Bayesian Multitask Classification with Gaussian Process Priors

Grigorios Skolidis and Guido Sanguinetti

***Abstract*— We present a novel approach to multitask learning in classification problems based on Gaussian process (GP) classification. The method extends previous work on multitask GP regression, constraining the overall covariance (across tasks and data points) to factorize as a Kronecker product. Fully Bayesian inference is possible but time consuming using sampling techniques. We propose approximations based on the popular variational Bayes and expectation propagation frameworks, showing that they both achieve excellent accuracy when compared to Gibbs sampling, in a fraction of time. We present results on a toy dataset and two real datasets, showing improved performance against the baseline results obtained by learning each task independently. We also compare with a recently proposed state-of-the-art approach based on support vector machines, obtaining comparable or better results.**

***Index Terms*— Bayesian inference, classification, Gaussian processes, multitask learning.**

I. Introduction

MULTITASK learning [1] has been a subject of intense research in the machine learning community in recent years. The frequent occurrence of multiple related learning tasks in real-world problems has motivated the development of machine learning approaches capable of capturing the similarity between tasks and hence leverage any information transfer between them [2]–[7]. From a motivational point of view, multitask learning can be particularly useful in situations where a limited amount of data is available in each task while data from many tasks are readily at hand. For example, predictions in the biomedical field suffer from a massive sample heterogeneity problem: samples from many individuals are often available, but the underlying distribution of the data from each individual may be different, so that simply pooling all data together may be inappropriate. Intuitively, a multitask approach should be able to avoid this problem by retaining the similarities between the different samples while taking into account the differences between them. Although it is difficult to specify conditions under which a multitask approach guarantees an increase in performance [8]–[11], empirically it has been shown in many examples that transfer learning does indeed happen, leading to improved results over models trained on the individual tasks.

From a Bayesian perspective, multitask learning can be simply implemented in a hierarchical fashion, whereby different models trained on separate tasks are joined by placing a common prior distribution over the model parameters [2], [6], [12], [13]. While these approaches can effectively capture global similarities between tasks, it limits the influence of the multitask setting to determining some common hyperparameters, thereby providing a relatively inflexible model. An attractive alternative is to model directly the correlations between tasks via a task covariance which can be learned from the data. This approach was recently proposed by [14] in the framework of Gaussian process (GP) regression, where the overall covariance structure of the data was specified as a Kronecker product of an input-covariance function and a task correlation matrix.

In this paper, we extend the results of [14] to the classification scenario by using a probit noise model as the output likelihood. The introduction of the nonlinear likelihood makes exact inference impossible, leading to the need for approximating techniques. While many of these techniques are by now part of the standard machine learning repertoire, there are a number of important classification problems where a multitask approach could give significant advantages. We first present a fully Bayesian treatment of the model, where Gibbs sampling is used to estimate the joint posterior distribution over parameters and latent variables. To our knowledge, a fully Bayesian estimation approach in this type of models has not been presented before. While sampling in general benefits from an ease of implementation, in the multitask GP scenario the implementation of a Gibbs sampler is nontrivial, due to the large amount of missing data. The technical details of how we implement the sampler are in themselves instructive about the structure of the model we propose, and are given in the appendix. While sampling leads to asymptotically exact inference, its computational burden prevents application in many real cases. We therefore propose two approximate inference approaches obtained by adapting the popular variational Bayes (VB) and expectation propagation (EP) frameworks to the multitask GP classification model. Experiments on a toy dataset show that both approaches provide an excellent approximation to the true posterior distribution. We complement this methodological component with an extensive comparative experimental part. Here, we test our methodology on three datasets, and compare it against three other methods. All experiments are performed multiple times, providing an

Manuscript received October 20, 2010; revised September 3, 2011; accepted September 4, 2011. Date of publication October 10, 2011; date of current version December 1, 2011. This work was supported in part by the Engineering and Physical Sciences Research Council under Grant EP/F009461/.

The authors are with the School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, U.K. (e-mail: G.skolidis@sms.ed.ac.uk; gsanguin@inf.ed.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2011.2168568

empirical assessment of both the average performance of the algorithms on each dataset, but also the variability of the predictions as the training set is changed.

The rest of this paper is organized as follows. In the next section, we briefly review multitask GP regression and introduce our classification model. We then discuss the three inference approaches considered. Gibbs sampling, EP, and VB. We then discuss the issue of transfer learning in our model, i.e., how information can flow from one task to another during the learning process. We then present results on three datasets, a synthetic benchmark problem used previously in multitask classification [15] as well as two real datasets. We conclude by discussing related approaches and the relative merits of our work, as well as its possible extensions.

II. Multitask GP

A. Multitask Regression

GPs are a popular nonparametric Bayesian tool for regression and classification, at the core of GP prediction is the *covariance function* or kernel, capturing the output covariance at different pairs of input points. In the regression case, observations are generally assumed to be noise-corrupted versions of an underlying stochastic process f depending on the input variable x

$$y_j \sim \mathcal{N}\left(f_j, \sigma^2\right)$$
$$\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, K)$$

where $\boldsymbol{\mu}(x)$ is the mean function and $K_{ij} = k(x_i, x_j)$ is the covariance function, capturing the input dependence of the target statistics.

In the multitask setting, we are interested in learning M related functions f_j for $j = 1, \ldots, M$, from training data $x_{ji}, y_{ji},\ i = 1, \ldots, n_j$, with $x \in \mathbb{R}^d$, and $n_1 + \cdots + n_M = N$. As usual in GP regression, we assume the following noise model:

$$y_{ji} = f_j(x_{ji}) + \epsilon_j, \text{ with } \epsilon_j \sim \mathcal{N}(0, \sigma_j^2) \tag{1}$$

where y_{ji} (x_{ji}) denotes the ith output (input) of the jth task. Let us consider the vector $\mathbf{y}$ of *complete responses* obtained by stacking the response in *all* tasks to each input point, such that $\mathbf{y} = \text{vec}(Y^T)$. Of course, in most applications not all entries of this vector will be observed, given the probabilistic nature of GPs, it is straightforward to treat the missing values. Let $\mathbf{f}$ be the latent function values corresponding to the complete responses, again stacked in a single vector ($\mathbf{f} = \text{vec}(F^T)$). Bonilla *et al.* [14] encapsulate the multitask regression problem by selecting the following form for the GP prior over the latent functions:

$$p(\mathbf{f}|X) \sim \mathcal{GP}(0, K^t \otimes K^x) \tag{2}$$

where $\otimes$ is the Kronecker product, and $K^t_{M\times M}$ and $K^x_{N\times N}$ are the task and data covariance matrix, respectively. Of central importance is the task covariance matrix, which can either be defined by a task covariance function $\mathbf{k}^t(.,.)$ when task-descriptor features x^t are available, or be a free form covariance matrix, specifying intertask correlations. Note that expressing the covariance function of the prior as a Kronecker product has been known in the geostatistics community as the "linear model of coregionalization" [16].

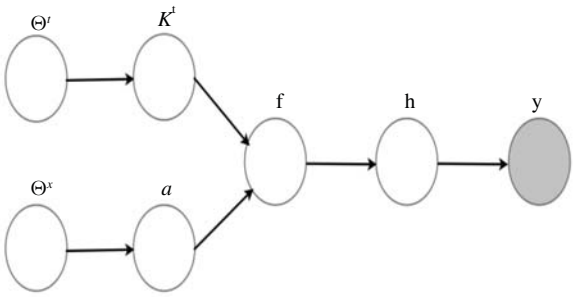

Fig. 1. General multitask probit model. Variables $\mathbf{y}$ represent the outputs, variables $\mathbf{h}$ and $\mathbf{f}$ represent the auxiliary and the latent function variables over the tasks, respectively. Variables K^t, α, θ^t, and θ^x are used to denote prior distributions of parameters of the task and data covariance function (see Section II-B for more details).

The noise model becomes $p(\mathbf{y}|\mathbf{f}) \sim \mathcal{N}(\mathbf{f}, D \otimes I)$, with $D_{M\times M}$ diagonal with $D_{jj} = \sigma_j^2$, and $I_{N\times N}$. The predictive distribution for the jth task $p(y_*|\ \mathbf{y})$ on a new point $\mathbf{x}_*$ is defined by

$$p(y_{*j}|\ \mathbf{y}) \sim \mathcal{N}(m_{y_{*j}|\mathbf{y}}, \Sigma_{y_{*j}|\mathbf{y}}) \tag{3}$$

with

$$\Sigma_{y_{*j}|\mathbf{y}} = \lambda_{**} - \lambda^T \Sigma^{-1} \lambda \tag{4}$$
$$m_{y_{*j}|\mathbf{y}} = \lambda^T \Sigma^{-1} \mathbf{y} \tag{5}$$

where $\lambda = k^t_j \otimes \mathbf{k}^x_{\mathbf{x},x_*}$, $\lambda_{**} = k^t_{jj} k^x_{x_*,x_*}$, and $\Sigma = K^t \otimes K^x + D \otimes I$, k^t_{jj} and k^t_j are the jth diagonal element and the jth column of K^t, respectively, $\mathbf{k}^x_{\mathbf{x},x_*}$ is the vector of covariances between the test point x_* and the training points $\mathbf{x}$, $\mathbf{k}^x_{x_*,x_*}$ is the computed variance between the test point itself, and K^x denotes the covariance matrix of all training points.

Hyperparameters θ^t and θ^x appear both in the task and data covariance functions, they can be estimated within this framework by maximizing the evidence or log marginal likelihood. This can be done either by standard gradient descent or using an E-M type algorithm exploiting the Kronecker factorization of the prior [14, Sec. 2.2].

B. Multitask GP Classification

A key asset of GP regression is the conjugacy of the Gaussian noise model (1) with the GP prior (2), enabling analytical marginalization of the latent variables. In the classification setting, the target values are discrete, and no such conjugacy is available. In the rest of this paper, we focus on binary classification employing the probit model [17], [18], generalization to the multiclass probit model or to other binary noise models is in principle straightforward.[1]

[1]We observe in passing that Bonilla *et al.* [14] also addressed a multiclass classification problem by treating it as a regression problem with Gaussian noise. While this may have been reasonable on the specific dataset considered, its general applicability as a classification method is questionable.

The graphical model of Fig. 1 illustrates the dependencies between the variables of a general multitask probit classification model. In the general case, the joint likelihood factorizes as

$$p(\mathbf{y}, \mathbf{h}, \mathbf{f}, K^t, \alpha|\theta^t, \theta^x) = p(\mathbf{y}|\mathbf{h})p(\mathbf{h}|\mathbf{f})p(\mathbf{f}|K^t, \alpha) \\ p(\alpha|\theta^x)p(K^t|\theta^t). \quad (6)$$

The relationship between outputs y and the auxiliary variable h is deterministic and given by

$$p(y_i|h_i) = \begin{cases} \delta(h_i)\delta(y_i) & \text{if } y_i = +1 \\ \delta(-h_i)\delta(-y_i) & \text{if } y_i = -1 \end{cases} \quad (7)$$

where δ is 1 if its argument is positive, zero otherwise. The auxiliary variable $\mathbf{h}$ is given a normal distribution with mean given by $\mathbf{f}$ and variance 1, leading to the probit model [19]. The latent variable $\mathbf{f}$ integrates the information coming from the data and the tasks through a matrix variate normal distribution with zero mean and covariance given by $K^t \otimes K^x$, as in [14]. The matrix K^t is the task covariance matrix encoding information about the tasks, while matrix K^x is the data covariance matrix encoding information coming from the data of each task. Parameters of the prior distribution over the task covariance matrix are denoted by θ^t. Parameters of the data covariance function $K^x(.,.)$ are denoted by α, with θ^x being the associated prior hyperparameters. In the rest of this paper, we will choose a free form for the task covariance matrix, while the data covariance matrix will be given by an automatic relevance determination (ARD) [20], i.e., a squared exponential covariance with diagonal matrix.

III. Inference in Multitask GP Classification

Coupling the probit likelihood with a GP prior on the latent functions f results in a non-Gaussian posterior distribution $p(\mathbf{f}|\mathbf{y})$, making exact inference impossible. In this section, we present three inference approaches to solve this problem. First we present a solution based on Gibbs sampling, this is asymptotically exact but computationally intensive. We then employ two deterministic methods to approximate the non-Gaussian posterior, the EP approximation [17], [20], [21], and a variation of the methods proposed in [17] and [18], based on probit regression with a variational expectation maximization (EM) algorithm. The hyperparameters of the task and data covariance function in the EP and variational probit regression models were estimated by type II maximum likelihood (ML) (point estimates) for computational efficiency. Incorporating hyperpriors in a variational GP-probit model is possible [18], and extension to the multitask setting should be straightforward.

A. Bayesian Inference: Gibbs Sampling

Markov chain Monte Carlo (MCMC) sampling methods are often employed in Bayesian models, as they provably sample from the true posterior distribution in the limit of infinite samples. The Gibbs sampling scheme is a particular type of Markov chain simulation where samples are drawn iteratively from the posterior of a subset of the variables conditioned on all others. For a complete treatment on sampling methods, the interested reader is referred to [22].

To complete the specification of our model (6), we need to define prior distributions over the parameters of the task and data covariance matrices. The hyperparameters α of the ARD data covariance are given a log-normal distribution to ensure positivity

$$p(\alpha|\theta^x) = \log\mathcal{N}\left(\alpha|\mu_x, \sigma_x^2\right)$$

where hyperparameters $\theta^x = \{\mu_x, \sigma_x^2\}$ are fixed to yield reasonably uninformative priors. The covariance matrix over the tasks K^t is given an inverse Wishart prior probability distribution to ensure conjugacy

$$K^t \sim \mathcal{IW}_M(\beta, \Lambda)$$

where β is the number of degrees of freedom, and Λ is an $M \times M$ positive definite matrix (the parameter matrix) [23].

Due to conjugacy, the conditional posterior distribution of $\mathbf{h}, \mathbf{f}, K^t$ can be found analytically. In contrast, no conjugate prior distribution is available for the ARD length scales α_i, and individual Metropolis–Hastings subsamplers must be employed to draw samples from their conditional posterior. Derivations of the conditional posteriors are given in the Appendix A. It is worth noting that the MCMC sampling scheme could further be accelerated by following recent advances in sampling for latent Gaussian models [24].

1) Missing Values Treatment: Consistency is an appealing feature of GPs, which enables missing values to be marginalized effortlessly. In the multitask situation, however, data are more structured, in that several output values correspond to a single input. Therefore, one also has to consider the situation when some, but not all, output values are missing, a situation which we could term *partially missing* data and which requires a little more care. A special but important case is when each input has a unique observed output, i.e., outputs are observed for only one task.

To obviate this problem, we define $\mathbf{f}^o$, the latent function values corresponding to the observed outputs, and introduce a *communication matrix* S relating total latent function values and observed latent function values

$$\mathbf{f}^o = S^T\mathbf{f} \quad (8)$$

which simply eliminates the unobserved outputs (which do not figure in the likelihood terms). As the operation given by S is linear, the resulting variable $\mathbf{f}^o$ is still a GP, with covariance that can be easily identified from the prior covariance. Full details are given in Appendix B.

B. Approximate Inference

1) EP Approximation: In this section we outline the EP approximation for multitask GP classification, for a complete treatment of EP for the conventional GP classifier, see [20]. In this case, we will marginalize the auxiliary variable h and work directly with the likelihood given by $p(y|f) = \Phi(yf) = \int_{-\infty}^{yf}\mathcal{N}(0,1)$. Hyperparameters of the input and task covariance matrix are given point estimates via type II ML, so that nodes K^t and α are treated as parameters and not

as random variables. To harmonize the notation, we will now denote these parameters as θ^x and θ^t.

The posterior over the latent variables, $p(\mathbf{f}|\mathbf{y})$, is proportional to a product of the prior and the factorized non-Gaussian likelihoods. EP approximates these non-Gaussian likelihoods by a product of unnormalized Gaussians $t_i(f_i)$

$$\prod_{i=1}^{N} p(y_i|f_i) \simeq \prod_{i=1}^{N} t_i(f_i|\tilde{Z}_i, \tilde{\mu}_i, \tilde{\sigma}_i^2) = \mathcal{N}(\tilde{\boldsymbol{\mu}}, \tilde{\Sigma}) \prod_{i=1}^{N} \tilde{Z}_i \quad (9)$$

where $\tilde{\boldsymbol{\mu}}$ is a vector of $\tilde{\mu}_i$ and $\tilde{\Sigma}$ is diagonal with $\tilde{\Sigma}_{ii} = \tilde{\sigma}_i^2$. This leads to the approximated distribution of the latent variables $q(\mathbf{f}|\mathbf{y})$

$$q(\mathbf{f}|\mathbf{y}) = \frac{1}{Z_{EP}} p(\mathbf{f}) \prod_{i=1}^{n} t_i\left(f_i|\tilde{Z}_i, \tilde{\mu}_i, \tilde{\sigma}_i^2\right) = \mathcal{N}(\mu, \Sigma) \quad (10)$$

where $\mu = \Sigma\tilde{\Sigma}^{-1}\tilde{\mu}$, and $\Sigma = [(K^t \otimes K^x)^{-1} + \tilde{\Sigma}^{-1}]^{-1}$.

EP updates the individual scaled functions t_i sequentially by termwise refinement. At each step, the current t_i is left out, giving rise to the cavity distribution $q_{-i}(f_i)$ which is then combined with the true likelihood $p(y_i|f_i)$, to give a non-Gaussian distribution. A Gaussian approximation is chosen to approximate the non-Gaussian distribution from the previous step, which is then used to compute the parameters of t_i by moment-matching between the two distributions. Hyperparameters can be estimated by approximate type II ML using the EP log marginal likelihood. The gradients with respect to hyperparameters θ_x and θ_t are given by

$$\frac{\partial log Z_{EP}}{\partial \theta_j} = \frac{1}{2}\tilde{\mu}^T(K^t \otimes K^x + \tilde{\Sigma})^{-1}\Omega(K^t \otimes K^x + \tilde{\Sigma})^{-1}\tilde{\mu} - \frac{1}{2}\mathrm{tr}\left((K^t \otimes K^x + \tilde{\Sigma})^{-1}\Omega\right) \quad (11)$$

and

$$\Omega = \begin{cases} K^t \otimes \frac{\partial K^x}{\partial \theta^j} & \text{if } \theta^j = \theta^x \\ \frac{\partial K^t}{\partial \theta^j} \otimes K^x & \text{if } \theta^j = \theta^t. \end{cases} \quad (12)$$

2) Variational Probit Regression: We now employ a variational approximation in the spirit of [18]. We refer to the model in Fig. 1 where again hyperparameters are estimated via type II ML. A variational treatment of this problem involves approximating posteriors over latent variables in an ensemble of factored posteriors of the form

$$p(\Theta|\mathbf{y}, X, t, \theta^t, \theta^x) \approx \prod_{i=1} Q(\Theta_i) = Q(\mathbf{f})Q(\mathbf{h}). \quad (13)$$

It can be shown that the lower bound on the log marginal likelihood $\log\ p(\mathbf{y}|X, \theta^t, \theta^x) \geq \mathbb{E}_{Q(\Theta}\{\log\ p(\mathbf{y}, \mathbf{f}, \mathbf{h}|X)\} - \mathbb{E}_{Q(\Theta}\{\log\ Q(\Theta)\}$ is maximized by distributions of an unnormalized form

$$Q(\Theta_i) \propto \exp\left(\mathbb{E}_{Q(\Theta\setminus\Theta_i)}\{\log\ p(y, \Theta|X, \theta^t, \theta^x)\}\right) \quad (14)$$

where $Q(\Theta \setminus \Theta_i)$ denotes the factorized distribution with the ith component removed.

A variational EM algorithm is derived in which in the E-step the expectations of the variational parameters are computed, while in the M-step the hyperparameters θ^t, θ^x are optimized given the expectations computed in the previous step. The lower bound on the log marginal likelihood is given by

$$\mathcal{L}(Q) = \mathbb{E}_{Q(\Theta)}[\log\ p(\mathbf{y}, \mathbf{h}, \mathbf{f}|X, \theta^t, \theta^x)] \quad \mathbb{E}_{Q(\Theta)}[\log\ Q(\mathbf{h})Q(\mathbf{f})] \quad (15)$$

$$= \sum_{i=1}^{N} \log z_i - \frac{1}{2}\log|I + K^t \otimes K^x| - \frac{1}{2}\tilde{\mathbf{f}}^T(K^t \otimes K^x)^{-1}\tilde{\mathbf{f}} \quad (16)$$

where z_i is the normalization constant of approximating distribution $Q(\mathbf{h})$, given by $z_i = \Phi(y_i\tilde{f}_i)$.

1) *E-Step:* Compute the sufficient statistics for each variational parameter $Q(\Theta_i)$ given by (14), based on current $\theta^t,\ \theta^x$

$$Q(\mathbf{f}) \propto \exp\left\{\mathbb{E}\{\log\ p(\mathbf{h}|\mathbf{f}) + \log\ p(\mathbf{f}|X)\}\right\} = \mathcal{N}_{\mathbf{f}}(\tilde{\mathbf{f}}, \Sigma) \quad (17)$$

$$Q(\mathbf{h}) \propto \exp\left\{\mathbb{E}\{\log\ p(\mathbf{y}|\mathbf{h}) + \log\ p(\mathbf{h}|\mathbf{f})\}\right\} = \prod_{i=1}^{N} \frac{\mathcal{N}_{h_i}(\tilde{f}_i, 1)}{\Phi(y_i\tilde{f}_i)} \quad (18)$$

where $\tilde{\mathbf{f}} = \Sigma\tilde{\mathbf{h}}$, and $\Sigma = K^t \otimes K^x(I + K^t \otimes K^x)^{-1}$.

2) *M-Step:* Optimize θ^t, θ^x based on the last E-step.

$$\frac{\partial\mathcal{L}(Q)}{\partial\theta^j} = -\frac{1}{2}\mathrm{trace}((I + K^t \otimes K^x)^{-1}\Omega) + \frac{1}{2}\tilde{\mathbf{f}}^T(K^t \otimes K^x)^{-1}\Omega(K^t \otimes K^x)^{-1}\tilde{\mathbf{f}} \quad (19)$$

where Ω is given in (12), as in the EP approximation.

IV. Transfer of Knowledge and Covariance Structure

A. Task Covariance Matrix

Learning the covariance structure of the tasks is performed either by drawing samples as in Section III-A from the posterior of the task covariance matrix or through optimization of the hyperparameters as in Sections III-B.1 and III-B.2, and is an important component of the method proposed here.

The task covariance matrix can either come from a covariance function, when task descriptor features are available, a case investigated by [3] and [25] in different contexts, or it can have a free form as in [14]. In the case when a free form covariance matrix is used, K^t acts as a correlation matrix capturing the dependencies between the tasks. Despite the "free-form" structure of K^t, positive definiteness restrictions must be retained. Bonilla *et al.* [14] achieved positive semidefinite guarantees by parameterizing the lower triangular matrix L of the Cholesky decomposition $K^t = LL^T$.

If we want to restrict the task covariance matrix to be a *correlation* matrix of the tasks, except from the symmetric and positive definite restrictions, it has to have a unit diagonal. This restriction, easy as it may seem, poses certain difficulties and many attempts have been made to solve this constrained optimization problem. A solution to this problem is presented in [26], where the correlation matrix is still decomposed as

$K^t = BB^T$, with B $(M \times M)$, and to view each row vector of B as coordinates lying in the unit hypersphere. If we denote by b_{ij} the elements of the matrix B, then these $M \times M$ coordinates are obtained from $M \times (M-1)$ angular coordinates θ_{ij}^t by

$$b_{ij} = \begin{cases} \cos\theta_{ij} \prod_{k=1}^{j-1} \sin\theta_{ik} & \text{for} \quad j = 1, \ldots, M-1 \\ \prod_{k=1}^{j-1} \sin\theta_{ik} & \text{for} \quad j = M. \end{cases}$$

A possible drawback of this setup would be that the number of correlation parameters θ_t that need to be estimated is given by $M^2 - M$, which grows quadratically with the number of tasks M. It is interesting to note that, if the resulting off-diagonal elements of K^t are different from zero, then samples from one task will affect predictions of the other tasks, as the mean of the predictive distribution of the jth task, given by

$$\mathbb{E}\left[p(y_{j*}|\mathbf{y})\right] = (\mathbf{k}_j^t \otimes \mathbf{k}_{\mathbf{x},x_*}^x)^T \Sigma^{-1} \mathbf{y} \tag{20}$$

is computed by weighting observations from task i, where $i \neq j$, by the ith element of $\mathbf{k}_j^t$. Similar regularization is performed in the computation of the variance of the predictive distribution. Therefore, the task covariance matrix acts as a transfer of knowledge between tasks.

B. Input Covariance Function

The choice of the covariance function of the inputs in single task training depends on the task of interest, and even similar tasks can be trained with different covariance functions. In this specific multitask scenario that we are adopting, the covariance function and, hence, the hyperparameters are shared amongst the tasks. While this reduces somewhat the flexibility of the model, performing the optimization of the hyperparameters using data from all tasks results in a regularized optimal solution, which can be seen as bias selection [8]. Furthermore, if the ARD covariance function [27] is used, then an optimal subset of features is estimated, which is shared across the tasks.

Thus the transfer of knowledge between tasks in this method is the result of two sources of information. The first is through the task covariance matrix, whose operation on the input covariance matrix results in samples from one task affecting predictions of other tasks. The second stems from the optimization of the shared hyperparameters of the input covariance function, as a task-regularized optimal solution.

V. Experimental Results

Evaluation of the proposed multitask framework is performed on a synthetic dataset and two real datasets. Throughout all of the experiments, we use the ARD covariance function [20] for the input covariance K^x, and a free-form correlation matrix [26] for the task covariance matrix K^t. We report results obtained both with a VB and EP[2] approximation to the posterior distribution, except in one of the real datasets, where the VB algorithm proved very slow because of the size of the dataset. Numerical inaccuracies in the EP approximation were corrected by using Rasmussen's solve_chol.m[3] function, which solves linear equations from the Cholesky factorization, while Rasmussen's excellent optimization routine minimize.m[4] was used in some of the simulations for the optimization of the hyperparameters. We also perform a fully Bayesian analysis on the synthetic dataset, although the high computational costs prevented application to the real datasets.

In order to check that there is transfer of knowledge, so that performance is improved when tasks are trained together compared to in isolation, the multitask GP is compared with the standard single-task GP probit classifier. For the single-task GP classifier we use the EP approximation with an ARD covariance function.

In addition, on all datasets we compare with two alternative multitask learning approaches. One is an alternative GP-based approach where the various tasks are coupled only through sharing the hyperparameters of the data covariance function k^x. This will be termed as inaudible noise distortion (IND)-multiconductor transmission line (MTL) since in this form of multitask learning there are no correlations between the latent functions of each task. This is in some sense a weaker form of multitask learning, and is essentially the one used in [12], inspired by well-established hierarchical Bayesian modeling paradigms. The second is a state-of-the-art multitask classifier based on support vector machines[5] (SVM) [4], which tackles the problem of multitask SVM feature and kernel selection, based on the maximum entropy discrimination framework. This performs nonlinear classification by mapping input data x implicitly into Hilbert space through a feature map ϕ. In this case, there are still M discriminant functions, one for each task, which however share a kernel selection vector s, which controls the weight of each base kernel. In all experiments a large number of kernels (10 on the toy data, 18 on the two real datasets) were inserted into the algorithm, which would then automatically decide which kernels are more relevant for classification. Throughout all experiments, the regularization parameters α (kernel selection variable) and C (threshold) (see [4] for details) were determined through cross-validation.

The experimental setup is identical for all datasets, and follows a similar approach to [15]. It is known that the merits of multitask compared to single-task learning are more apparent when few data per task are available. For this reason, in all experiments we show results starting with few data points per task and increasing up to a number where single-task learning approaches the performance of multitask. In order to estimate the dependence of the results on the training set, each experiment was repeated 100 times on independently sampled training sets of the same size. It is important to note that in the case of single-task learning each dataset from each task is learnt in isolation, which results in more models being trained.

To assess the performance of the various algorithms, we considered the receiver operator characteristic (ROC) curves, which plot *sensitivity* versus 1-*specificity* of the classifier for

[2]MATLAB code to allow replication of the experiments available at http://homepages.inf.ed.ac.uk/gsanguin/software.html.

[3]Available at http://www.gaussianprocess.org/gpml/code/matlab/doc.

[4]Available at http://www.kyb.tuebingen.mpg.de/bs/people/carl/code.

[5]Available at http://www.cs.columbia.edu/jebara/code/multisparse.

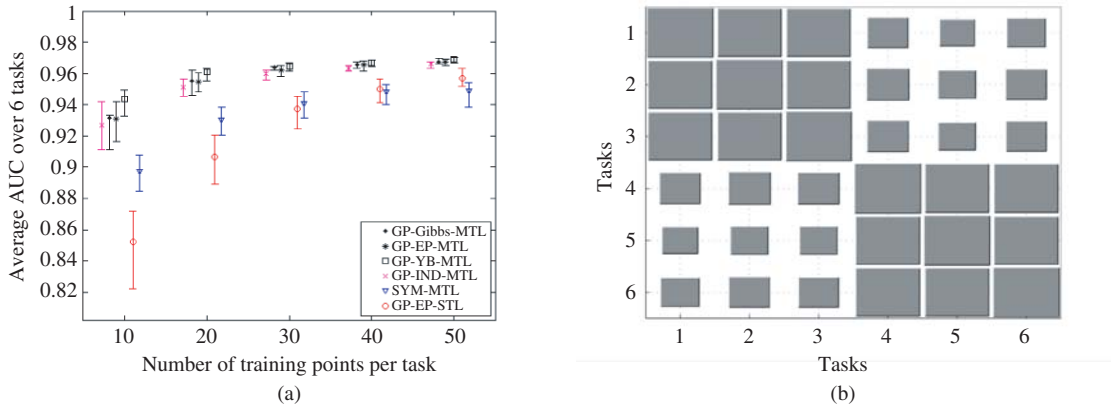

Fig. 2. Experimental results on the toy dataset. (a) Mean AUC over the six tasks. (b) Hinton diagram of the task covariance matrix, averaged over the 100 independent trials of the results obtained by EP with 50 data points per task.

varying values of the threshold posterior probabilities (bias in the SVM case). The performance measure used to assess the different algorithms is the area under the curve (AUC) [28], as it provides a measure that depends on the positive and negative predictive value (PPV and NPV), and not just the overall accuracy. Assessment of the different algorithms is then performed by plotting the median of the AUC over the 100 runs of the experiments and the resulting confidence intervals for varying sizes of the training set. Statistical significance of the results was assessed with standard t-test.

While the purpose of this paper is to propose a model, rather than optimize its running time, it is still important, from the practitioner's perspective, to assess the relative speed of the various algorithms proposed. Naturally, single-task learning and the multitask induced by hyperparameter sharing are faster as they consider a simpler covariance structure. Among the approximate inference approaches to multitask GP, in our experiments EP was consistently faster than both Gibbs sampling and the variational approximation (this depends on the number of VB iterations, of course). As a yardstick, EP took approximately 1 h to run on the Arrhythmia dataset with seven tasks and 50 training points per task. As a comparison, the SVM-based method took roughly 40 min considering the cross-validation for the determination of the model parameters.

A. Synthetic Dataset

The synthetic dataset is a toy dataset previously used in [15]. It is comprised of six binary classification tasks. Data for the first three tasks is generated from a mixture of two partially overlapping Gaussian distributions, and similarly for the remaining three tasks. Hence, the six tasks cluster in two groups, tasks within the same cluster essentially differ only in the noise applied. Given the size of the problem, a fully Bayesian treatment was possible, however, the number of repetitions was limited to 40 because of the long running times.

Fig. 2(a) shows the mean and the error bars of the AUC over the six tasks for Gibbs sampling, both approximations for multitask GP classification (EP & VB), multitask with independent GPs (IND-MTL), single-task GP learning, and for the SVM-based method. It is clear that all multitask learning algorithms significantly outperform single-task learning, reflecting the fact that indeed in this case there are important correlations between the tasks. It is also worth noticing that all multitask GP methods improve upon the SVM-based method (with a statistically significant improvement in 82% of the cases at 5% p-value). Larger differences in performance are noticed when few data points are used for training. As expected, those differences decay as the number of training points increases. Comparing the multitask GP methods, we see that VB performs slightly better than EP and IND in the case when few data points are available, although the differences are minimal when more than 10 points per task are available. It is worthwhile comparing the results to those reported in [15], which used a semisupervised multitask logistic regression model to perform classification. The reported AUC in that paper (average only) was of approximately 90% when 30 labeled data points per task were used, roughly 5% less than the performance achieved by our method. This is probably due to the greater flexibility of the GP as a classifier, compared to the logistic regression employed in [15].

Fig. 2(b) shows the Hinton diagram of the optimized task covariance matrix (larger blocks denote values closer to 1), showing that the algorithm is able to correctly learn the similarity between the tasks.

B. Arrhythmia Dataset

In this dataset, we are provided with seven recordings of electrocardiogram (ECG) signals from seven different patients. Each recording corresponds to a large number of heart beats, the goal is to classify each heart beat into two classes, either normal or premature ventricular contraction (PVC) arrhythmic beats. This problem was already considered using single-task GP classifiers in [29]. Data for this set of experiments were taken from the Massachusetts Institute of Technology-Boston's Beth Israel Hospital Arrhythmia database [30], each recording was sampled at 360 Hz. Annotation provided by the database was used to separate the beats before any preprocessing. Each beat segment, consisting of 360 data points (1 min), was transformed into the frequency domain using a fast Fourier transform with a Hanning window. Only the first 10 harmonics

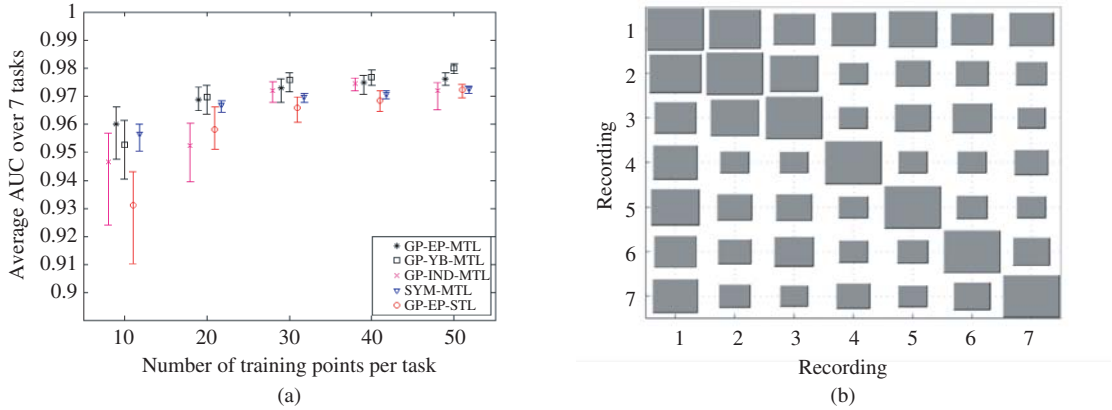

Fig. 3. Experimental results on the *Arrhythmia* dataset. (a) Mean AUC over the seven tasks. (b) Hinton diagram of the task covariance matrix, averaged over the 100 independent trials of the results obtained by EP with 50 data points per task.

TABLE I
DESCRIPTION OF THE ARRHYTHMIA DATASET

Task ID	1	2	3	4	5	6	7
Recording ID	106	200	203	217	221	223	233
Total number of data	2021	2567	2970	406	2349	2417	3053
Number of normal heart beats	1503	1740	2526	244	1954	1955	2224
Number of PVC heart beats	518	827	444	162	395	462	829

are used as features for classifying heart beats, as most of the information of the signal is contained in these harmonics. In this case, each recording/patient is treated as a separate task. A detailed description of the dataset is given in Table I.

The mean AUC over the 100 repetitions is shown in Fig. 3(a) for all five methods considered. It is noticed that the EP, VB, and the SVM methods significantly outperform the IND and the segmental transmission line (STL) method with small error bars. It is interesting to note that for 20 training data points per task STL performs better than the MTL with independent latent functions. Further investigation however reveals that there are two types of tasks in terms of performance. Some tasks (1, 2, 3, 4, 7) appear to be reasonably straightforward, achieving more than 95% of AUC for all multitask and single-task learning methods applied, and for all training set sizes considered. In contrast, tasks 5 and 6 are harder classification tasks, and in these cases the multitask approaches yield a considerable advantage, both in terms of performance (higher median area under receiver-operating characteristics) and in terms of robustness (smaller error bars). These results are given in Fig. 4(a) and (b), respectively. As in the synthetic example, this difference increases as the number of training points per task increases. The Hinton diagram of the task covariance matrix, in Fig. 3(b), shows the similarity between tasks. Not surprisingly, the results are not as clear-cut as in the toy dataset, and no conclusions can be made.

C. Landmine Dataset

The final set is a landmine detection problem, consisting of 19 tasks where each point is represented by nine features, previously investigated in [15] and [31]. This dataset is tested only with the EP approximation and the independent multitask GP setting, as the large number of tasks made the VB approach slower. Also, we consider at least 20 training points per task, given the larger number of parameters to be estimated in the task covariance matrix. Results[6] shown in Fig. 5(a) demonstrate a significant difference, in terms of performance, between the multitask learning algorithms with both GPs and SVMs over the single-task learning with GPs, while comparing the performance of the multitask algorithms it is noticed that the EP approximation outperforms the other two. A more accurate analysis of the results shows that the GP method significantly (at 5% p-value) outperforms the SVM method in the overwhelming majority of tasks (Table II). Liu *et al.* [15] reported a mean AUC of approximately 78% on this problem (irrespective of the training size, since their semisupervised approach benefited from using unlabeled data). Our approach achieves a slightly better performance when only 20 data points are available, and a significantly better performance as the size of the training set increases. Fig. 5(b) shows the Hinton diagram of the task covariance matrix, clearly showing the presence of two clusters containing the first 10 and the remaining 9 tasks. This is in good agreement with previously reported results on this dataset [15, Fig. 3(c)].

VI. RELATED WORK

Multitask learning, transfer learning, and link analysis are intimately connected techniques that use data from different tasks either to improve generalization performance or to

[6]As in the previous two examples, the reported median is over the 100 repetitions and not over the tasks.

TABLE II

t-TEST FOR THE LANDMINE DATASET. THE SECOND COLUMN (NO. OF TASKS) SHOWS THE NUMBER OF TASKS WITH BETTER MEANS, FOR THE METHOD STATED IN THE FIRST COLUMN. THE THIRD COLUMN (STATISTICALLY SIGNIFICANT) SHOWS THE NUMBER OF TASKS THAT WERE STATISTICALLY SIGNIFICANT ON THE 0.05 SIGNIFICANCE LEVEL (p-VALUE)

	No. of tasks					Statistically significant				
Data points per task	**20**	**40**	**60**	**80**	**100**	**20**	**40**	**60**	**80**	**100**
MTL-EP > STL-EP	19	19	19	19	18	19	19	19	19	18
MTL-EP > MTL-SVM	15	16	15	16	16	15	16	15	16	16

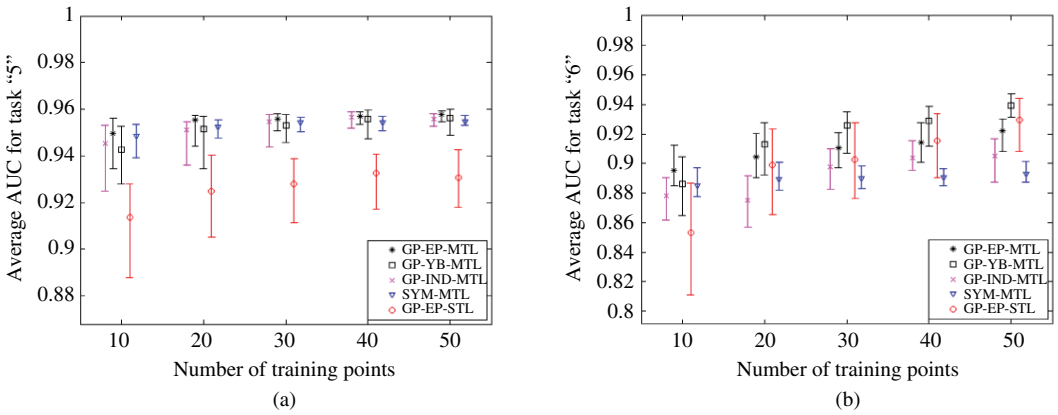

Fig. 4. Additional results on the *Arrhythmia* dataset. (a) AUC for task 5 (Recording ID: 221). (b) AUC for task 6 (Recording ID: 223).

model the relations between different entities [32]. The model proposed in [14], as well as investigated in the classification scenario in this paper, is closely related to the work of Yu *et al.* [25], which employs two kernel functions through the tensor product to model the dependencies between different set of entities. The properties of the model proposed in [14] and [25] was found attractive enough to motivate work in both directions, i.e., multitask learning and link analysis. This paper in [33] is inspired by [14], where they employ a t-noise model for the likelihood. Zhu *et al.* [34] demonstrate an algorithm based on MCMC able to handle very large datasets, in the context of stochastic relational models. Moreover, the work for link analysis in [35] in the context of GPs demonstrates intimate connections with other transfer learning algorithms.

Other approaches for multitask learning based on GPs include [2], [3], [6], [12], [13], [36]–[39]. Broadly speaking, these approaches can be grouped according to the level at which the multitask assumption is enforced. At the highest level, [39] propose to view the different tasks as convolution of a single lower dimensional GP with a deterministic process (extending the ideas introduced in [40] in the context of transcriptional regulation). A similar line is taken by [38], where tasks are assumed to share an underlying single GP, with each task being characterized by its own correlated noise (given by a task-specific GP). Naturally, this enforces a very strong version of multitask learning, as the different tasks can deterministically be reduced to a single stochastic process plus noise. Somewhat related, this paper in [36] presents a semiparametric model based on GPs to tackle the problem of multiple responses, which is a form of multitask learning. In this paper, each response variable is modeled as a GP whose covariance function is an outcome of a linear mixture of base kernels. On the other hand, [2], [6], [12], [13], [37] join the tasks by placing a common prior distribution over the model parameters. In [2] and [13], the learning of the covariance matrix is performed in a nonparametric fashion, Yu *et al.* [6] presents a multitask learning algorithm with t-processes as a more robust alternative of [13] and [2] against outlier tasks. Furthermore, Birlutiu *et al.* [37] presented an extension of the methods presented in [13] and [2] to multitask learning from qualitative preference statements. All of these methods enforce a weaker form of multitask learning, as the coupling is at the level of shared hyperparameters which simply control qualitative features of the latent stochastic process (e.g., its length scale). Our approach, in line with [3], takes a somewhat intermediate path. The various tasks have different underlying stochastic processes, which are forced to be correlated through the task covariance matrix. It is worth mentioning that the framework proposed here can straightforwardly handle multiple response problems found in many areas of research such as image processing [41] or bioinformatics [42], reflecting the usefulness of the method and possible avenues of application.

Although multitask learning was originally intended to improve statistical and computational efficiency, most of this paper presented until now concentrates on statistical efficiency. Our approach, as all GP approaches, suffers from cubic scaling in the number of data points.[7] Therefore, to

[7]In general, this is true only if there is a single observed response per input. In the worst case of fully observed responses, the algorithm would scale cubically in the product of number of tasks and number of data points. The number of tasks however is generally much smaller than the number of data points.

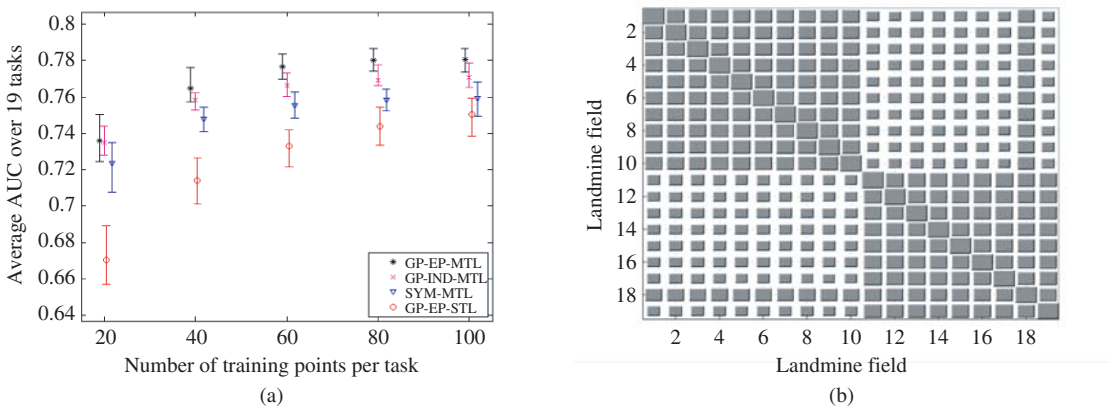

Fig. 5. Experimental results on the *landmine* dataset. (a) Mean AUC over the 19 tasks. (b) Hinton diagram of the task covariance matrix, averaged over the 100 independent trials of the results obtained by EP with 100 data points per task.

be able to handle very large datasets, it would be advisable to couple the methodology with a sparsification technique, as in [39].

VII. Conclusion

In this paper, we presented a Bayesian multitask classification model based on GP priors. This paper extends the regression model of Bonilla *et al.* [14] to the classification scenario, introducing a number of novel aspects. First of all, we considered non-Gaussian likelihoods in order to address classification problems. Secondly, we presented a number of inference strategies, including an asymptotically exact sampling scheme and two efficient deterministic approximations. Finally, we discussed in detail the treatment of partially missing data, which is important when posterior estimation of the hyperparameters is required (as opposed to simple type II ML). We demonstrated on a number of important real-world tasks the benefits of this approach, both over single-task learning and competing multitask approaches.

This paper can be extended in several directions. Multiclass classification problems using a multivariate probit model are easy to accommodate within our model, particularly using a variational approximation [18]. A natural extension of the multitask learning framework proposed in this paper would be to consider unlabelled data, in a semisupervised MTL setting, to further improve predictions, the multitask model of [14] with the recently proposed model for Bayesian semisupervised learning of [43] is a promising route to follow. Also, efficient learning strategies employing sparsification methods could be considered along the lines of [39]. Perhaps a more interesting extension would be to consider the meta-learning scenario, where some tasks have no associated training data [8]. While this can be addressed if a parametric form of task covariance function is employed, it is unclear whether such an assumption would have the required flexibility. It would be interesting to explore nonparametric approaches to achieve meta-z when a free form task covariance matrix is employed.

Appendix A
Gibbs Sampler

The posterior of the auxiliary variables $\mathbf{h}$ conditioned on all other variables is given by

$$p(\mathbf{h}|\mathbf{f},\mathbf{y}) = \prod_{n=1}^{N}\left(f_n + y_n \frac{\mathcal{N}_{h_n}(f_n,1)}{\Phi(y_n f_n)}\right) \tag{21}$$

resulting in a product of truncated univariate Gaussians. Continuing the posterior of the latent variables $\mathbf{f}$ as follows:

$$p(\mathbf{f}|\mathbf{h},K^t,\alpha) = \mathcal{N}_{MN}(\Sigma\mathbf{h},\Sigma) \tag{22}$$

where

$$\Sigma = \left(I + \left(K^t \otimes K^x\right)^{-1}\right)^{-1} = \left(K^t \otimes K^x\right)\left(I + K^t \otimes K^x\right)^{-1}.$$

Individual MH subsamplers are employed to sample from the full conditional distribution of the hyper-parameters α as

$$\begin{aligned} p(\alpha|\mathbf{f},\theta^x) &\propto p(\mathbf{f}|\alpha,K^t)p(\alpha|\theta^x) \\ &\propto \mathcal{N}_{MN}(0,K^t \otimes K^x_\alpha)\log\mathcal{N}(\alpha|\mu_x,\sigma_x^2). \end{aligned} \tag{23}$$

At each time step, a proposed sample α^* is accepted with probability $A(\alpha^*,\alpha)$, where

$$A(\alpha^*,\alpha) = \min\left(1, \frac{\mathcal{N}_{MN}(0,K^t \otimes K^x_{\alpha^*})\log\mathcal{N}(\alpha^*|\mu_x,\sigma_x^2)}{\mathcal{N}_{MN}(0,K^t \otimes K^x_{\alpha})\log\mathcal{N}(\alpha|\mu_x,\sigma_x^2)}\right). \tag{24}$$

The posterior of the task covariance matrix K^t is given by

$$p(K^t|\mathbf{f},\theta^t) = \frac{p(\mathbf{f}|K^t)p(K^t|\theta^t)}{Z_{K^t}} \tag{25}$$

where using that $\mathbf{f} = \mathrm{vec}(F^T)$, the prior of the latent function f can be written as

$$p(\mathbf{f}|K^t) = \frac{\exp\left\{-\frac{1}{2}\mathrm{Tr}\left(\left(K^t\right)^{-1}F\left(K^x\right)^{-1}F^T\right)\right\}}{(2\pi)^{\frac{1}{2}NM}|K^t|^{\frac{1}{2}N}|K^x|^{\frac{1}{2}M}}. \tag{26}$$

Combining the prior of the task covariance matrix and the prior over the latent values (26), we get the posterior of the

task covariance matrix as an inverse Wishart distribution given by

$$p(K^t|\mathbf{f},\theta^t) = \frac{2^{-\frac{1}{2}(\beta_N-M-1)M}|\Lambda_N|^{\frac{1}{2}(\beta_N-M-1)}}{\Gamma_M\left[\frac{1}{2}(\beta_N-M-1)\right]|K^t|^{\frac{1}{2}(\beta_N)}} \operatorname{etr}\left\{-\frac{1}{2}(K^t)^{-1}(\Lambda_N)\right\} \quad (27)$$

where $\Lambda_N = F(K^x)^{-1}F^T + \Lambda$, and $\beta_N = \beta + N$. It is worth taking a moment examining the form of the parameter matrix Λ_N of the posterior of the task covariance matrix. Λ_N is computed from $F(K^x)^{-1}F^T + \Lambda$, where $F \in \mathbb{R}^{M\times N}$, as the target matrix Y. This representation assumes that the complete set of responses is available, where each line of the F or Y matrix represents the latent values or targets for each input at that task. In most real-world applications though, it is desirable to combine data from different tasks for which outputs of the other tasks are not observed. Latent values for outputs that are not observed, which we will refer to as f^u, cannot be computed in this Bayesian setting, making the F matrix sparse. A possible solution would be to develop an EM algorithm to estimate the missing values, as was done in [14] with GPs or in [6] with t-processes. The implications this situation brings are investigated in the following section, along with a solution to this problem.

Appendix B
Treatment of Missing Values

We consider now the situation where some of the outputs are missing, termed as *partially missing values treatment*. For simplicity, we consider the case where for each task we are given a set of inputs $\mathbf{x}_j$ and a set of outputs $\mathbf{y}_j$ which we want to combine, while the outputs for this set of inputs for the other tasks are unobserved. We define $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$, with $\mathbf{x}_j$ denoting the *set* of inputs of the jth task of length n_j, for which outputs are observed. For the rest of this section it would also be useful to define the input matrix $\mathbf{X}$ in terms of each data point $x^i,\ i = 1:N$

$$\mathbf{X} = \left[\overbrace{x^1, \ldots, x^{n_1}}^{\mathbf{x}_1}, \ldots, \overbrace{x^{n_1+\cdots+n_{M-1}+1}, \ldots, x^{n_1+\cdots+n_M}}^{\mathbf{x}_M}\right].$$

Furthermore, we write F as a structured matrix of latent values f^o corresponding to the set of *observed* outputs, and f^u corresponding to the set of *un-observed* outputs

$$F = \begin{bmatrix} f^o_{1n_1} & f^u_{1n_2} & \cdots & \cdots & f^u_{1n_M} \\ f^u_{2n_1} & f^o_{2n_2} & f^u_{2n_3} & \cdots & f^u_{2n_M} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ f^u_{Mn_1} & f^u_{Mn_2} & \cdots & \cdots & f^o_{Mn_M} \end{bmatrix} \quad (28)$$

where, for example, $f^o_{2n_2}$ is the $(1\times n_2)$ vector of latent values associated with the observed outputs of task 2 and inputs $\mathbf{x}_2$, while $f^u_{2n_3}$ is the $(1\times n_3)$ vector of latent values associated with the unobserved outputs of task 2 and inputs $\mathbf{x}_3$.

We introduce the *communication* matrix S between all latent function realizations $\mathbf{f}$, corresponding to the observed and unobserved outputs, and the latent function realizations $\mathbf{f}^o = \left[f^o_{1n_1}, f^o_{2n_2}, \ldots, f^o_{Mn_M}\right]^T$, on the observed outputs only, whose relationship is given by

$$\mathbf{f}^o = S^T\mathbf{f} \quad (29)$$

where $S \in \mathbb{R}^{MN\times N}$, with functional form $S = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N]$, where each orthonormal column vector $s_n \in \mathbb{R}^{NM\times 1}$ has "1" on the ijth location indicated by the input vector x^i, $n = 1, \ldots, N$, and the task j that it belongs, and in *all* other locations zeros. The situation where for a specific input more than one or all outputs are observed can easily be handled by adding an additional column vector s with zeros everywhere except on the location given by the product of the index of the input i, and the index of the task j, as before. Moreover, it is known that a linear operator on a GP results again in a GP, thus the GP on the visible locations will be given by

$$\mathbf{f}^o \sim \mathcal{N}(0, S^T\left(K^t \otimes K^x\right)S). \quad (30)$$

The reason for utilizing this linear transformation is that the vector of all latent values $\mathbf{f}$ can now be written as $\mathbf{f} = S\mathbf{f}^o$. As a result, the F matrix is defined using the vec operator as $\operatorname{vec}(F^T) = S\mathbf{f}^o$. Although this operation makes F sparse, the computation of the posterior parameter matrix Λ_N can be utilized in a principled way. Taking a closer look at this operation reveals that the elements of F, which in essence were set to zero, have only an additive impact on the final form of the posterior of the task covariance matrix, which means that, if some outputs are not observed, it is sound not to consider them.

Acknowledgment

The authors would like to thank C. Williams, E. Bonilla, E. Vasilaki, M. Filippone, and the anonymous reviewers for the useful comments and conversations.

References

[1] R. Caruana, "Multi-task learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.
[2] K. Yu, V. Tresp, and A. Schwaighofer, "Learning Gaussian processes from multiple tasks," in *Proc. 22nd Int. Conf. Mach. Learn.*, Bonn, Germany, Aug. 2005, pp. 1012–1019.
[3] E. V. Bonilla, F. V. Agakov, and C. K. I. Williams, "Kernel multi-task learning using task-specific features," in *Proc. 11th Int. Workshop Artif. Intell. Stat.*, San Juan, Puerto Rico, Mar. 2007, pp. 43–50.
[4] T. Jebara, "Multi-task feature and kernel selection for SVMs," in *Proc. 21st Int. Conf. Mach. Learn.*, Banff, AB, Canada, Jul. 2004, pp. 55–62.
[5] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, 2008.
[6] S. Yu, V. Tresp, and K. Yu, "Robust multi-task learning with t-processes," in *Proc. 24th Int. Conf. Mach. Learn.*, Corvallis, OR, Jun. 2007, pp. 1103–1110.
[7] B. Bakker and T. Heskes, "Task clustering and gating for Bayesian multitask learning," *J. Mach. Learn. Res.*, vol. 4, pp. 83–99, Jan. 2003.
[8] J. Baxter, "A model of inductive bias learning," *J. Artif. Intell. Res.*, vol. 12, pp. 149–198, Mar. 2000.
[9] S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," in *Proc. 16th Annu. Conf. Comp. Learn. Theory*, Washington D.C., Aug. 2003, pp. 567–580.
[10] S. Ben-David and R. S. Borbely, "A notion of task relatedness yielding provable multiple-task learning guarantees," *Mach. Learn.*, vol. 73, no. 3, pp. 273–287, Jan. 2008.
[11] K. M. A. Chai, "Generalization errors and learning curves for regression with multi-task Gaussian processes," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2009, pp. 279–287.

[12] N. Lawrence and J. Platt, "Learning to learn with the informative vector machine," in *Proc. 21st Int. Conf. Mach. Learn.*, Banff, AB, Canada, Jul. 2004, pp. 65–72.
[13] A. Schwaighofer, V. Tresp, and K. Yu, "Learning Gaussian process kernels via hierarchical Bayes," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2005, pp. 1209–1216.
[14] E. Bonilla, K. M. Chai, and C. Williams, "Multi-task Gaussian process prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2008, pp. 153–160.
[15] Q. Liu, X. Liao, H. Li, J. R. Stack, and L. Carin, "Semisupervised multitask learning," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1074–1086, Jun. 2009.
[16] A. G. Journel and C. J. Huijbregts, *Mining Geostatistics*. London, U.K.: Academic, 1978.
[17] M. Opper and O. Winther, "Gaussian processes for classification: Mean-field algorithms," *Neural Comput.*, vol. 12, no. 11, pp. 2655–2684, 2000.
[18] M. Girolami and S. Rogers, "Variational Bayesian multinomial probit regression with Gaussian process priors," *Neural Comput.*, vol. 18, no. 8, pp. 1790–1817, 2006.
[19] J. H. Albert and S. Chib, "Bayesian analysis of binary and polychotomous response data," *J. Amer. Stat. Assoc.*, vol. 88, no. 422, pp. 669–679, 1993.
[20] C. E. Rasmussen and C. K. Williams, *Gaussian Processing Machine Learning*. Cambridge, MA: MIT Press, 2005.
[21] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proc. 17th Conf. Uncert. Artif. Intell.*, Seattle, WA, Aug. 2001, pp. 362–369.
[22] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*. Chapman and Hall/CRC, 2004.
[23] A. K. Gupta and D. K. Nagar, *Matrix Variate Distributions*. London, U.K.: Chapman & Hall, 2000.
[24] M. Titsias, N. Lawrence, and M. Rattray, "Efficient sampling for Gaussian process inference using control variables," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2009, pp. 1681–1688.
[25] K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu, "Stochastic relational models for discriminative link prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2007, pp. 1553–1560.
[26] R. Rebonato and P. Jäckel, "The most general methodology to create a valid correlation matrix for risk management and option pricing purposes," *J. Risk*, vol. 2, no. 2, pp. 1–12, 2000.
[27] R. M. Neal, *Bayesian Learning for Neural Networks* (Lecture Notes in Statistics). New York: Springer-Verlag, 1996, p. 118.
[28] J. A. Hanley and B. J. Mcneil, "The meaning and use of the area under a receiver operating characteristic curve," *Radiology*, vol. 143, no. 1, pp. 29–36, Apr. 1982.
[29] G. Skolidis, R. H. Clayton, and G. Sanguinetti, "Automatic classification of arrhytmic beats using Gaussian processes," *Comput. Cardiol.*, vol. 35, pp. 921–924, Sep. 2008.
[30] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. 215–220, 2000.
[31] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-task learning for classification with Dirichlet process priors," *J. Mach. Learn. Res.*, vol. 8, pp. 35–63, Jan. 2007.
[32] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[33] Y. Zhang and D.-Y. Yeung, "Multi-task learning using generalized t process," in *Proc. 14th Int. Workshop Artif. Intell. Stat.*, Sardinia, Italy, May 2010, pp. 964–971.
[34] S. Zhu, K. Yu, and Y. Gong, "Stochastic relational models for large-scale dyadic data using MCMC," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2009, pp. 1993–2000.
[35] K. Yu and W. Chu, "Gaussian process models for link analysis and transfer learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2008, pp. 1657–1664.
[36] Y. Teh, M. Seeger, and M. Jordan, "Semiparametric latent factor models," in *Proc. 10th Int. Workshop Artif. Intell. Stat.*, Jan. 2005, pp. 333–340.
[37] A. Birlutiu, P. Groot, and T. Heskes, "Multi-task preference learning with an application to hearing aid personalization," *Neurocomputing*, vol. 73, nos. 7–9, pp. 1177–1185, 2010.
[38] G. Pillonetto, F. Dinuzzo, and G. De Nicolao, "Bayesian online multitask learning of Gaussian processes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 193–205, Feb. 2010.
[39] M. Alvarez and N. D. Lawrence, "Sparse convolved Gaussian process for multi-output regression," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2009, pp. 57–64.
[40] N. D. Lawrence, G. Sanguinetti, and M. Rattray, "Modelling tramscriptional regulation using Gaussian processes," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2007, pp. 785–792.
[41] M. Boutell, J. Luo, X. Shen, and C. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, 2004.
[42] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya, "Hierarchical multi-label prediction of gene function," *Bioinformatics*, vol. 22, no. 7, pp. 830–836, 2006.
[43] M. Adankon, M. Cheriet, and A. Biem, "Semisupervised learning using bayesian interpretation: Application to LS-SVM," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 513–524, Apr. 2011.

Grigorios Skolidis received the Bachelors degree in automation engineering from Technological Educational Institute, Piraeus, Greece, in 2006, and the Masters degree in automatic control and systems engineering from the University of Sheffield, Sheffield, U.K., in 2008. He is currently pursuing the Ph.D. degree in computer science with the School of Informatics, University of Edinburgh, Edinburgh, U.K.

His current research interests include machine learning, signal processing, and biomedical application.

Guido Sanguinetti received the Masters degree in physics from the University of Genova, Genova, Italy, and the D.Phil. degree in mathematics from the University of Oxford, Oxford, U.K., in 1998 and 2002, respectively.

He was a Research Associate and then a Lecturer in computer science with the University of Sheffield, Sheffield, U.K., from 2004 and 2009. He joined the School of Informatics, University of Edinburgh, Edinburgh, U.K., as a Scottish Informatics and Computer Science Alliance Lecturer in machine learning in 2010. His current research interests include machine learning methodologies with application in bioinformatics and systems biology, particularly in the field of dynamical systems.