SCALABLE HIERARCHICAL MIXTURE OF GAUSSIAN PROCESSES FOR PATTERN CLASSIFICATION

T. N. A. Nguyen[⋆] *A. Bouzerdoum*[⋆†] *S. L. Phung*[⋆]

[⋆] School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Australia
[†] College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar

ABSTRACT

This paper introduces a novel Gaussian process (GP) classification method that combines advantages of global and local GP approximators through a two-layer hierarchical model. The upper layer consists of a global sparse GP to coarsely model the entire dataset. The lower layer is a mixture of GP experts which uses local information to learn a fine-grained model. A variational inference algorithm is developed for simultaneous learning of the global GP, the experts and the gating network. Stochastic optimization can be employed for large-scale problems. Experiments on benchmark binary classification datasets demonstrate the advantages of the method in terms of scalability and classification accuracy.

Index Terms— Gaussian processes, variational inference, pattern classification

1. INTRODUCTION

Gaussian processes are powerful tools for Bayesian regression and classification. Model selection for GPs is realized by maximizing the marginal likelihood, and inference is performed by calculating the posterior of latent variables. In GP regression, closed-form solutions can be obtained. In classification, due to the non-Gaussian likelihood, we must resort to approximate inference methods to estimate the marginal likelihood and posterior. For more details on the approximation methods, the reader is referred to the reviews in [1] and [2].

The main limitation of GP is its high computational cost, mainly due to the inversion and storage of the kernel matrix. In regression setting, many sparse approximation methods have been proposed to overcome this limitation; a review can be found in [3]. Common to these methods is the approximation of training data with a small set of inducing points. In FITC [4], inducing points are optimized against the approximate marginal likelihood. In [5], inducing points are found by minimizing a variational lower bound of the marginal likelihood. This method has been shown to produce better placement of inducing points than FITC. In [6], Hensman et al. reformulated the variational bound derived in [5] to enable stochastic optimization, allowing the application to problems with millions of samples.

Recently, there has been much interest in sparse GP approximation for classification [7, 8, 9]. The generalized FITC presented in [8] combines the sparse approximation prior derived in FITC with a Bernoulli likelihood and uses expectation propagation (EP) to approximate the posterior. Like FITC, it leads to suboptimal placement of the inducing points. In addition, there is no systematic way to apply stochastic optimization to further reduce the computational cost for this method. It is therefore only limited to problems with a few thousand data samples. Recently, in [9], Hensman et al. proposed a variational sparse GP classifier which optimizes a generalized form of the objective function derived in [6]. The classifier inherits the two desirable properties of [6]: the ability to place inducing points optimally and the feasibility of using stochastic optimization.

The sparse approximation methods normally work well for simple datasets. However, a single GP accompanied by a small set of global inducing points cannot account for the non-stationarity and locality in large and complex datasets, as argued in [10]. To overcome this limitation, we turn to a structure called mixture of GPs [10, 11, 12, 13, 14]. In a mixture of GPs, a gating network divides the input space into regions within which a specific GP expert is responsible for making predictions. In this way, non-stationarity and locality in the data can be naturally addressed. The main limitation of mixtures of GPs is that each expert is independently trained using only the local data assigned to it, without taking into account the global information, i.e. the correlation between the clusters. The trained experts are therefore likely to overfit the local data. The second limitation of mixtures of GPs is due to the complexity of the inference problem, which usually involves simultaneous learning of the experts and the gating network. Therefore, approximation techniques are often required. Many existing mixtures of GPs, such as those in [10, 15, 12], resort to the intensive MCMC sampling, which can be very slow. Recently, in [13, 16, 14], variational inference has been used as a more flexible and faster alternative to MCMC sampling for mixtures of GP experts in regression setting. However, it is not trivial to adapt these variational mixtures of GPs to classification setting. To the best of our knowledge, there are still no publicly available methods using variational mixtures of GPs for classification.

In this paper, we propose a GP approximation method for classification that combines the advantages of sparse approximation and mixture of GPs to exploit both the global and local information from the data. Our model has a two-layer hierarchical structure. In the upper layer, a sparse GP accompanied by a set of global inducing points is used to coarsely model the whole dataset. The lower layer comprises multiple GP experts, each of which makes use of the local information for fine-grained modeling. These experts share a common prior mean function modeled by the upper layer to avoid overfitting. Inference in our model involves simultaneous learning of the global GP, the experts and the gating network. For this, we develop a two-step variational inference algorithm and adopt the idea of [9] to enable stochastic optimization in large-scale problems.

The remainder of the paper is organized as follows. Section 2 introduces the theoretical background of GP classification. Section 3 presents the proposed model, and Section 4 describes the variational inference approach for the model. Section 5 presents the experiments and results. Finally, Section 6 concludes the paper.

2. BACKGROUND

We consider a *binary classification problem* where a training set $\mathcal{D}$ consists of input data, $\mathbf{X}=(\mathbf{x}_1^\intercal, ..., \mathbf{x}_N^\intercal)^\intercal$ with input points (row vectors) $\mathbf{x}_n \in \mathcal{X} \subset \mathbb{R}^D$, and class observations $\mathbf{y}=(y_1, ..., y_N)^\intercal$, the

task is to compute the output y^* at a new test location $\mathbf{x}^*$. We assume that there is an underlying latent function $f(\mathbf{x}) : \mathcal{X} \longmapsto \mathbb{R}$ that is distributed according to a GP, which is characterized by a mean function $m(\mathbf{x})$ and a covariance function $\kappa(\mathbf{x}, \mathbf{x}')$: $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'))$. The GP places a prior on the latent variables:

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{m_X}, \mathbf{K_{XX}}), \tag{1}$$

where $\mathbf{f} = [f_1, ..., f_N]^{\mathrm{T}}$ with $f_n \equiv f(\mathbf{x}_n)$, $\mathbf{m_X} = [m(\mathbf{x}_1), ..., m(\mathbf{x}_N)]^{\mathrm{T}}$ and $\mathbf{K_{XX}}$ denotes the covariance matrix formed by evaluating $\kappa(\mathbf{x}, \mathbf{x}')$ at all pairs of input vectors. The observed outputs are then related to the latent variables according to the probit likelihood:

$$p(y_n|f_n) = \mathcal{B}(y_n|\phi(f_n)) = \phi(f_n)^{y_n}(1 - \phi(f_n))^{1-y_n}, \tag{2}$$

where $\mathcal{B}$ denotes Bernoulli distribution and $\phi(z) = \int_{-\infty}^{z} \mathcal{N}(x|0, 1)dx$.

The main object of interest is the posterior over latent variables

$$p(\mathbf{f}|\mathbf{y}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f})/p(\mathbf{y}) = \prod_{n=1}^{N} p(y_n|f_n)p(\mathbf{f})/p(\mathbf{y}).$$

Since the likelihood $p(y_n|f_n)$ is non-Gaussian, $p(\mathbf{f}|\mathbf{y})$ is not tractable and must be approximated. The marginal likelihood $p(\mathbf{y})$ must also be approximated and then minimized to find the optimal parameters for $\kappa(\mathbf{x}, \mathbf{x}')$. See [2] for a review of an assortment of approximation methods. These methods require $O(N^3)$ in computation.

Given the posterior $p(\mathbf{f}|\mathbf{y})$, prediction can be made by first computing the distribution of the latent variable f^* at the test point $\mathbf{x}^*$: $p(f^*|\mathbf{y}) = \int p(f^*|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}$. Subsequently, marginalizing f^* gives a probabilistic prediction: $p(y^*|\mathbf{y}) = \int p(y^*|f^*)p(f^*|\mathbf{y})df^*$.

The complexity of $O(N^3)$ is prohibitive for large datasets. To reduce the computational cost, many sparse GP approximation schemes have been proposed [7, 8, 9]. In these schemes, the latent variables $\mathbf{f}$ are summarized by a set of inducing points consisting of inducing inputs $\mathbf{Z}$ and their corresponding latent variables $\mathbf{g}$. The inducing inputs $\mathbf{Z}$ are points in the input space $\mathcal{X}$, and the inducing variables $\mathbf{g}$ are points on the same latent function as $\mathbf{f}$. Using global inducing points, a sparse approximation normally cannot deal with non-stationarity and locality in complex datasets. Next, we introduce a method to overcome this limitation.

3. HIERARCHICAL MIXTURE OF GP EXPERTS FOR CLASSIFICATION

Here we develop a GP classification model that makes use of both global and local information in the dataset through a two-layer hierarchical structure. In the upper layer, a sparse GP, hereinafter referred to as the *global* GP, is used to coarsely model the entire dataset. In the lower layer, a gating network divides the input space into regions; and within each region, a specific local GP, hereinafter referred to as the *expert*, is used for finer modeling. The graphical representation of the model is shown in Fig. 1.

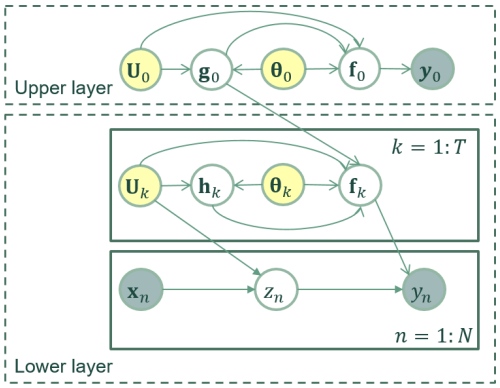

Fig. 1: Graphical representation of the hierarchical mixture of GP experts model for classification. Observation $\mathbf{y}_0$ is a duplicate of $\mathbf{y}$.

The global GP in the upper layer is associated with a latent function $f_0(\mathbf{x})$, a zero mean function and a covariance function $\kappa_0(\mathbf{x}, \mathbf{x}')$: $f_0(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, \kappa_0(\mathbf{x}, \mathbf{x}'))$. Let T be the number of local experts in the lower layer. Each expert is associated with a latent function $f_k(\mathbf{x})$, a mean function $m(\mathbf{x})$ and a covariance function $\kappa_k(\mathbf{x}, \mathbf{x}')$: $f_k(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), \kappa_k(\mathbf{x}, \mathbf{x}'))$, for $k = 1, ..., T$. Each global or local GP is sparsely represented with a set of augmented inducing points. Let $\mathbf{f}_k$, $\mathbf{g}_k$, $\mathbf{U}_k$, $\boldsymbol{\theta}_k$ and $\mathbf{K}^{(k)}$, respectively, denote the training latent variables, inducing variables, inducing inputs, hyperparameters of covariance function and covariance matrices for the k-th GP ($k = 0, ..., T$). $\mathbf{K}^{(k)}_{\mathbf{AB}}$ is formed by evaluating the function $\kappa_k(\mathbf{x}, \mathbf{x}'))$ at all pairs of points $(\mathbf{x}, \mathbf{x}')$ with $\mathbf{x}$ in $\mathbf{A}$ and $\mathbf{x}'$ in $\mathbf{B}$. The global GP places a joint prior distribution on the latent variables:

$$p\left(\begin{bmatrix} \mathbf{g}_0 \\ \mathbf{f}_0 \end{bmatrix}\right) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}^{(0)}_{\mathbf{U}_0\mathbf{U}_0} & \mathbf{K}^{(0)}_{\mathbf{U}_0\mathbf{X}} \\ \mathbf{K}^{(0)}_{\mathbf{X}\mathbf{U}_0} & \mathbf{K}^{(0)}_{\mathbf{XX}} \end{bmatrix}\right). \tag{3}$$

Applying the Gaussian identities presented in Section A.2 of [17], the marginal $p(\mathbf{g}_0)$ and conditional $p(\mathbf{f}_0|\mathbf{g}_0)$ are then given by

$$p(\mathbf{g}_0) = \mathcal{N}(\mathbf{0}, \mathbf{K}^{(0)}_{\mathbf{U}_0\mathbf{U}_0}), \tag{4}$$

$$p(\mathbf{f}_0|\mathbf{g}_0) = \mathcal{N}\big(\mathbf{K}^{(0)}_{\mathbf{X}\mathbf{U}_0}[\mathbf{K}^{(0)}_{\mathbf{U}_0\mathbf{U}_0}]^{-1}\mathbf{g}_0, \mathbf{K}^{(0)}_{\mathbf{XX}} - \mathbf{K}^{(0)}_{\mathbf{X}\mathbf{U}_0}[\mathbf{K}^{(0)}_{\mathbf{U}_0\mathbf{U}_0}]^{-1}\mathbf{K}^{(0)}_{\mathbf{U}_0\mathbf{X}}\big). \tag{5}$$

To enforce correlation among the local experts, all the local sparse GPs share a prior mean function $m(\mathbf{x})$, which encodes global information from the upper layer. We set $m(\mathbf{x})$ to be the mean of the conditional $p(f_0(\mathbf{x})|\mathbf{g}_0)$ given in Eq. (5): $m(\mathbf{x}) = \mathbf{K}^{(0)}_{\mathbf{X}\mathbf{U}_0}[\mathbf{K}^{(0)}_{\mathbf{U}_0\mathbf{U}_0}]^{-1}\mathbf{g}_0$. Conditioning on $\mathbf{g}_0$ for the mean function, each local GP places a joint distribution on its latent variables:

$$p\left(\begin{bmatrix} \mathbf{g}_k \\ \mathbf{f}_k \end{bmatrix} \middle| \mathbf{g}_0\right) = \mathcal{N}\left(\begin{bmatrix} m(\mathbf{U}_k) \\ m(\mathbf{X}) \end{bmatrix}, \begin{bmatrix} \mathbf{K}^{(k)}_{\mathbf{U}_k\mathbf{U}_k} & \mathbf{K}^{(k)}_{\mathbf{U}_k\mathbf{X}} \\ \mathbf{K}^{(k)}_{\mathbf{X}\mathbf{U}_k} & \mathbf{K}^{(k)}_{\mathbf{XX}} \end{bmatrix}\right). \tag{6}$$

Applying the aforementioned Gaussian identities again results in

$$p(\mathbf{g}_k|\mathbf{g}_0) = \mathcal{N}(m(\mathbf{U}_k), \mathbf{K}^{(k)}_{\mathbf{U}_k\mathbf{U}_k}), \tag{7}$$

$$p(\mathbf{f}_k|\mathbf{g}_k, \mathbf{g}_0) = \mathcal{N}\big(\mathbf{K}^{(k)}_{\mathbf{X}\mathbf{U}_k}[\mathbf{K}^{(k)}_{\mathbf{U}_k\mathbf{U}_k}]^{-1}(\mathbf{g}_k - m(\mathbf{U}_k)) + m(\mathbf{X}), \mathbf{K}^{(k)}_{\mathbf{XX}} - \mathbf{K}^{(k)}_{\mathbf{X}\mathbf{U}_k}[\mathbf{K}^{(k)}_{\mathbf{U}_k\mathbf{U}_k}]^{-1}\mathbf{K}^{(k)}_{\mathbf{U}_k\mathbf{X}}\big). \tag{8}$$

For simplicity, we introduce new latent variables $\mathbf{h}_k = \mathbf{g}_k - m(\mathbf{U}_k)$ to substitute for $\mathbf{g}_k$. As a result, Eqs. (7) and (8) become:

$$p(\mathbf{h}_k|\mathbf{g}_0) = \mathcal{N}(\mathbf{0}, \mathbf{K}^{(k)}_{\mathbf{U}_k\mathbf{U}_k}), \tag{9}$$

$$p(\mathbf{f}_k|\mathbf{h}_k, \mathbf{g}_0) = \mathcal{N}\big(\mathbf{K}^{(k)}_{\mathbf{X}\mathbf{U}_k}[\mathbf{K}^{(k)}_{\mathbf{U}_k\mathbf{U}_k}]^{-1}\mathbf{h}_k + \mathbf{K}_{\mathbf{X}\mathbf{U}_0}\mathbf{K}^{-1}_{\mathbf{U}_0\mathbf{U}_0}\mathbf{g}_0, \mathbf{K}^{(k)}_{\mathbf{XX}} - \mathbf{K}^{(k)}_{\mathbf{X}\mathbf{U}_k}[\mathbf{K}^{(k)}_{\mathbf{U}_k\mathbf{U}_k}]^{-1}\mathbf{K}^{(k)}_{\mathbf{U}_k\mathbf{X}}\big). \tag{10}$$

Let $\mathbf{y}_0$ and $\mathbf{y}$ denote the training outputs of the upper and lower layers, respectively: $\mathbf{y}_0$ is a duplicate of $\mathbf{y}$. $\mathbf{f}_0$ and $\mathbf{y}_0$ are related by a probit likelihood given in Eq. (2). In the lower layer, for each observation $(\mathbf{x}_n, y_n)$, a latent variable z_n indicates the expert to which the observation belongs. The likelihood for the outputs $\mathbf{y}$ is also probit:

$$p(y_n|f_1(\mathbf{x}_n), ..., f_T(\mathbf{x}_n)) = \prod_{k=1}^{T} p(y_n|f_k(\mathbf{x}_n))^{[z_n==k]} = \mathcal{B}(y_n|\phi(f_{z_n}(\mathbf{x}_n))).$$

Expert indicators are specified by a gating network based on the inputs. Since the target here is large-scale problems, the simple gating network suggested in [14] is employed for fast expert allocation. In this gating network, the prior over z_n is defined as

$$p(z_n = k) = \frac{\mathcal{N}(\mathbf{x}_n|\mathbf{m}_k, \mathbf{V})}{\sum_{j=1}^{T} \mathcal{N}(\mathbf{x}_n|\mathbf{m}_j, \mathbf{V})}, \tag{11}$$

where $\mathbf{m}_k = \frac{1}{M}\sum_{m=1}^{M} \mathbf{u}^{(k)}_m$ represents the centroid of expert k, and $\mathbf{V} = \mathrm{diag}(v_1, ..., v_D)$ with $v_d = \frac{1}{T(M-1)}\sum_{k=1}^{T}\sum_{m=1}^{M}(u^{(k)}_{md} - m_{kd})^2$. The prior (11) is based on the observation that the closer $\mathbf{x}_n$ to $\mathbf{m}_k$, the more similar it is to the inducing inputs $\mathbf{U}_k$ and the better its output can be predicted by expert k; hence, it is given a higher probability to be assigned to that expert.

4. INFERENCE

Let $\mathbf{f}$, $\mathbf{h}$, $\mathbf{U}$, $\boldsymbol{\theta}$ and $\mathbf{z}$ denote the sets of all variables $\mathbf{f}_k, \mathbf{h}_k, \mathbf{U}_k, \boldsymbol{\theta}_k$ and z_n, respectively, for $k{=}1, ..., T$ and $n{=}1, ..., N$. Inference for the model involves estimating the posterior distribution of the latent variables $p(\mathbf{f}, \mathbf{f}_0, \mathbf{h}, \mathbf{g}_0, \mathbf{z}|\mathbf{y}, \mathbf{y}_0)$, and fixing the kernel hyperparameters and the inducing inputs. Our target is to use variational inference with possibility of applying stochastic optimization for large datasets. For this purpose, a set of global variables is required so that the model conditioned on these variables factorizes in the observations and latent variables; see Fig. 1 in [6] for an illustration of such models. The inducing variables $\mathbf{g}_0$ and $\mathbf{h}_k$, for $k{=}1, ..., T$, are well-suited for the role of global variables in our model. However, marginalizing these variables as in [5] eliminates the global parameters and re-introduces dependencies between the observations. Hence, we choose to represent the variational distributions of these variables explicitly as $q(\mathbf{g}_0)$ and $q(\mathbf{h}_k)$. It can be seen later that the variational distributions for $\mathbf{f}$ and $\mathbf{f}_0$ can be derived in terms of $q(\mathbf{h})$ and $q(\mathbf{g}_0)$. We then approximate the joint posterior distribution of $\mathbf{h}$, $\mathbf{g}_0$ and $\mathbf{z}$ by a factorized tractable variational distribution,

$$p(\mathbf{z}, \mathbf{h}, \mathbf{g}_0|\mathbf{y}, \mathbf{y}_0) \approx q(\mathbf{z}, \mathbf{h}, \mathbf{g}_0) = \prod_{n=1}^{N} q(z_n) q(\mathbf{g}_0) \prod_{k=1}^{T} q(\mathbf{h}_k). \quad (12)$$

A lower bound on the log marginal likelihood is first derived by applying the standard variational equation, $\ln p(\mathbf{y}, \mathbf{y}_0) \geq$

$$\begin{aligned}&\mathbb{E}_{q(\mathbf{z},\mathbf{h},\mathbf{g}_0)}[\ln p(\mathbf{y}\mathbf{y}_0|\mathbf{z}, \mathbf{h}, \mathbf{g}_0)] - \mathrm{KL}(q(\mathbf{z}, \mathbf{h}, \mathbf{g}_0)||p(\mathbf{z}, \mathbf{h}, \mathbf{g}_0))\\ &= \mathbb{E}_{q(\mathbf{z})q(\mathbf{g}_0)q(\mathbf{h})}[\ln p(\mathbf{y}|\mathbf{z}, \mathbf{h}, \mathbf{g}_0)] + \mathbb{E}_{q(\mathbf{g}_0)}[\ln p(\mathbf{y}_0|\mathbf{g}_0)]\\ &\quad -\mathrm{KL}(q(\mathbf{h})||p(\mathbf{h})) - \mathrm{KL}(q(\mathbf{g}_0)||p(\mathbf{g}_0)) - \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z})), \quad (13)\end{aligned}$$

where KL denotes Kullback-Leibler divergence. Applying Jensen's inequality to $p(\mathbf{y}|\mathbf{z}, \mathbf{h}, \mathbf{g}_0)$ and $p(\mathbf{y}_0|\mathbf{g}_0)$ yields

$$\ln p(\mathbf{y}|\mathbf{z}, \mathbf{h}, \mathbf{g}_0) \geq \mathbb{E}_{p(\mathbf{f}|\mathbf{h},\mathbf{g}_0)}[\ln p(\mathbf{y}|\mathbf{f}, \mathbf{z})], \quad (14)$$

$$\ln p(\mathbf{y}_0|\mathbf{g}_0) \geq \mathbb{E}_{p(\mathbf{f}_0|\mathbf{g}_0)}[\ln p(\mathbf{y}_0|\mathbf{f}_0)]. \quad (15)$$

This gives a further lower bound $\mathcal{L}$ on the log marginal likelihood:

$$\begin{aligned}&\mathcal{L} = \mathbb{E}_{q(\mathbf{z})}[\mathbb{E}_{q(\mathbf{f})}[\ln p(\mathbf{y}|\mathbf{f}, \mathbf{z})]] + \mathbb{E}_{q(\mathbf{f}_0)}[\ln p(\mathbf{y}_0|\mathbf{f}_0)]\\ &-\mathrm{KL}(q(\mathbf{h})||p(\mathbf{h})) - \mathrm{KL}(q(\mathbf{g}_0)||p(\mathbf{g}_0)) - \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z})), \quad (16)\end{aligned}$$

where $q(\mathbf{f}_0)$ and $q(\mathbf{f})$ are defined as: $q(\mathbf{f}_0) \triangleq \int p(\mathbf{f}_0|\mathbf{g}_0) q(\mathbf{g}_0) d\mathbf{g}_0$ and $q(\mathbf{f}) \triangleq \int p(\mathbf{f}|\mathbf{h}, \mathbf{g}_0) q(\mathbf{h}) q(\mathbf{g}_0) d\mathbf{h} d\mathbf{g}_0$.

It has been shown in [5] that the implicit optimal variational distribution $q(\mathbf{g}_0)$ to maximize the right hand side of Eq. (15) is Gaussian (see Eq. (10) in [5]). Similarly, the optimal distribution $q(\mathbf{h}, \mathbf{g}_0)$ to maximize the right hand side of Eq. (14), and hence the optimal $q(\mathbf{h}_k)$, is also Gaussian. We parametrize them as follows:

$$q(\mathbf{g}_0) \triangleq \mathcal{N}(\mathbf{m}_0, \mathbf{S}_0) \text{ and } q(\mathbf{h}_k) \triangleq \mathcal{N}(\mathbf{m}_k, \mathbf{S}_k). \quad (17)$$

Since $q(\mathbf{z})$ is assumed to factorize as in (12), the bound (16) becomes

$$\begin{aligned}\mathcal{L} =& \sum_{n=1}^{N} \sum_{k=1}^{T} q(z_n = k) \mathbb{E}_{q(f_k(\mathbf{x}_n))}[\ln p(y_n|f_k(\mathbf{x}_n))]\\ &+ \sum_{n=1}^{N} \mathbb{E}_{q(f_0(\mathbf{x}_n))}[\ln p(y_n|f_0(\mathbf{x}_n))] - \mathrm{KL}(q(\mathbf{h})||p(\mathbf{h}))\\ &- \mathrm{KL}(q(\mathbf{g}_0)||p(\mathbf{g}_0)) - \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z})). \quad (18)\end{aligned}$$

Only the marginals of $q(\mathbf{f})$ and $q(\mathbf{f}_0)$, i.e. $q(f_k(\mathbf{x}_n))$ for $k = 0, ..., T$ and $n{=}1, ..., N$, are needed to compute $\mathcal{L}$. With $q(\mathbf{g}_0)$, $q(\mathbf{h}_k)$, $p(\mathbf{f}_0|\mathbf{g}_0)$ and $p(\mathbf{f}_k|\mathbf{h}_k, \mathbf{g}_0)$ given in Eqs. (17), (5) and (10), it is straightforward to compute $q(f_0(\mathbf{x}_n))$ in terms of $\mathbf{m}_0$ and $\mathbf{S}_0$, and $q(f_k(\mathbf{x}_n))$ in terms of $\mathbf{m}_0, \mathbf{S}_0, \mathbf{m}_k$ and $\mathbf{S}_k$ for $k{=}1, ..., T$. Eq. (18) are left with only one-dimensional integrals of the log-likelihoods, which can be computed by numerical methods, such as Gauss-Hermite quadrature [18].

Inference is performed by maximizing the bound (18) with respect to (w.r.t.) the variational distributions $q(\mathbf{z})$, $q(\mathbf{h})$, $q(\mathbf{g}_0)$, the inducing inputs $\mathbf{U}$ and the kernel hyperparameters $\boldsymbol{\theta}$. To deal with the complex dependence between z and $\mathbf{U}$, we present an iterative optimization algorithm that alternates between the two following steps:

1. Fix $q(\mathbf{z})$ and maximize the bound w.r.t. the parameters of $q(\mathbf{h})$, $q(\mathbf{g}_0)$, $\mathbf{U}$ and $\boldsymbol{\theta}$ using gradient based optimization.
2. Fix $q(\mathbf{h})$, $q(\mathbf{g}_0)$, $\mathbf{U}$ and $\boldsymbol{\theta}$, and maximize the bound w.r.t. $q(\mathbf{z})$.

We now discuss each step in details. For the first step, the following equation contains the relevant terms of the bound to be maximized:

$$\begin{aligned}\mathcal{L}_1 =& \sum_{n=1}^{N} \sum_{k=1}^{T} q(z_n = k) \mathbb{E}_{q(f_k(\mathbf{x}_n))}[\ln p(y_n|f_k(\mathbf{x}_n))] \quad (19)\\ &+ \sum_{n=1}^{N} \mathbb{E}_{q(f_0(\mathbf{x}_n))}[\ln p(y_n|f_0(\mathbf{x}_n))] - \mathrm{KL}(q(\mathbf{h})||p(\mathbf{h})) - \mathrm{KL}(q(\mathbf{g}_0)||p(\mathbf{g}_0)).\end{aligned}$$

During optimization, to maintain positive-definiteness of the covariances $\mathbf{S}_k$, we represent them as $\mathbf{S}_k{=}\mathbf{L}_k\mathbf{L}_k^{\mathrm{T}}$, and perform unconstrained optimization w.r.t. $\mathbf{L}_k$. The difficult part for optimization is to find the derivatives of the intractable terms $\mathbb{E}_{q(f_k(\mathbf{x}_n))}[\ln p(y_n|f_k(\mathbf{x}_n))]$. As an intermediate step, we find their derivatives w.r.t. the means and variances of $q(f_k(\mathbf{x}_n))$ (denoted by μ_{nk} and σ^2_{nk}). To this end, the following Gaussian identities presented in [19] are used:

$$\begin{aligned}\frac{\partial}{\partial \mu}\mathbb{E}_{\mathcal{N}(x|\mu,\sigma^2)}[f(x)] &= \mathbb{E}_{\mathcal{N}(x|\mu,\sigma^2)}[\frac{\partial}{\partial x} f(x)]\\ \frac{\partial}{\partial \sigma^2}\mathbb{E}_{\mathcal{N}(x|\mu,\sigma^2)}[f(x)] &= \frac{1}{2}\mathbb{E}_{\mathcal{N}(x|\mu,\sigma^2)}[\frac{\partial^2}{\partial x^2} f(x)].\end{aligned} \quad (20)$$

By substituting f, μ and σ^2 in (20) with $\ln p(y_n|f_k(\mathbf{x}_n))$, μ_{nk} and σ^2_{nk}, we transform the derivatives of $\mathbb{E}_{q(f_k(\mathbf{x}_n))}[\ln p(y_n|f_k(\mathbf{x}_n))]$ w.r.t. μ_{nk} and σ^2_{nk} into one-dimensional integrals, which can be computed by quadrature methods. Finally, the derivatives w.r.t. $\mathbf{m}_k$, $\mathbf{L}_k$, $\mathbf{U}_k$ and θ_k can be calculated by applying straight-forward algebra.

In the second step, the relevant terms to be maximized are

$$\begin{aligned}\mathcal{L}_2 =& \mathbb{E}_{q(\mathbf{z})}\left\{\mathbb{E}_{q(\mathbf{f})}[\ln p(\mathbf{y}|\mathbf{f}, \mathbf{z})]\right\} - \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z})) + \text{const}\\ =& \mathbb{E}_{q(\mathbf{z})}[\ln \tilde{p}(\mathbf{y}, \mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\ln q(\mathbf{z})] + \text{const}, \quad (21)\end{aligned}$$

where $\tilde{p}(\mathbf{y}, \mathbf{z}))$ is a new distribution defined by the relation

$$\ln \tilde{p}(\mathbf{y}, \mathbf{z})) = \mathbb{E}_{q(\mathbf{f})}[\ln (p(\mathbf{y}|\mathbf{f}, \mathbf{z}) p(\mathbf{z}))] + \text{const}.$$

$\mathcal{L}_2$ is actually the negative KL divergence between $q(\mathbf{z})$ and $\tilde{p}(\mathbf{y}, \mathbf{z})$, which is maximized when $q(\mathbf{z}) = \tilde{p}(\mathbf{y}, \mathbf{z})$, i.e.,

$$\begin{aligned}\sum_{n=1}^{N} \ln q(z_n) =& \sum_{n=1}^{N} \sum_{k=1}^{T} \mathbb{E}_{q(f_k(\mathbf{x}_n))}[\ln p(y_n|f_k(\mathbf{x}_n))^{[z_n==k]}]\\ &+ \sum_{n=1}^{N} \sum_{k=1}^{T} \ln p(z_n = k)^{[z_n==k]}.\end{aligned}$$

With $p(z_n = k)$ given in Eq. (11), $q(z_n)$ is then a multinomial distribution, i.e., $q(z_n{=}k){=}r_{nk}$, where $r_{nk} = \rho_{nk}/\sum_{i=1}^{T}\rho_{ni}$ is the *responsibility* of expert k for $\mathbf{x}_n$, and ρ_{nk} is given by

$$\ln \rho_{nk} = \mathbb{E}_{q(f_k(\mathbf{x}_n))}[\ln p(y_n|f_k(\mathbf{x}_n))] + \ln \mathcal{N}(\mathbf{x}_n|\mathbf{m}_k, \mathbf{V}). \quad (22)$$

We assume that all the GPs have the same number of inducing points M. Most of the computational cost for both optimization steps arises from computing the expected likelihood terms $\mathbb{E}_{q(f_k(\mathbf{x}_n))}[\ln p(y_n|f_k(\mathbf{x}_n))]$ for $k = 0, ..., T$ and $n = 1, ..., N$. This computation has the overall time complexity of $O(NM^2T)$. Cost reduction for the second step comes from a careful inspection of Eq. (22). In particular, the first term in Eq. (22) measures the quality of prediction by expert k, which increases when $\mathbf{x}_n$ is similar to the inducing inputs $\mathbf{U}_k$. This is more likely as $\mathbf{x}_n$ is getting closer to $\mathbf{m}_k$, i.e., the second term increases. This observation allows us to bypass the expensive computation of the first term and arrive at a simplified assignment $\rho_{nk}{=}\mathcal{N}(\mathbf{x}_n|\mathbf{m}_k, \mathbf{V})$. To reduce computational cost for the first step, we assume that each data point is assigned to only one expert, which is the one with highest responsibility: $z_n = \mathrm{argmax}_k\, r_{nk}$. The responsibilities are then reassigned as: $q(z_n = k) = 1$ iff $z_n = k$ and $q(z_n = k) = 0$ otherwise. As a result, the term $q(f_k(\mathbf{x}_n))$ in Eq. (19) is only needed when $q(z_n{=}k)$ is non-zero, i.e., the point $\mathbf{x}_n$ is assigned to expert k. The time complexity is then reduced to $O(NM^2)$. The memory complexity is $O(NM)$.

Since the bound (18) includes the sum over data points, we can further reduce the computational cost by optimizing it in a stochastic fashion: selecting a mini-batch of data at random for each iteration. This gives the time and memory complexity of $\max(O(BM^2), O(M^3))$ and $\max(O(BM), O(M^2))$, where B is the batch size.

The predictive distribution for an unseen point $\mathbf{x}^*$ is given as:

$$p(y^*|\mathbf{x}^*, \mathbf{y}) \approx \sum_{k=1}^{T} p(z^* = k|\mathbf{x}^*) \int p(y^*|f_k(\mathbf{x}^*)) q(f_k(\mathbf{x}^*)) df_k(\mathbf{x}^*).$$

5. EXPERIMENTS

This section presents the experiments to evaluate the performance of the proposed *hierarchical mixture of GP experts* classifier (HMGPC) on multiple benchmark classification datasets of varying sizes.

We use the squared exponential (SE) kernel with automatic relevance determination (ARD) for all the GP classifiers. HMGPC is implemented in Python as an extension to GPflow, which is a GP library using TensorFlow [20] with the capability of making use of GPU for faster computation. Optimization is performed using *Adadelta* optimizer [21]. Experiments are carried out on a 3.47GHz CPU with 8GB of RAM. GPU is not enabled in the experiments.

5.1. UCI benchmark datasets

First, we evaluate HMGPC on a number of binary classification datasets of small to medium size from the UCI repository [22]. The datasets with their sizes and input dimensions are listed in Table 1. 5-fold cross-validation is used with all the datasets.

Performance of HMGPC is compared to a number of GP classifiers including the state-of-the-art *generalized FITC* (EP-FITC) [8], the *variational sparse GP classifier* (VSGPC) [9], the *mixture of GP classifiers* (MGPC) and the *full GP classifier with EP* (EP-GP) [23]. MGPC is a special case of HMGPC where the upper layer is removed. It is left with a set of local GP experts, and therefore makes use of only local information. EP-FITC and VSGPC use only global information where the entire dataset is summarized by a set of inducing points. The four methods HMGPC, MGPC, VSGPC and EP-FITC have the same computational complexity of $O(NM^2)$ in time and $O(NM)$ in memory. The number of inducing points M is set to 2.5% of the training size for all of these methods for fair comparison. The number of clusters in HMGPC and MGPC is fixed to 3. EP-GP is the full GP classifier in which EP is used to approximate the posterior. It has the time and memory complexity of $O(N^3)$ and $O(N^2)$, respectively. Therefore, it cannot be tested on the datasets with significantly more than a thousand samples.

In this experiment, stochastic optimization is not used (i.e., $B = N$). For the methods that are affected by random factors, each of them is run 5 times. The optimization process of each method is run until convergence or until it reaches 1000 iterations, whichever is the earlier. The average error rates across different runs and folds together with their standard deviations are reported in Table 1. The average training times are also reported. HMGPC gives the best performance in all the tested datasets. It provides big gains in error rates over the second best classifier in 4 out of 6 datasets (26.3% in *splice*, 48.2% in *WFRN*, 15.6% in *phishing* and 29.8% in *EEG*). Interestingly, it even outperforms EP-GP which requires much more training time and memory.

5.2. The US Flight dataset

This dataset has more than 2 million samples and is originally used in [6] for regression task to predict the flight delay based on 8 attributes. Here we consider the binary classification task to predict whether a flight was delayed or not, i.e., whether its delay time is more than 15 minutes. We randomly select 1 million points for training and 100K points for test. Such a large dataset is prohibitive for normal sparse GP classifiers such as EP-FITC, but can be handled by HMGPC and VSGPC with stochastic optimization. Each of these two methods is tested with 1000 inducing points and two different batch sizes of 2500 and 5000. HMGPC uses 3 clusters in its lower-layer. As baselines, we use logistic regression, random forest with depth of 2 and 100 estimators, decision tree with a maximum depth of 2, AdaBoost with decision tree as base predictor, and linear SVM [24].

Table 1: Error rates (%) (along with their standard deviations in brackets) and training times (s) on UCI benchmark datasets. The best performances are shown in **bold**. The size and input dimension of each dataset are given under its name.

	Datasets ($N\backslash D$)	*splice* (1000\60)	*german* (1000\24)	*CTG* (975\23)	*WFRN* (5456\24)	*phishing* (11055\68)	*EEG* (14980\15)
Error rate	HMGPC	**5.6 (± 1.5)**	**22.4 (± 5.3)**	**7.7 (± 2.4)**	**2.9 (± 0.5)**	**3.8 (± 0.6)**	**3.3 (± 0.1)**
	VSGPC	7.6 (± 1.3)	23.2 (± 4.3)	8.2 (± 2.3)	5.6 (± 1.0)	4.7 (± 0.4)	4.7 (± 0.2)
	MGPC	15.6 (± 3.1)	24.6(± 4.2)	28.6(± 18.5)	12.1 (± 0.6)	7.8 (± 0.6)	31.7 (± 0.5)
	EP-FITC	18.0 (± 14.7)	25.4 (± 4.7)	9.2 (± 5.1)	6.6 (± 1.9)	4.5 (± 0.3)	44.9 (± 0.9)
	EP-GP	9.9 (± 5.0)	22.5 (± 5.9)	7.8 (± 2.2)	- (-)	- (-)	- (-)
Training time	HMGPC	53	40	47	382	3246	2592
	VSGPC	38	26	34	232	2063	1862
	MGPC	39	25	39	263	2391	1726
	EP-FITC	752	628	264	2964	8840	2666
	EP-GP	8876	6735	29567	- (-)	- (-)	- (-)

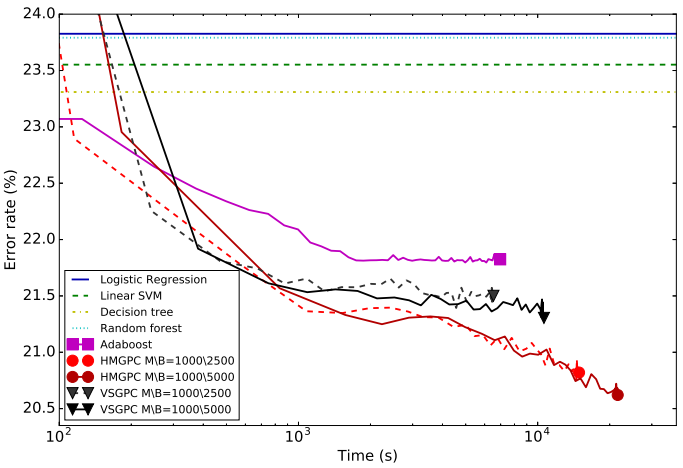

Fig. 2: Average test error rates vs. training time on *US flight* dataset.

The test error rates as functions of training time are shown in Fig. 2. It can be seen that both VSGPC and HMGPC are able to exceed the accuracy of all the baseline methods in a just few minutes. HMGPC outperforms all the other methods in terms of performance-time trade-offs. It also gives the lowest error rate at convergence.

6. CONCLUSION

In this article, a novel GP classification method was presented based on a hierarchical structure of sparse GPs. The model exploits both global and local information from the data through a two-layer model with a sparse global GP in the upper layer and a mixture of sparse GPs in the lower layer. Simultaneous learning of the GPs and the gating network is achieved by minimizing a variational lower bound of the log marginal likelihood. Experiments on benchmark datasets showed that the proposed model outperforms many state-of-the-art sparse GP methods and generic classifiers. Stochastic optimization is also supported to cater for large-scale problems.

Acknowledgments

This work is supported by a grant from the Australian Research Council.

7. REFERENCES

[1] Malte Kuss and Carl Edward Rasmussen, "Assessing approximate inference for binary Gaussian process classification," *Journal of Machine Learning Research*, vol. 6, pp. 1679–1704, 2005.

[2] Hannes Nickisch and Carl Edward Rasmussen, "Approximations for binary Gaussian process classification," *Journal of Machine Learning Research*, vol. 9, no. 10, pp. 2035–2078, 2008.

[3] Joaquin Quiñonero-Candela and Carl Edward Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[4] Edward Snelson and Zoubin Ghahramani, "Sparse Gaussian processes using pseudo-inputs," *Advances in Neural Information Processing Systems*, vol. 18, pp. 1257–1264, 2006.

[5] Michalis K Titsias, "Variational learning of inducing variables in sparse Gaussian processes," *Proc. 12th Intern. Conf. on Artificial Intelligence and Statistics*, pp. 567–574, Florida, USA, Apr. 16–18, 2009.

[6] James Hensman, Nicolo Fusi, and Neil D Lawrence, "Gaussian processes for big data," *Proc. 29th Conf. on Uncertainty in Artificial Intelligence*, pp. 282–290, Bellevue, WA, USA, Jul. 11–15, 2013.

[7] Neil Lawrence, Matthias Seeger, and Ralf Herbrich, "Fast sparse Gaussian process methods: The informative vector machine," *Advances in Neural Information Processing Systems*, vol. 15, pp. 625–632, 2003.

[8] Andrew Naish-Guzman and Sean Holden, "The generalized fitc approximation," *Advances in Neural Information Processing Systems*, vol. 20, pp. 1057–1064, 2007.

[9] James Hensman, Alexander Matthews, and Zoubin Ghahramani, "Scalable variational Gaussian process classification," *Proc. 8th Intern. Conf. on Artificial Intelligence and Statistics*, pp. 351–360, California, USA, May 9–12, 2015.

[10] Carl Edward Rasmussen and Zoubin Ghahramani, "Infinite mixtures of Gaussian process experts," *Advances in Neural Information Processing Systems*, vol. 14, pp. 881–888, 2002.

[11] Volker Tresp, "Mixtures of Gaussian processes," *Advances in Neural Information Processing Systems*, vol. 13, pp. 654–660, 2000.

[12] Jian Qing Shi, Roderick Murray-Smith, and DM Titterington, "Bayesian regression and classification using mixtures of Gaussian processes," *Intern. Journal of Adaptive Control and Signal Processing*, vol. 17, no. 2, pp. 149–161, 2003.

[13] Chao Yuan and Claus Neubauer, "Variational mixture of Gaussian process experts," *Advances in Neural Information Processing Systems*, vol. 21, pp. 1897–1904, 2009.

[14] Trung Nguyen and Edwin Bonilla, "Fast allocation of Gaussian process experts," *Proc. 31st Intern. Conf. on Machine Learning*, pp. 145–153, Beijing, China, Jun. 21–26, 2014.

[15] Edward Meeds and Simon Osindero, "An alternative infinite mixture of Gaussian process experts," *Advances in Neural Information Processing Systems*, vol. 18, pp. 883–890, 2006.

[16] Shiliang Sun and Xin Xu, "Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 466–475, 2011.

[17] Christopher KI Williams and Carl Edward Rasmussen, "Gaussian processes for machine learning," *the MIT Press*, vol. 2, no. 3, pp. 4, 2006.

[18] M Abramowitz and I Stegun, "Handbook of mathematical functions with formulas, graphs, and mathematical tables (9th printing) dover," *New York*, p. 890, 1972.

[19] Manfred Opper and Cédric Archambeau, "The variational Gaussian approximation revisited," *Neural computation*, vol. 21, no. 3, pp. 786–792, 2009.

[20] Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman, "GPflow: A Gaussian process library using TensorFlow," *Journal of Machine Learning Research*, vol. 18, no. 40, pp. 1–6, apr 2017.

[21] Matthew D Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[22] Arthur Asuncion and David Newman, " The UC Irvine Machine Learning Repository," `https://archive.ics.uci.edu/ml/datasets.html`, 2007, [Online; accessed 19-Octorber-2016].

[23] Hyun-Chul Kim and Zoubin Ghahramani, "Bayesian Gaussian process classification with the EM-EP algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1948–1959, 2006.

[24] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, no. Aug, pp. 1871–1874, 2008.