_i^c)} \exp\left(\frac{(y'_i^c - n_i^c)f_i^c}{2} - \frac{(f_i^c)^2}{2}\omega_i^c\right),$$

where the terms corresponding to $\tilde{\omega}$ are now absorbed into the terms corresponding to $\boldsymbol{\omega}$.

A.2 Block coordinate ascent (CAVI) updates

The variational distribution is $q(\mathbf{u}, \boldsymbol{\lambda}, \mathbf{n}, \boldsymbol{\omega}) = q(\mathbf{u})q(\boldsymbol{\lambda})q(\boldsymbol{\omega}, \mathbf{n})$ and the factors are

$$\begin{aligned} q(\mathbf{u}) &= \prod_c \mathcal{N}(\mathbf{u}^c | \boldsymbol{\mu}^c, \Sigma^c), \quad q(\boldsymbol{\lambda}) = \prod_i \text{Ga}(\lambda_i | \alpha_i, \beta_i), \\ q(\boldsymbol{\omega}, \mathbf{n}) &= \prod_{i,c} \text{PG}(\omega_i^c | y'_i^c + n_i^c, b_i^c) \text{Po}(n_i^c | \gamma_i^c). \end{aligned}$$

In the CAVI scheme (Hoffman et al., 2013) each factor is iteratively updated by the following equation. Suppose we want to update the variational distribution corresponding

to the latent variable $\boldsymbol{\theta} \in \{\mathbf{u}, \boldsymbol{\lambda}, \mathbf{n}, \boldsymbol{\omega}\}$. Let $\bar{\boldsymbol{\theta}}$ be the set of the other latent variables, then $q^*(\boldsymbol{\theta})$ is updated by

$$q^*(\boldsymbol{\theta}) \propto \exp\left(\mathbb{E}_{q(\bar{\boldsymbol{\theta}})} [\log p(\boldsymbol{\theta} | \bar{\boldsymbol{\theta}})]\right). \quad (11)$$

Using this equation gives the closed-form update for each variational parameter.

$$\begin{aligned} \bar{f}_i^c &= \sqrt{\mathbb{E}_{q(f^c)} [(f_i^c)^2]} \\ &= \sqrt{\tilde{K}_{ii}^c + \kappa_i^c \Sigma^c \kappa_i^c \top + (\kappa_i^c \boldsymbol{\mu}^c) \top \kappa_i^c \boldsymbol{\mu}^c} \\ \gamma_i^c &= \frac{\exp(\psi(\alpha_i)) \exp\left(-\frac{\kappa_i^c \boldsymbol{\mu}^c}{2}\right)}{\beta_i \cosh\left(\frac{\bar{f}_i^c}{2}\right)} \end{aligned} \quad (12)$$

$$\alpha_i = 1 + \sum_{c=1}^C \gamma_i^c, \quad \beta_i = C \quad (13)$$

$$b_i^c = \bar{f}_i^c, \quad (14)$$

$$\theta_i^c = \mathbb{E}_{q(\omega_i^c, n_i^c)} [\omega_i^c] = \frac{y'_i^c + \gamma_i^c}{2b_i^c} \tanh \frac{b_i^c}{2} \quad (15)$$

$$\Sigma^c = \left(\kappa^c \top \text{diag}(\boldsymbol{\theta}^c) \kappa^c + (K_{mm}^c)^{-1} \right)^{-1}, \quad (16)$$

where $\psi(\cdot)$ is the digamma function. When $\kappa\mu \ll 0$, equation (12) easily overflows. One can solve this problem by approximating $\exp(-0.5\kappa\mu)/\cosh(0.5\bar{f})$ with $\sigma(\kappa\mu)$ by neglecting the variance terms $\tilde{K} + \kappa\Sigma\kappa^\top$ in \bar{f} .

Equation (12) and (13) shows a direct interdependence between α_i and γ_i^c . We use inner loop of alternating between updating both variables until convergence to solve the problem. We find that 5 iterations in the inner loop are enough.

Finally, if class subsampling (the extreme classification version of our algorithm Alg. 2) is used, α_i is approximated by

$$\alpha_i = 1 + \frac{C}{|\mathcal{K}|} \sum_{c \in \mathcal{K}} \gamma_i^c, \quad (17)$$

where C is the number of classes and $|\mathcal{K}|$ is the number of sub-sampled classes.

A.3 Subsampling the classes (extreme classification version)

The extreme classification version of our algorithm is presented in Alg. 2. In each iteration we only consider a minibatch of the classes $\mathcal{B} \subset \{1, \dots, C\}$ and the variational parameters b_i^c , α_i^c , μ^c , Σ^c (lines 13, 11, 18, 19 in Alg. 1) are only updated for $i \in \mathcal{B}$. The updates that are global w.r.t. the classes, i.e. λ_i and the hyperparameters

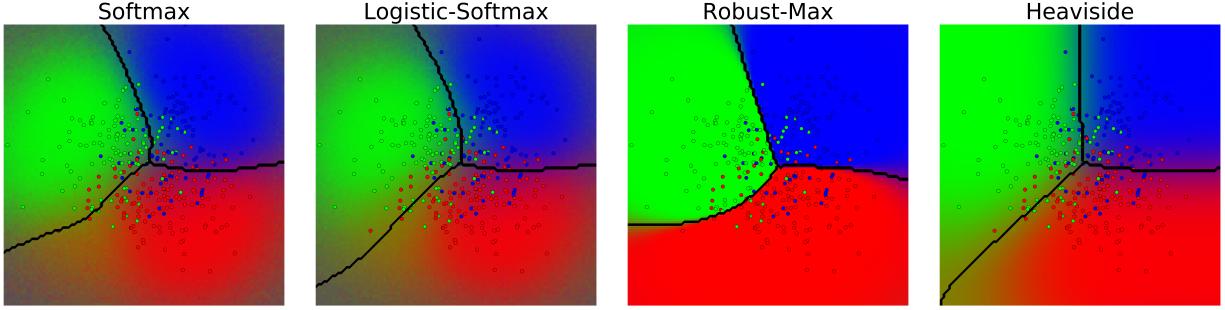


Figure 8: RGB representation of the predictive likelihood for a toy dataset as described in section 5.1 with variance $\sigma^2 = 0.5$. Each class is attributed a color channel (Red, Green, Blue) and predictive likelihoods are mapped into RGB values.

h (lines 11, 22) are now replaced by stochastic gradient updates.

Algorithm 2 Conjugate multi-class Gaussian process classification with class subsampling

```

1: Input: data  $X, y$ , minibatch size  $|\mathcal{S}|$  and  $|\mathcal{B}|$ 
2: Output: variational posterior GPs  $p(u^c|\mu^c, \Sigma^c)$ 
3: Set the learning rate schedules  $\rho_t, \rho_t^h$  appropriately
4: Initialize all variational parameters and hyperparameters
5: Select  $M$  inducing points locations (e.g. kMeans)
6: for iteration  $t = 1, 2, \dots$  do
7:   # Sample minibatch:
8:   Sample a minibatch of the data  $\mathcal{S} \subset \{1, \dots, N\}$ 
9:   Sample a set of labels  $\mathcal{K} \subset \{1, \dots, C\}$ 
10:  # Local variational updates
11:  for  $i \in \mathcal{S}$  do
12:    Update  $(\alpha_i, \gamma_i^c)_{c \in \mathcal{K}}$  (Eq. 12,17)
13:    for  $c \in \mathcal{K}$  do
14:      Update  $b_i^c$  (Eq. 14)
15:    end for
16:  end for
17:  # Global variational GP updates
18:  for  $c \in \mathcal{K}$  do
19:     $\mu^c \leftarrow (1 - \rho_t)\mu^c + \rho_t \hat{\mu}^c$  (Eq. 15)
20:     $\Sigma^c \leftarrow (1 - \rho_t)\Sigma^c + \rho_t \hat{\Sigma}^c$  (Eq. 16)
21:  end for
22:  # Hyperparameter updates
23:  Gradient step  $h \leftarrow h + \rho_t^h \nabla_h \mathcal{L}$ 
24: end for

```

A.4 Visualization of the different likelihoods

To get a better intuition of the behavior of each likelihood, we visualize the prediction function of each method as a contour plot using the toy dataset from section 5.1. To visualize the predictive likelihood, we map the predictive values of each class to a RGB color channel (where each class corresponds to one color and mixing of colors indicates a contribution of multiple classes). A highly saturated color corresponds to a high confidence in the class prediction, while mixed colors indicate zones of transition between classes and lower confidence. The results

are shown in Figure 8 for a toy dataset consisting of 500 points generated from a mixture of Gaussians with variance $\sigma^2 = 0.5$. As expected, the robust-max likelihood leads to extremely sharp decision boundaries and high confidences for all regions (even for the overlapping regions). The other likelihoods lead to better calibration resulting in soft boundaries and less confident predictions in the overlapping regions.

A.5 Convexity of the negative ELBO

In the following we prove that the negative ELBO ($-\mathcal{L}$) of our augmented model is convex in the global variational parameters μ^c and Σ^c . To prove this statement, we write the negative ELBO in terms of μ^c and Σ^c ,

$$-\mathcal{L}(\mu^c, \Sigma^c) = \frac{1}{2} \left[\sum_{i=1}^N (y'_i - \gamma_i^c) \mu_i^c - \theta_i^c ((\mu_i^c)^2 + \Sigma_{ii}^c) \right] \\ + \frac{1}{2} \left[\mu^{c\top} K^{-1} \mu^c + \text{tr}(K^{-1} \Sigma^c) - \log |\Sigma^c| \right].$$

Differentiating twice in μ^c gives $\text{diag}(\theta^c) + K^{-1}$ which is positive definite since $\theta_i^c > 0$ for all i and by definition of K . Therefore, the negative ELBO is convex in μ^c for all c .

Differentiating twice in Σ^c gives $(\Sigma^c)^{-1} \otimes (\Sigma^c)^{-1}$, where \otimes is the Kroenecker product. This is again positive definite since $(\Sigma^c)^{-1}$ is positive definite and the Kroenecker product preserves positive definiteness. Therefore, the negative ELBO is also convex in Σ^c for all c .