Gaussian process classification using posterior linearisation

Ángel F. García-Fernández, Filip Tronarp, Simo Särkkä

***Abstract*—This paper proposes a new algorithm for Gaussian process classification based on posterior linearisation (PL). In PL, a Gaussian approximation to the posterior density is obtained iteratively using the best possible linearisation of the conditional mean of the labels and accounting for the linearisation error. PL has some theoretical advantages over expectation propagation (EP): all calculated covariance matrices are positive definite and there is a local convergence theorem. In experimental data, PL has better performance than EP with the noisy threshold likelihood and the parallel implementation of the algorithms.**

***Index Terms*—Gaussian process classification, posterior linearisation, Bayesian inference.**

I. Introduction

Classification is an important problem with a large number of applications, for example, in handwriting and speech recognition, and medical diagnosis [1]. In (supervised) classification, a set of training data points with their corresponding classes are available to learn the underlying structure of the problem. Based on this information, the objective is to infer the classes of new data points. This classification problem can be posed using Gaussian processes (GPs) [2]–[8].

In binary GP classification, it is assumed that there is a latent function, distributed as a GP [2], whose value at a certain data point is related to the probability that this data point belongs to one class. The GP prior has some hyperparameters that can be marginalised out [9], or estimated by maximising the log marginal likelihood [2]. Then, for the estimated hyperparameters, we compute the posterior distribution over the latent function evaluated at the training data points, which in turn allows us to predict the classes for new data points.

The main difficulties in GP classification are the approximations of the posterior and the log marginal likelihood. Markov chain Monte Carlo algorithms [10] can provide very accurate approximations, but they usually have a high computational burden. This is the reason why there is interest in using computationally efficient approximations. We proceed to review two of such approximations in the literature, though other approaches exists [11]–[13].

One possibility is to use the Laplace approximation [2]. A drawback of the Laplace approximation is that it cannot handle likelihood functions in which the gradient is zero almost everywhere, such as the noisy threshold likelihood [4].

A. F. García-Fernández is with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 3GJ, United Kingdom. He is also with the ARIES Research Center, Universidad Antonio de Nebrija, Madrid, Spain (email: angel.garcia-fernandez@liverpool.ac.uk). F. Tronarp and S. Särkkä are with the Department of Electrical Engineering and Automation, Aalto University, 02150 Espoo, Finland (emails: {filip.tronarp, simo.sarkka}@aalto.fi).

Expectation propagation (EP) [14] can be used with all likelihood functions and often outperforms Laplace approximation in GP classification [15]–[17]. EP is an iterative algorithm in which, at each iteration, a Gaussian approximation to the likelihood for one data point is reapproximated. In order to do this, we first remove the considered approximated likelihood from the posterior approximation, which results in the "cavity" distribution. Then, we use the true likelihood and the cavity distribution to provide a new Gaussian approximation to the likelihood by performing moment matching. EP has two drawbacks that are relevant to this paper: 1) the cavity distributions can have negative-definite covariance matrices with possibly large negative eigenvalues that are not due to numerical errors [1], [18], and 2) there is no convergence proof in the literature that indicates conditions of convergence [2]. In order to deal with 1), simple, ad-hoc solutions are sometimes used, for example, processing the likelihoods in a different order to see if this resolves the issue, or arbitrarily setting the negative-definite covariance to a predefined positive-definite matrix [18]. More robust EP algorithms, such as damped EP or double loop algorithms, also require pragmatic solutions to avoid negative-definite covariance matrices [19].

This paper proposes an algorithm for GP classification based on posterior linearisation (PL) [20], [21]. In PL, we compute a Gaussian approximation to the posterior distribution by linearising the conditional mean of each label, in the region where the posterior lies, and by setting the conditional covariance to a value that accounts for the linearisation error. Importantly, the selection of the linearisation parameters is done in an optimal way by minimising the mean square error of the linearisation. This optimal linearisation is given by statistical linear regression (SLR) [22] of the conditional mean with respect to the posterior.

In practice, PL is implemented by an iterative procedure, in which we can process the likelihoods sequentially or in parallel. PL has some advantages compared to EP: 1) PL always provides positive-definite covariance matrices, so ad-hoc fixes are not necessary, and 2) there is a local convergence proof [21] that indicates sufficient local conditions of convergence. The proposed algorithm also includes a method to approximate the marginal likelihood for estimating the hyperparameters. In the analysed experimental data, EP and PL have comparable performance in terms of classification errors except in the parallel implementations for the noisy threshold likelihood, where PL provides lower errors.

II. Problem formulation

We consider a set $\mathcal{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$, which contains n data points $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)$, with $\mathbf{x}_i \in \mathbb{R}^{n_x}$, where n_x is the dimension of each data point, and their binary labels $\mathbf{y} = (y_1, ..., y_n)$, with $y_i \in \{-1, 1\}$. In binary classification, given this set, we are interested in predicting the binary labels $\mathbf{y}^\star = (y_1^\star, ..., y_m^\star)$ for m new data points $\mathbf{x}^\star = (\mathbf{x}_1^\star, ..., \mathbf{x}_m^\star)$ [2]. We proceed to explain how this problem is formulated using GPs.

It is assumed that the label y_i of a data point $\mathbf{x}_i$ only depends on the value of a latent real-valued function evaluated at $\mathbf{x}_i$, $f_i = f(\mathbf{x}_i)$. Then, there are several widely-used models for the probability mass function $p(y_i|f_i)$, for example, the probit, logit, and noisy threshold, whose respective $p(y_i|f_i)$ are [4]

$$p(y_i|f_i) = \Phi(y_i f_i), \tag{1}$$

$$p(y_i = 1|f_i) = 1/(1 + \exp(-f_i)), \tag{2}$$

$$p(y_i|f_i) = \epsilon + (1 - 2\epsilon) H(y_i f_i), \tag{3}$$

where $\Phi(z) = \int_{-\infty}^{z} \mathcal{N}(x; 0, 1)\, dx$, $\mathcal{N}(\cdot; \overline{x}, P)$ is the Gaussian density with mean $\overline{x}$ and covariance P, $\epsilon \in (0, 1)$, and $H(z) = 1$ if $z > 0$, and $H(z) = 0$ otherwise.

It is further assumed that function $f(\cdot)$ is distributed according to a zero-mean Gaussian process with a given covariance function $k_\theta(\cdot, \cdot)$, where $\theta \in \mathbb{R}^{n_\theta}$ is a vector that contains all the hyperparameters. As a result, the prior density of $\mathbf{f} = [f_1, ..., f_n]^\top$ becomes

$$p(\mathbf{f}|\mathbf{x}, \theta) = \mathcal{N}(\mathbf{f}; \mathbf{0}, \mathbf{K}), \tag{4}$$

where $\mathbf{K}$ is an $n \times n$ covariance matrix such that $\mathbf{K}_{ij} = k_\theta(\mathbf{x}_i, \mathbf{x}_j)$. The posterior of $\mathbf{f}$ is

$$p(\mathbf{f}|\mathcal{D}; \theta) = \left[\prod_{i=1}^{n} p(y_i|f_i)\right] p(\mathbf{f}|\mathbf{x}; \theta) / p(\mathcal{D}; \theta), \tag{5}$$

where the marginal likelihood is

$$p(\mathcal{D}; \theta) = \int \left[\prod_{i=1}^{n} p(y_i|f_i)\right] p(\mathbf{f}|\mathbf{x}; \theta)\, \mathrm{d}\mathbf{f}. \tag{6}$$

The hyperparameters θ are usually not known and are often estimated by maximising (6) [2]. Once we have estimated θ, the posterior over $\mathbf{f}^\star = \mathbf{f}(\mathbf{x}^\star) = [f_1^\star, ..., f_m^\star]^\top$, which considers new data points, is computed as

$$p(\mathbf{f}^\star|\mathcal{D}, \mathbf{x}^\star; \theta) = \int p(\mathbf{f}^\star|\mathcal{D}, \theta, \mathbf{x}^\star, \mathbf{f})\, p(\mathbf{f}|\mathcal{D}; \theta)\, \mathrm{d}\mathbf{f}. \tag{7}$$

Finally, all information about the class labels for the new data points is given by the distribution of $\mathbf{y}^\star$ given $\mathcal{D}$, $\mathbf{x}^\star$ and θ, which can be written as

$$p(\mathbf{y}^\star|\mathcal{D}, \mathbf{x}^\star; \theta) = \int \left[\prod_{i=1}^{m} p(y_i^\star|f_i^\star)\right] p(\mathbf{f}^\star|\mathcal{D}, \mathbf{x}^\star; \theta)\, \mathrm{d}\mathbf{f}^\star. \tag{8}$$

Based on this distribution, we can predict the class labels, which is our main objective, for example, by computing their expected value [2]. Unfortunately, none of the densities of interest, (5)-(8), has a closed-form expression, so approximations are necessary. As in the Laplace and EP methods, in this paper, we consider Gaussian approximations of (5) and (7).

A. Enabling approximation

We obtain a Gaussian approximation to the posterior by approximating the conditional mean $\mathrm{E}[y_i|f_i]$ as an affine function and the conditional variance $\mathrm{C}[y_i|f_i]$ as a constant

$$\mathrm{E}[y_i|f_i] \approx A_i f_i + b_i, \quad \mathrm{C}[y_i|f_i] \approx \Omega_i, \tag{9}$$

where $A_i \in \mathbb{R}$, $b_i \in \mathbb{R}$, and $\Omega_i > 0$, and the conditional moments $\mathrm{E}[y_i|f_i]$ and $\mathrm{C}[y_i|f_i]$ are taken with respect to $p(y_i|f_i)$, which is a discrete distribution.

Under approximation (9), the linear mean square error (LMMSE) estimate $\overline{\mathbf{u}}$ of $\mathbf{f}$ and its mean square error matrix $\mathbf{W}$ are available in closed-form [23]

$$\overline{\mathbf{u}} = \mathbf{K}\mathbf{A}^\top \left(\mathbf{A}\mathbf{K}\mathbf{A}^\top + \mathbf{\Omega}\right)^{-1} (\mathbf{y} - \mathbf{b}), \tag{10}$$

$$\mathbf{W} = \mathbf{K} - \mathbf{K}\mathbf{A}^\top \left(\mathbf{A}\mathbf{K}\mathbf{A}^\top + \mathbf{\Omega}\right)^{-1} \mathbf{A}\mathbf{K}, \tag{11}$$

where $\mathbf{A} = \mathrm{diag}([A_1, ..., A_n])$, $\mathbf{b} = [b_1, ..., b_n]^\top$ and $\mathbf{\Omega} = \mathrm{diag}([\Omega_1, ..., \Omega_n])$. Then, the posterior (5) is approximated as Gaussian with mean $\overline{\mathbf{u}}$ and covariance matrix $\mathbf{W}$, which implies that the posterior of $\mathbf{f}^\star$, see (7), is Gaussian with mean $\overline{\mathbf{u}}^\star$ and covariance matrix $\mathbf{W}^\star$

$$\overline{\mathbf{u}}^\star = (\mathbf{K}^\star)^\top \mathbf{A}^\top \left(\mathbf{A}\mathbf{K}\mathbf{A}^\top + \mathbf{\Omega}\right)^{-1} (\mathbf{y} - \mathbf{b}), \tag{12}$$

$$\mathbf{W}^\star = \mathbf{K}^{\star\star} - (\mathbf{K}^\star)^\top \mathbf{A}^\top \left(\mathbf{A}\mathbf{K}\mathbf{A}^\top + \mathbf{\Omega}\right)^{-1} \mathbf{A}\mathbf{K}^{*}, \tag{13}$$

where $\mathbf{K}^\star$ and $\mathbf{K}^{\star\star}$ are $n \times m$ and $m \times m$ matrices such that $\mathbf{K}^\star_{ij} = k_\theta\left(\mathbf{x}_i, \mathbf{x}_j^\star\right)$ and $\mathbf{K}^{\star *}_{ij} = k_\theta\left(\mathbf{x}_i^\star, \mathbf{x}_j^\star\right)$.

The quality of the approximations of the posterior moments (10)-(13) only depends on the choice of A_i, b_i, Ω_i for $i = 1, ..., n$, so it is of utmost importance to select them properly.

III. Posterior linearisation of GPs

In this section, we first explain SLR using conditional moments in Section III-A. Then, we explain iterated posterior linearisation in Section III-B. We propose a method for approximating the marginal likelihood in Section III-C. Finally, a discussion of the algorithm is provided in Section III-D.

A. Statistical linear regression of conditional moments

With SLR, we can optimally linearise $\mathrm{E}[y_i|f_i]$ in a mean square error sense to make approximation (9). In addition, SLR provides us with the linearisation error, which is used to approximate $\mathrm{C}[y_i|f_i]$ as a constant, as required in (9). The SLR linearisation parameters are denoted as $\left(A_i^+, b_i^+, \Omega_i^+\right)$ and we proceed to explain how to obtain them.

In SLR of random variables [21], we are given a density $p(\cdot)$ on variable f_i, whose first two moments are $\overline{f_i}$ and P_i, and the conditional moments $\mathrm{E}[y_i|f_i]$ and $\mathrm{C}[y_i|f_i]$, which describe the relation between y_i and f_i. Parameters A_i and b_i are then selected by minimising the mean square error over random variables y_i and f_i

$$\mathrm{E}_{f_i, y_i}\left[(y_i - A_i f_i - b_i)^2\right] = \mathrm{E}_{f_i}\left[(\mathrm{E}[y_i|f_i] - A_i f_i - b_i)^2\right] + \mathrm{E}_{f_i}\left[\mathrm{C}[y_i|f_i]\right], \tag{14}$$

where we have highlighted the expectations that are taken with respect to variables f_i and y_i. Therefore,

$$\left(A_i^+, b_i^+\right) = \underset{A_i, b_i}{\arg\min}\, \mathrm{E}_{f_i}\left[(\mathrm{E}[y_i|f_i] - A_i f_i - b_i)^2\right].$$

In SLR, the parameter Ω_i^+ is the resulting mean square error in (14) for the optimal values A_i^+ and b_i^+ so

$$\Omega_i^+ = \mathrm{E}_{f_i}\left[\left(\mathrm{E}\left[y_i|f_i\right] - A_i^+ f_i - b_i^+\right)^2\right] + \mathrm{E}_{f_i}\left[\mathrm{C}\left[y_i|f_i\right]\right].$$

In other words, SLR makes the best affine fit of the conditional mean $\mathrm{E}\left[y_i|f_i\right]$ in the region indicated by $p\left(\cdot\right)$, the density of f_i, and sets $\mathrm{C}\left[y_i|f_i\right]$ as the corresponding mean square error. The resulting $\left(A_i^+, b_i^+, \Omega_i^+\right)$ is given by [21]

$$A_i^+ = \mathrm{C}_{f_i}\left[f_i, \mathrm{E}\left[y_i|f_i\right]\right]/P_i, \quad b_i^+ = \mathrm{E}_{f_i}\left[\mathrm{E}\left[y_i|f_i\right]\right] - A_i^+ \overline{f_i}, \tag{15}$$

$$\Omega_i^+ = \mathrm{C}_{f_i}\left[\mathrm{E}\left[y_i|f_i\right]\right] + \mathrm{E}_{f_i}\left[\mathrm{C}\left[y_i|f_i\right]\right] - \left(A_i^+\right)^2 P_i, \tag{16}$$

where $\mathrm{C}_{f_i}\left[\cdot\right]$ denotes both the variance and covariance with respect to f_i. We can then obtain $\left(A_i^+, b_i^+, \Omega_i^+\right)$ in terms of moments of $\mathrm{E}\left[y_i|f_i\right]$ and $\mathrm{C}\left[y_i|f_i\right]$ with respect to density $p\left(\cdot\right)$. Expressions of these moments for the likelihoods (1)-(3) are given in the supplementary material.

B. Iterated posterior linearisation

This section explains how to use SLR to make approximations (9) in an optimal fashion. If we did not know the labels $y_1, ..., y_n$, the best approximation of the conditional moments would be given by SLR with respect to the prior, which is given by (4), as this density indicates the region where f_i lies. However, we know these labels, and the insight of posterior linearisation [20] is that, given these labels, the best linearisation $\left(A_i, b_i\right)$, in a mean square error sense, and the resulting mean square error Ω_i are given by SLR with respect to the posterior.

Direct application of posterior linearisation is not useful as we need to know the posterior to select $\left(\mathbf{A}, \mathbf{b}, \mathbf{\Omega}\right)$, which is used to calculate the posterior. Nevertheless, posterior linearisation can be approximated by using iterated SLRs, giving rise to iterated posterior linearisation. That is, as we do not know the posterior, we perform SLR with respect to the best available approximation of the posterior. At the end of each iteration, we expect to obtain an improved approximation of the posterior, which is used to compute an improved SLR at the next iteration. The steps of the parallel version of the algorithm are:

1) Set $j = 1$ and $\overline{\mathbf{u}}^j = \mathbf{0}$, $\mathbf{W}^j = \mathbf{K}$.
2) For $i = 1, ..., n$, compute $\left(A_i^j, b_i^j, \Omega_i^j\right)$ using SLR with respect to the ith marginal of a Gaussian with moments $\left(\overline{\mathbf{u}}^j, \mathbf{W}^j\right)$, see (15)-(16).
3) With the current linearisation, $\left(A_i^j, b_i^j, \Omega_i^j\right)$ for $i = 1, ..., n$, compute the new posterior moments $\left(\overline{\mathbf{u}}^{j+1}, \mathbf{W}^{j+1}\right)$ using (10) and (11), where only the values of $\left(\mathbf{A}, \mathbf{b}, \mathbf{\Omega}\right)$ change.
4) Set $j = j + 1$ and repeat from step 2 for a fixed number of iterations or until some convergence criterion is met.

It is important to notice that in the aforementioned algorithm, we first relinearise all likelihoods and then compute the posterior with the current linearisation. This is beneficial from a computational point of view because the linearisations can be done in parallel and we only perform one update per iteration. Nevertheless, it is also possible to recompute the posterior after relinearising a single likelihood. In GP classification, $\mathbf{K}$ can be close to singular so we have adapted the numerically stable implementations of Laplace and EP algorithms in [2] to our PL implementations.

C. Marginal likelihood approximation

In this section, we propose the use of sigma-point/quadrature rules to approximate the marginal likelihood, which is given by (6). Importantly, we select the quadrature points with respect to the posterior approximation, as this density has its mass in the region where the integrand is high. We therefore write (6) as

$$p\left(\mathcal{D};\theta\right) = \hat{p}\left(\mathcal{D};\theta\right)\int\left[\prod_{i=1}^{n}\frac{p\left(y_i|f_i\right)}{\hat{p}\left(y_i|f_i\right)}\right] \times \frac{\left[\prod_{i=1}^{n}\hat{p}\left(y_i|f_i\right)\right]\mathcal{N}\left(\mathbf{f};\mathbf{0},\mathbf{K}\right)}{\hat{p}\left(\mathcal{D};\theta\right)}d\mathbf{f}, \tag{17}$$

where $\hat{p}\left(\mathcal{D};\theta\right) = \mathcal{N}\left(\mathbf{y};\mathbf{b},\mathbf{AKA}^\top + \mathbf{\Omega}\right)$ and $\hat{p}\left(y_i|f_i\right) = \mathcal{N}\left(y_i; A_i f_i + b_i, \Omega_i\right)$ $i = 1, ..., n$. Consequently, the marginal likelihood can be seen as $\hat{p}\left(\mathcal{D};\theta\right)$ times a correction factor that depends on the similarity between $p\left(y_i|f_i\right)$ and $\hat{p}\left(y_i|f_i\right)$ in the region indicated by the posterior density.

There are some drawbacks when integrating with respect to the joint density of $\mathbf{f}$ using sigma-points/quadrature rules. First, accurate and computationally efficient integration in high-dimensional spaces is more difficult than in low-dimensional spaces. Second, sigma-point/quadrature rules require the Cholesky decomposition of the covariance matrix and this covariance can be ill-conditioned in GP classification [2], so it is not always possible to compute it. We therefore discard correlations in the posterior for approximating the correction factor in (17) such that

$$p\left(\mathcal{D};\theta\right) \approx \hat{p}\left(\mathcal{D};\theta\right)\prod_{i=1}^{n}\int\left[\frac{p\left(y_i|f_i\right)}{\hat{p}\left(y_i|f_i\right)}\right]\mathcal{N}\left(f_i;\overline{u}_i, W_i\right)\mathrm{d}f_i, \tag{18}$$

where $\overline{u}_i$ and W_i represent the posterior mean and variance of f_i, which are obtained from (10) and (11).

In short, in order to compute (18), we compute the posterior moments, $\overline{\mathbf{u}}$ and $\mathbf{W}$, and the resulting SLR parameters $\left(\mathbf{A}, \mathbf{b}, \mathbf{\Omega}\right)$, which are required in $\hat{p}\left(\mathcal{D};\theta\right)$ and $\hat{p}\left(y_i|f_i\right)$. Accurate approximation of (17) is quite important as it is used to estimate the hyperparameters θ. Without an accurate estimation of θ, the results of a GP classifier are poor, as the GP does not model the training data properly. The results in Section IV indicate that approximation (18) is accurate for classification purposes.

D. Discussion

The iterated SLRs of PL can be done in parallel for each likelihood and sequentially. We can also combine both types of linearisation approaches, for example, by performing several updates sequentially and then in parallel. Importantly, the main benefit of PL compared to EP is that all involved densities in PL have positive-definite covariance matrices. This is ensured by the fact that $\mathbf{\Omega}$ is positive definite by definition. Numerical inaccuracies could render a negative-definite $\mathbf{\Omega}$ but this would be easy to address, as its eigenvalues would be close to zero. Another option is to use square root solutions [24].

As EP, iterated PL is not ensured to converge in general. Nevertheless, there is a local convergence proof, which is given in [21, Thm. 2], that states sufficient conditions for convergence.

IV. Experimental results

This section assesses Laplace, EP, and PL, in their parallel and sequential forms, in six real-world data sets from [25]. One additional synthetic example that analyses a case where EP fails is provided in the supplementary material. In particular, we consider the data sets: breast cancer (9, 699), crab gender (6, 200), glass chemical (9, 214), ionosphere (33, 351), thyroid (5,215) and housing (13,506), where the number of attributes (dimension of data points) and data points in each data set are given in parentheses. For the last two data sets, the groups for binary classification are formed as in [4]. Data points have been normalised to have zero mean and an identity covariance matrix [15]. We use ten-fold cross-validation [2] to compute the average classification error.

We use the covariance function [10]

$$k_\theta\left(x_i, x_j\right) = \sigma_1^2 \exp\left(-\frac{1}{2\ell^2}\left\|x_i - x_j\right\|^2\right) + \sigma_2^2 \delta\left[i-j\right],$$

where $\delta\left[\cdot\right]$ denotes the Kronecker delta, σ_2^2 is set to 0.1, the hyperparameters $\theta = \left(\sigma_1^2, \ell\right)$, and $x_i \neq x_j$ for $i \neq j$. EP and Laplace have been implemented as indicated in [2].

We report results for all the likelihoods in (1)-(3). The integrations that do not admit closed-form expressions are approximated using a Gauss-Hermite quadrature of order 10. We first estimate the hyperparameters θ by maximising (6) using the BFGS Quasi-Newton algorithm implemented in Matlab function *fminunc*. The optimisation method is run on variable $\ln\left(\theta\right)$ with initial point $\left(\ln \sigma_1^2, \ln \ell\right) = \left(\ln\left(10\right), \ln\left(1\right)\right)$. Then, we approximate the posterior (5) as Gaussian and, finally, we compute the expected value of (8) to predict the label for each test data point. We consider 10 iterations for EP and PL. If the variance of a likelihood approximation is negative for EP, it is set to a small positive number to avoid negative-definite covariance matrices, as suggested in [18].

The resulting average classification errors for each data set and averaged over all data sets (Ave.) are shown in Table I, where Prob., Log., and NT stand for probit, logit, and noisy threshold. PEP and SEP refer to parallel and sequential EP, and PPL and SPL to parallel and sequential PL. The cases where the classification error is considerably high are bolded. In general, sequential algorithms work better than parallel algorithms, though, as we will analyse, with an increase in the computational burden. In fact, SEP and SPL show comparable results. The highest differences among the methods appear with the noisy threshold likelihood and the parallel implementations. In this case, PPL works well but PEP provides high errors in four out of six experiments. The reason is that EP returns negative-definite covariance matrices for this likelihood and the fixes do not make PEP attain a low error. Laplace does not work with this likelihood, as the gradient of the likelihood is zero.

Table I: Average classification errors for real data sets.

Like.	Alg.	Can.	Crab	Glass	Ionos.	Thy.	Hous.	Ave.
Prob.	L	0.051	0.050	0.067	0.108	0.061	0.069	0.067
	PEP	0.034	0.045	0.067	0.088	0.062	0.064	0.057
	SEP	0.034	0.045	0.067	0.088	0.062	0.064	0.057
	PPL	0.037	0.035	0.076	0.083	0.071	0.069	0.059
	SPL	0.034	0.045	0.067	0.091	0.057	0.069	0.058
Log.	L	0.036	0.045	0.071	0.105	0.057	0.067	0.061
	PEP	**0.251**	0.045	0.071	0.083	0.062	0.070	**0.127**
	SEP	0.034	0.045	0.067	0.083	0.062	0.070	0.057
	PPL	0.039	0.040	0.071	0.088	0.071	0.067	0.060
	SPL	0.034	0.045	0.067	0.083	0.057	0.067	0.056
NT	L	-	-	-	-	-	-	-
	PEP	**0.202**	0.085	0.071	**0.384**	**0.454**	**0.124**	**0.214**
	SEP	0.034	0.045	0.067	0.086	0.062	0.069	0.058
	PPL	0.043	0.025	0.081	0.091	0.062	0.058	0.058
	SPL	0.034	0.035	0.067	0.091	0.057	0.070	0.057

For the logit model, PEP provides a high error in the cancer data set. This is a numerical error that can be solved by using a Gauss-Hermite quadrature rule of order 32, which increases the computational load. In contrast, PPL is able to work well in this data set with the Gauss-Hermite quadrature rule of order 10. The sequential processing of the data for EP and PL work well in this data set. In the rest of the experiments, EP, PL and Laplace provide comparable results, though there are slight differences in the errors. Importantly, parallel PL is more robust than parallel EP, as errors are not markedly high in any of the data sets. We think this is a relevant, practical advantage of PL over EP.

Finally, we provide the average computational times of our Matlab implementations (probit model and crab data set), as a indication of their computational complexities, though running times depend on the data set. With an Intel Core Xenon processor, we have: 0.4 s (Laplace), 0.7 s (PEP), 6.1 s (SEP), 0.8 s (PPL), and 5.5 s (SPL). Laplace is the fastest algorithm. PEP is slightly faster than PPL, as its optimisation requires a slightly fewer number of iterations to converge, although PEP has robustness problems for the noisy threshold likelihood. Sequential algorithms are slower, but they usually perform better than parallel algorithms. In this case, SPL is marginally faster than SEP and both methods have comparable performance.

V. Conclusions

We have proposed a novel method for classification using Gaussian processes based on posterior linearisation. The proposed algorithm consists of performing iterated statistical linear regressions with respect to the current approximation to the posterior. An important benefit compared to expectation propagation is that the proposed method does not provide negative-definite covariance matrices by construction, which implies that it does not need ad-hoc procedures to solve the resulting problems. In addition, posterior linearisation has a local convergence theorem. In the experimental results, PL and EP have comparable performance except in the parallel implementations with the noisy threshold likelihood where PL performs better than EP.

References

[1] K. P. Murphy, *Machine Learning: a Probabilistic Perspective*. The MIT Press, 2012.

[2] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[3] F. Pérez-Cruz, S. V. Vaerenbergh, J. J. Murillo-Fuentes, M. Lázaro-Gredilla, and I. Santamaría, "Gaussian processes for nonlinear signal processing: An overview of recent advances," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 40–50, July 2013.

[4] H.-C. Kim and Z. 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, Dec. 2006.

[5] K. Markov and T. Matsui, "Music genre classification using Gaussian process models," in *IEEE International Workshop on Machine Learning for Signal Processing*, Sept 2013, pp. 1–6.

[6] Y. Xiao, H. Wang, and W. Xu, "Hyperparameter selection for Gaussian process one-class classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 2182–2187, Sept. 2015.

[7] P. Morales-Álvarez, A. Pérez-Suay, R. Molina, and G. Camps-Valls, "Remote sensing image classification with large-scale Gaussian processes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 1103–1114, Feb 2018.

[8] P. M. Olmos, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Joint nonlinear channel equalization and soft LDPC decoding with Gaussian processes," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1183–1192, March 2010.

[9] J. Vanhatalo, V. Pietiläinen, and A. Vehtari, "Approximate inference for disease mapping with sparse Gaussian processes," *Statistics in Medicine*, vol. 29, July 2010.

[10] R. M. Neal, "Regression and classification using Gaussian process priors," in *Bayesian Statistics 6*, J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, Eds. Oxford University Press, 1998.

[11] M. N. Gibbs and D. J. C. Mackay, "Variational Gaussian process classifiers," *IEEE Transactions on Neural Networks*, vol. 11, no. 6, pp. 1458–1464, Nov 2000.

[12] M. Opper and O. Winther, "Gaussian processes for classification: Mean-field algorithms," *Neural Computation*, vol. 12, no. 11, pp. 2655–2684, 2000.

[13] J. Hensman, A. Matthews, and Z. Ghahramani, "Scalable variational Gaussian process classification," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015, pp. 351–360.

[14] T. P. Minka, "Expectation propagation and approximate Bayesian inference," in *Proceedings of the 17th Conference in Uncertainty in Artifical Intelligence*, 2001, pp. 362–369.

[15] M. Kuss and C. E. Rasmussen, "Assessing approximate inference for binary Gaussian process classification," *Journal of Machine Learning Research*, vol. 6, pp. 1679–1704, 2005.

[16] ——, "Assessing approximations for Gaussian process classification," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, P. B. Schölkopf, and J. C. Platt, Eds., 2006, pp. 699–706.

[17] C. Villacampa-Calvo and D. Hernández-Lobato, "Scalable multi-class Gaussian process classification using expectation propagation," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017, pp. 3550–3559.

[18] A. Vehtari and et al, "Expectation propagation as a way of life: a framework for Bayesian inference on partitioned data," *available in arXiv*, 2018. [Online]. Available: https://arxiv.org/abs/1412.4869

[19] P. Jylänki, J. Vanhatalo, and A. Vehtari, "Robust Gaussian process regression with a Student-t likelihood," *Journal of Machine Learning Research*, vol. 12, pp. 3227–3257, 2011.

[20] A. F. García-Fernández, L. Svensson, M. R. Morelande, and S. Särkkä, "Posterior linearization filter: principles and implementation using sigma points," *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5561–5573, Oct. 2015.

[21] F. Tronarp, A. F. García-Fernández, and S. Särkkä, "Iterative filtering and smoothing in non-linear and non-Gaussian systems using conditional moments," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 408–412, March 2018.

[22] I. Arasaratnam, S. Haykin, and R. Elliott, "Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, May 2007.

[23] B. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice-Hall, 1979.

[24] I. Arasaratnam and S. Haykin, "Square-root quadrature Kalman filtering," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2589–2593, June 2008.

[25] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[26] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[27] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[28] V. Tolvanen, P. Jylänki, and A. Vehtari, "Expectation propagation for nonstationary heteroscedastic Gaussian process regression," in *IEEE International Workshop on Machine Learning for Signal Processing*, Sept. 2014, pp. 1–6.

Supplementary material of "Gaussian process classification using posterior linearisation"

In this supplementary material, we first provide the expressions for the moments required in SLR for the probit, logit, and noisy-threshold likelihood. We then analyse a synthetic example where EP, without ad-hoc fixes, fails.

SLR for probit, logit and noisy-threshold likelihoods

In this section, we provide the expressions for the SLR moments required in (15)-(16) for the probit, logit, and noisy-threshold likelihoods. For the probit model, we have

$$\mathrm{E}_{f_i}\left[\mathrm{E}\left[y_i|f_i\right]\right]=2\alpha-1,\ \mathrm{C}_{f_i}\left[\mathrm{E}\left[y_i|f_i\right]\right]+\mathrm{E}_{f_i}\left[\mathrm{C}\left[y_i|f_i\right]\right]=1-\alpha^2,$$

$$\mathrm{C}_{f_i}\left[f_i,\mathrm{E}\left[y_i|f_i\right]\right]=\frac{2P_i}{\sqrt{1+P_i}}\mathcal{N}\left(\frac{\overline{f}_i}{\sqrt{1+P_i}};0,1\right),$$

where $\alpha=\Phi\left(\overline{f}_i/\sqrt{1+P_i}\right)$. We have used Eq. (3.84) in [2] to obtain the last expression.

For the noisy threshold model, we have

$$\begin{aligned}\mathrm{E}_{f_i}\left[\mathrm{E}\left[y_i|f_i\right]\right]&=2\beta-1,\\ \mathrm{C}_{f_i}\left[\mathrm{E}\left[y_i|f_i\right]\right]+\mathrm{E}_{f_i}\left[\mathrm{C}\left[y_i|f_i\right]\right]&=1-\beta^2,\\ \mathrm{C}_{f_i}\left[f_i,\mathrm{E}\left[y_i|f_i\right]\right]&=2\left(1-2\epsilon\right)\sqrt{P_i}\\ &\quad\times\mathcal{N}\left(\frac{\overline{f}_i}{\sqrt{P_i}};0,1\right),\end{aligned}$$

where $\beta=\left[1-\Phi\left(\overline{f}_i/\sqrt{P_i}\right)\right]\epsilon+\left(1-\epsilon\right)\Phi\left(\overline{f}_i/\sqrt{P_i}\right)$.

For the logit model, the conditional moments are

$$\mathrm{E}\left[y_i|f_i\right]=\frac{1-\exp\left(-f_i\right)}{1+\exp\left(-f_i\right)},\ \mathrm{C}\left[y_i|f_i\right]=1-\left(\frac{1-\exp\left(-f_i\right)}{1+\exp\left(-f_i\right)}\right)^2.$$

The integrals $\mathrm{E}_{f_i}\left[\cdot\right]$ and $\mathrm{C}_{f_i}\left[\cdot\right]$ of these moments that are required in (15)-(16) do not have closed-form expressions, but they can be approximated using Gaussian quadrature rules/sigma-points [26], [27].

Synthetic example where EP fails

We consider a Gaussian prior over (f_1,f_2) with mean $(-0.5,-3)$ and unit variances for both variables with correlation 0.8. We also consider the noisy threshold likelihood, see (3), with $\epsilon=0.01$, $y_1=1$, and $y_2=1$.

We run EP first on the first variable and then on the second variable. After the first round of updates, the cavity distribution over f_1 has a variance of -117.9, which stops the EP iterations. If we process the measurements in the other order, we face the same problem. In this case, the variance of the cavity distribution over f_1 is -14.3. If we run EP in parallel [28], after the second update, the cavity distribution over f_1 also has a variance of -117.9. These negative variances are not due to numerical errors, they are the result of the EP iterations. In particular, the problematic part is the variance of the second variable, which increases after its update, so the likelihood approximation has a negative-definite variance. This creates problems at the next step of the iteration [19].

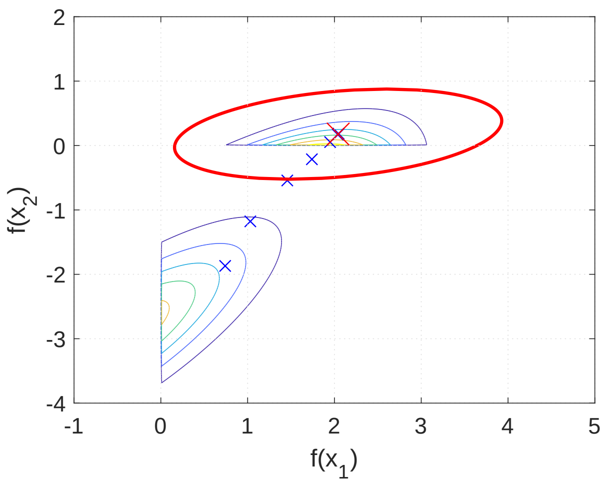

Figure 1: Contour plot of the posterior (Section V). PL posterior approximation: mean (red-cross) and the 3-σ ellipse (red line). Blue crosses represent the sequence of means of parallel PL. The PL solution matches the mode with highest density: EP does not provide a solution.

We show the contour plot of the posterior, which has been obtained using a fine grid of points, and the PL solution in Figure 1(a). Both parallel and sequential implementations of PL converge to the same solution and they match the mode of the posterior with highest density, which is a reasonable Gaussian approximation to a bimodal density. In the figure, we can also see the sequence of posterior means provided by the parallel PL iterations. In 6 iterations, the algorithm gets quite close to the fixed point. Laplace approximation would not change the prior as the gradient of the likelihood is zero almost everywhere. This example demonstrates that this type of situation can be satisfactorily handled with PL, but not with EP, without fixes, and Laplace methods.