MULTI-TASK LEARNING IN DEEP GAUSSIAN PROCESSES WITH MULTI-KERNEL LAYERS

A PREPRINT

Ayman Boustati [*]
Mathematics of Systems CDT
University of Warwick

Richard R. Savage
Department of Statistics
University of Warwick

May 30, 2019

ABSTRACT

We present a multi-task learning formulation for Deep Gaussian processes (DGPs), describing a multi-kernel architecture for DGP layers. The proposed model is a non-linear mixture of latent Gaussian processes (GPs) with components shared between the tasks, in addition to separate task-specific components. Our formulation allows for learning complex relationships between tasks. We benchmark our model on three real-world datasets showing empirically that our formulation is able to improve the learning performance and transfer information between the tasks, outperforming state-of-the-art GP-based single-task learning and multi-task learning models.

***K*eywords** Multi-task learning · Deep Gaussian processes · Probabilistic modelling · Variational inference

1 Introduction

Multi-task learning is a framework in machine learning that leverages information shared between the training signals of related tasks with the aim of improving generalisation across the tasks [Caruana, 1997]. It has been successful in many applications such as: Reinforcement Learning [Teh et al., 2017], machine translation [Dong et al., 2015], computational biology [Widmer and Ratsch, 2012] and computer vision [Misra et al., 2016]. Multi-task learning can be particularly powerful in environments where training data are scarce for a single task, but otherwise available for multiple related tasks. This is sometimes known as *auxiliary* multi-task learning [Romera-Paredes et al., 2012].

Gaussian processes (GPs) [Rasmussen and Williams, 2006] are a powerful and flexible family of machine learning models. GPs have achieved state-of-the-art results on many applications [Deisenroth, 2011, Alaa and van der Schaar, 2017a, Snoek et al., 2012]. GPs as a modelling tool, are successful in both white-box (where prior knowledge is incorporated into the problem) and black-box settings; however, they are sometimes limited in their capacity and expressiveness. An extension of GPs known as Deep Gaussian Processes (DGPs) [Damianou and Lawrence, 2013], can overcome the limitations of standard GPs as they are able to model more complex input-to-output relationships. DGPs are a hierarchical composition of GPs that retain the advantages of GP-based models such as their non-parametric formulation, quantification of uncertainty and robustness to overfitting, while at the same time offering a richer class of models with the ability to learn representations from the data.

In this work, we present a multi-task learning framework for DGPs. Our proposed framework can capture complex non-linear relationships between tasks. Our modelling strategy assumes that similar tasks arise from an underlying process that is shared between them combined with processes that are specific to each task individually. Most GP-based multi-task models make this assumption, *linearly* combining the latent processes modelled by GPs to generate the outputs for the tasks [Nguyen and Bonilla, 2014, Wilson et al., 2012, Whye Teh et al., 2005]. Using a linear combination of processes limits the type of task relationships that can be learned. In our formulation, we take advantage of the recent advances in DGP inference [Salimbeni and Deisenroth, 2017], combining the latent processes *non-linearly* using another GP layer. This enables learning richer task relationships. This approach bears resemblance to certain multi-task

[*]Corresponding author: `a.boustati@warwick.ac.uk` .

learning architectures in neural network models, where layers at the bottom of the network are responsible for extracting shared information between the tasks and layers at the top are responsible for combining the shared information along with task specific information to generate the task outputs [Kovač, 2005, Bakker and Heskes, 2003, Caruana, 1997].

We demonstrate the capabilities of our model through experimentation on three real-world datasets showing that our multi-task learning formulation for DGPs outperforms other single-task and multi-task GP-based models in our experimental cases.

2 Model Specification

2.1 Modelling With Gaussian Processes

GPs can be used in predictive modelling where the labels are modelled as a transformation of a non-parametric latent function of the inputs [Rasmussen and Williams, 2006]:

$$\mathbf{y}_n | f; \mathbf{x}_n \sim p(\mathbf{y}_n | f(\mathbf{x}_n))$$

where f is a latent function drawn from a GP which is usually set to be zero-mean, i.e. $f(\cdot) \sim \mathcal{GP}(\mathbf{0}, k(\cdot,\cdot))$, and $p(\cdot)$ is an appropriate likelihood, e.g. Gaussian or Bernoulli. Exact inference in this model is possible only when the likelihood is Gaussian and with computational complexity of $O(N^3)$. Therefore, practitioners often resort to a sparse variational approximation that allows the use of other likelihoods, as well as a reduction in computational complexity to $O(NM^2)$ [Titsias, 2009, Hensman et al., 2013], where M is the number of inducing locations (pseudo inputs) which is typically much smaller than N.

The sparse variational approximation seeks to approximate the true GP posterior p, with an approximate posterior q, by minimising the Kullback-Leibler (KL) divergence between q and p. This is equivalent to maximising a lower bound on the marginal likelihood of the model, sometimes referred to as the evidence lower bound (ELBO):

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{F},\mathbf{U})}\left[\log \frac{p(\mathbf{Y},\mathbf{F},\mathbf{U})}{q(\mathbf{F},\mathbf{U})}\right] \tag{1}$$

where $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]$ are the outputs, $\mathbf{F} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]$ are the latent function values and $\mathbf{U} = [f(\mathbf{z}_1), \dots, f(\mathbf{z}_M)]$ are the function values at the inducing locations $\mathbf{z}_m$.

The sparse approximation above is a very practical tool, as it not only allows for more scalable inference in standard GP models, but also enables tractable inference in models involving the composition of GPs; for instance DGPs. A DGP is simply a composition of functions with a GP prior on each and i.i.d Gaussian noise between the layers [Damianou and Lawrence, 2013]:

$$\mathbf{y}_n | f^L; \mathbf{x}_n \sim p(\mathbf{y}_n | f^L(f^{L-1}(\dots f^1(\mathbf{x}_n))))$$

where $f^l(\cdot) \sim \mathcal{GP}(m^l(\cdot), k^l(\cdot,\cdot))$ (to simplify notation, the inter-layer noise is absorbed into the kernel $k^l(\cdot,\cdot)$ for $l \in \{1, \dots, L-1\}$). Inference in this model is possible through the use of the variational sparse approximation framework, where we can construct an ELBO that is given by [Damianou and Lawrence, 2013, Salimbeni and Deisenroth, 2017]:

$$\mathcal{L}_{\text{DGP}} = \mathbb{E}_{q(\{\mathbf{F}^l,\mathbf{U}^l\}_{l=1}^L)}\left[\log \frac{p(\mathbf{Y}, \{\mathbf{F}^l,\mathbf{U}^l\}_{l=1}^L)}{q(\{\mathbf{F}^l,\mathbf{U}^l\}_{l=1}^L)}\right] \tag{2}$$

2.2 Multi-task Formulation

Consider the case where we have T tasks. Let $\mathbf{X}^t$ be the $N^t \times D^{\text{in}}$ data matrix for task $t \in \{1, \dots, T\}$ and $\mathbf{Y}^t$ be an $N^t \times D^{\text{out}}$ matrix corresponding to the outputs for task t. We propose a model where all T tasks share a set of I latent representations $\{\mathbf{G}_i\}_{i=1}^I$ generated by a set of I functions $\{g_i(\cdot)\}_{i=1}^I$, where $g_i : \mathbb{R}^{D^{\text{in}}} \to \mathbb{R}^{D^{G_i}}$. In addition, they possess a set of J_t task specific representations $\{\mathbf{H}_j^t\}_{j=1}^{J^t}$ generated by a set of J^t task specific functions $\{h_j^t(\cdot)\}_{j=1}^{J^t}$ for each task t, where $h_j^t : \mathbb{R}^{D^{\text{in}}} \to \mathbb{R}^{D^{H_j^t}}$. The features in the combined latent space $\{\{\mathbf{G}_i\}_{i=1}^I, \{\mathbf{H}_j^t\}_{j=1}^{J^t}\}$ are in turn warped with a task specific random function $f^t(\cdot)$ with $f^t : \mathbb{R}^{D^t} \to \mathbb{R}^{D^{\text{out}}}$ to generate the noiseless outputs $\mathbf{F}^t$ for task t (Figure 1).[2]

[2] $D^t = \sum_{i=1}^I D^{G_i} + \sum_{j=1}^J D^{H_j^t}$

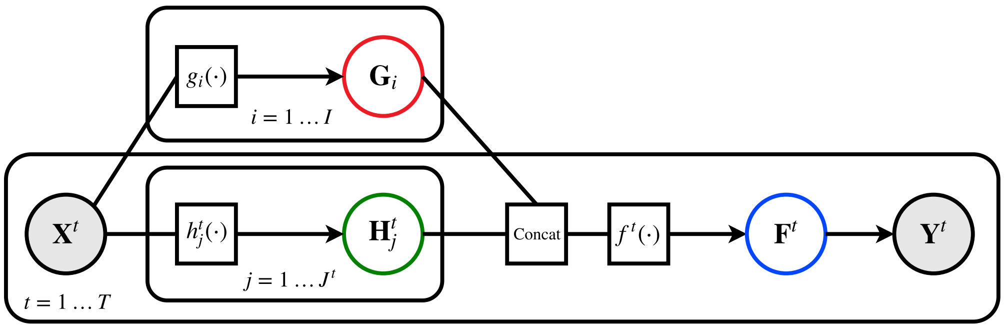

Figure 1: A graphical representation of the model. The white circles represent latent variables, The grey circles represent observed variables and the squares are computational graph nodes. Some circles are colour coded to match their corresponding terms in equations (3, 4, 11,12).

We place independent GP priors on both the shared and task specific latent functions:

$$g_i \sim \mathcal{GP}(m_i(\cdot), k_i(\cdot,\cdot))$$
$$h_j^t \sim \mathcal{GP}(m_j^t(\cdot), k_j^t(\cdot,\cdot))$$
$$f^t \sim \mathcal{GP}(m^t(\cdot), k^t(\cdot,\cdot))$$

This generative process describes a 2-level hierarchical model with GPs that can be formulated as a 2-layer DGP, where the first layer is a multi-kernel layer responsible for sharing the information between tasks, while the second layer is a multi-kernel layer responsible for the task specific outputs in a similar fashion to multi-head neural networks [Bakker and Heskes, 2003]. The model is fully described by the joint probability distribution below:

$$p(\{\mathbf{Y}^t, \mathbf{F}^t, \{\{\mathbf{H}_j^t\}_{j=1}^{J^t}\}_{t=1}^{T}, \{\mathbf{G}_i\}_{i=1}^{I}) =$$
$$\overbrace{[\prod_{t=1}^{T} \prod_{n=1}^{N^t} p(\mathbf{y}_n^t|\mathbf{F}^t)\ ]}^{\text{task-specific likelihood}} \overbrace{[\prod_{t=1}^{T} p(\mathbf{F}^t|\{\mathbf{H}_j^t\}_{j=1}^{J^t}, \{\mathbf{G}_i\}_{i=1}^{I})]}^{\text{GP prior on second layer}}$$
$$\underbrace{[\prod_{t=1}^{T} \prod_{j=1}^{J^t} p(\mathbf{H}_j^t; \mathbf{X}^t)][\prod_{i=1}^{I} p(\mathbf{G}_i; \mathbf{X})]}_{\text{GP prior on first layer}} \tag{3}$$

where $\mathbf{X}$ is the full $N \times D^{\text{in}}$ data matrix for $N = \sum_{t=1}^{T} N^t$.

Sparse Variational Approximation Inference in this model is intractable due to the non-linear dependencies introduced between the layers [Damianou and Lawrence, 2013]. Hence, we resort to a variational sparse approximation for DGPs introduced in Salimbeni and Deisenroth [2017]. We define the joint distribution of an expanded model as follows:

$$p(\{\mathbf{Y}^t, \mathbf{F}^t, \tilde{\mathbf{F}}^t, \{\mathbf{H}_j^t, \tilde{\mathbf{H}}_j^t\}_{j=1}^{J^t}\}_{t=1}^{T}, \{\mathbf{G}_i, \tilde{\mathbf{G}}_i\}_{i=1}^{I}) =$$
$$[\prod_{t=1}^{T} \prod_{n=1}^{N^t} p(\mathbf{y}_n^t|\mathbf{F}^t)]$$
$$[\prod_{t=1}^{T} p(\mathbf{F}^t|\tilde{\mathbf{F}}^t; \{\mathbf{H}_j^t\}_{j=1}^{J^t}, \{\mathbf{G}_i\}_{i=1}^{I}, \mathbf{Z}_{\mathbf{F}^t}) p(\tilde{\mathbf{F}}^t; \mathbf{Z}_{\mathbf{F}^t})]$$
$$[\prod_{t=1}^{T} \prod_{j=1}^{J^t} p(\mathbf{H}_j^t|\tilde{\mathbf{H}}_j^t; \mathbf{X}^t, \mathbf{Z}_{\mathbf{H}_j^t}) p(\tilde{\mathbf{H}}_j^t; \mathbf{Z}_{\mathbf{H}_j^t})]$$
$$[\prod_{i=1}^{I} p(\mathbf{G}_i|\tilde{\mathbf{G}}_i; \mathbf{X}, \mathbf{Z}_{\mathbf{G}_i}) p(\tilde{\mathbf{G}}_i; \mathbf{Z}_{\mathbf{G}_i})] \tag{4}$$

where we introduce the inducing variables $\{\tilde{\mathbf{G}}_i\}_{i=1}^I, \{\{\tilde{\mathbf{H}}_j^t\}_{j=1}^{J^t}\}_{t=1}^T, \{\tilde{\mathbf{F}}^t\}_{t=1}^T$ corresponding to the values of the latent functions evaluated at a set of M inducing locations, i.e. $\tilde{\mathbf{G}}_i = g_i(\mathbf{Z}_{G_i})$, $\tilde{\mathbf{H}}_j^t = h_j^t(\mathbf{Z}_{\mathbf{H}_j^t})$, $\tilde{\mathbf{F}}^t = f^t(\mathbf{Z}_{\mathbf{F}^t})$. Introducing the inducing variables allows us to factorise the joint GP priors into the prior on the inducing variables and the conditional given the inducing variables.

For instance, in the case of $(\mathbf{G}_i, \tilde{\mathbf{G}}_i)$, we have:

$$p(\mathbf{G}_i, \tilde{\mathbf{G}}_i; \mathbf{X}, \mathbf{Z}_{\mathbf{G}_i}) = p(\mathbf{G}_i | \tilde{\mathbf{G}}_i; \mathbf{X}, \mathbf{Z}_{\mathbf{G}_i}) p(\tilde{\mathbf{G}}_i; \mathbf{Z}_{\mathbf{G}_i}) \tag{5}$$

where,

$$p(\tilde{\mathbf{G}}_i; \mathbf{Z}_{\mathbf{G}_i}) = \mathcal{N}(\tilde{\mathbf{G}}_i | m_i(\mathbf{Z}_{\mathbf{G}_i}), k_i(\mathbf{Z}_{\mathbf{G}_i}, \mathbf{Z}_{\mathbf{G}_i})) \tag{6}$$

$$p(\mathbf{G}_i | \tilde{\mathbf{G}}_i; \mathbf{X}, \mathbf{Z}_{\mathbf{G}_i}) = \mathcal{N}(\mathbf{G}_i | \tilde{\boldsymbol{\mu}}_{\boldsymbol{i}}, \tilde{\boldsymbol{\Sigma}}_i) \tag{7}$$

with

$$\tilde{\boldsymbol{\mu}}_i = m_i(\mathbf{X}) + \boldsymbol{\alpha}_i(\mathbf{X})^T(\tilde{\mathbf{G}}_i - m_i(\mathbf{Z}_{\mathbf{G}_i})) \tag{8}$$

$$\tilde{\boldsymbol{\Sigma}}_i = k_i(\mathbf{X}, \mathbf{X}) - \boldsymbol{\alpha}_i(\mathbf{X})^T k_i(\mathbf{Z}_{\mathbf{G}_i}, \mathbf{Z}_{\mathbf{G}_i}) \boldsymbol{\alpha}_i(\mathbf{X}) \tag{9}$$

$$\boldsymbol{\alpha}_i(\mathbf{X}) = k_i(\mathbf{Z}_{\mathbf{G}_i}, \mathbf{Z}_{\mathbf{G}_i})^{-1} k_i(\mathbf{Z}_{\mathbf{G}_i}, \mathbf{X}) \tag{10}$$

The same treatment can be applied to the pairs $(\mathbf{H}_j^t, \tilde{\mathbf{H}}_j^t)$ and $(\mathbf{F}^t, \tilde{\mathbf{F}}^t)$ taking the input pairs $(\mathbf{X}^t, \mathbf{Z}_{\mathbf{H}_j^t})$ and $(\{\{\mathbf{G}_i\}_{i=1}^I, \{\mathbf{H}_j^t\}_{j=1}^{J^t}\}, \mathbf{Z}_{\mathbf{F}^t})$ respectively.

Expanding the probability space this way allows us to define an approximate variational posterior as follows:

$$\begin{aligned} & q(\{\mathbf{F}^t, \tilde{\mathbf{F}}^t, \{\mathbf{H}_j^t, \tilde{\mathbf{H}}_j^t\}_{j=1}^{J^t}\}_{t=1}^T, \{\mathbf{G}_i, \tilde{\mathbf{G}}_i\}_{i=1}^I) = \\ & [\prod_{t=1}^T p(\mathbf{F}^t | \tilde{\mathbf{F}}^t; \{\mathbf{H}_j^t\}_{j=1}^J, \{\mathbf{G}_i\}_{i=1}^I, \mathbf{Z}_{\mathbf{F}^t}) q(\tilde{\mathbf{F}}^t)] \\ & [\prod_{t=1}^T \prod_{j=1}^{J^t} p(\mathbf{H}_j^t | \tilde{\mathbf{H}}_j^t; \mathbf{X}^t, \mathbf{Z}_{\mathbf{H}_j{}^t}) q(\tilde{\mathbf{H}}_j^t)] \\ & [\prod_{i=1}^I p(\mathbf{G}_i | \tilde{\mathbf{G}}_i; \mathbf{X}, \mathbf{Z}_{\mathbf{G}_i}) q(\tilde{\mathbf{G}}_i)] \end{aligned} \tag{11}$$

where we define the q's on the right-hand side as free-form Gaussians, e.g. $q(\tilde{\mathbf{G}}_i) = \mathcal{N}(\tilde{\mathbf{G}}_i | \mathbf{m}_i, \mathbf{G}_i)$ and similarly for $\tilde{\mathbf{H}}_j^t$ and $\tilde{\mathbf{F}}^t$.

Evidence Lower Bound (ELBO) Now we are able to construct an ELBO using the formulation in (2):

$$\begin{aligned} \mathcal{L} = & \sum_{t=1}^T \sum_{n=1}^{N^t} \mathbb{E}_{q(\mathbf{f}_n^t)}[\log p(\mathbf{y}_n^t | \mathbf{f}_n^t)] \\ & - \sum_{t=1}^T \text{KL}\left[q(\tilde{\mathbf{F}}^t) \middle\| p(\tilde{\mathbf{F}}^t; \mathbf{Z}_{\mathbf{F}^t})\right] \\ & - \sum_{t=1}^T \sum_{j=1}^{J^t} \text{KL}\left[q(\tilde{\mathbf{H}}_j^t) \middle\| p(\tilde{\mathbf{H}}_j^t; \mathbf{Z}_{\mathbf{H}_j^t})\right] \\ & - \sum_{i=1}^I \text{KL}\left[q(\tilde{\mathbf{G}}_i) \middle\| p(\tilde{\mathbf{G}}_i; \mathbf{Z}_{\mathbf{G}_i})\right] \end{aligned} \tag{12}$$

This bound has complexity $\mathcal{O}(NM^2(\sum_{i=1}^I D^{G_i} + \sum_{t=1}^T \sum_{j=1}^{J^t} D^{H_j^t} + D^{\text{out}}))$ and can be evaluated approximately as in Salimbeni and Deisenroth [2017] using Monte Carlo samples from the variational posterior. We refer the reader to Salimbeni and Deisenroth [2017] for a more detailed derivation of the lower bound in (12) and considerations on evaluating it.

Extension To Deeper Architectures Our description of the model focused on a 2-layer DGP; however, the framework that we described is general and can be applied to a DGP with arbitrary depth. Looking at the ELBO in (12), we can see that the contribution of the first layer appears in the KL terms involving the $\tilde{\mathbf{G}}_i$'s and the $\tilde{\mathbf{H}}_j^t$'s. Therefore, a layer of this structure can be added at any level of a DGP cascade by simply adapting the ELBO to include these terms. Since the sparse approximation and the inference framework in Salimbeni and Deisenroth [2017] does not require the computation of the full covariance within the layers, it is possible to propagate a Monte Carlo sample (using the re-parameterisation trick [Rezende et al., 2014, Kingma et al., 2015]) through this layer structure at any point in the cascade without the need to compute inter-task covariance. This enables approximate evaluation of the reconstruction error term in (12). A similar argument can be made for the structure of the second layer.

3 Related Work

There is a rich literature on multi-task learning in GP models. Most GP multi-task models correlate the outputs by mixing a set of independent processes with a different set of coefficients for each output. Examples of such models include: the semi-parametric latent factor model (SLFM) [Whye Teh et al., 2005], intrinsic coregionalisation model (ICM) [Bonilla et al., 2007, Skolidis and Sanguinetti, 2011] and the linear model for coregionalisation (LMC) [Goovaerts, 1997, Alvarez et al., 2011]. Nguyen and Bonilla [2014] extend SLFM to handle dependent tasks in very large datasets using the framework in Hensman et al. [2013]. In Titsias and Lazaro-Gredilla [2011], the authors give this type of models a Bayesian treatment by placing a spike-and-slab prior on the mixing coefficients. More complex models that can handle more complex relationships between the outputs are formulated in Wilson et al. [2012] and Nguyen and Bonilla [2013] introducing mixing weights with dependencies on the inputs, and in Boyle and Frean [2004] and Alvarez and Lawrence [2009] by convolving processes. In the DGP literature, there are two formulations for the multi-task setting. Kandemir [2015] proposes to linearly combine the hidden layers of different DGP models trained on different tasks, inducing information transfer between the models. Alaa and van der Schaar [2017b] uses the ICM kernel in the hidden and output layers of the DGPs to implement multi-task learning for survival analysis with competing risks. The framework that we presented naturally extends the models above by replacing the linear mixing of latent processes with another GP, thus allowing to model more complex relationships between the tasks.

A similar idea to multi-task learning is that of multi-view learning, where multiple views of the data are combined into a single latent representation. In the Gaussian process literature, Manifold Relevance Determination (MRD) [Damianou et al., 2012] is a latent variable model for multi-view learning, where the latent space is softly separated into a component that is shared between all of the views and private components that are view-specific. In our formulation of multi-task learning, we attempt to do the inverse mapping, i.e. disentangling multiple representations from a common feature space. We also encode a hard separation between shared components and task-specific components of the latent space.

4 Experimental Results

We test our model on three real-world datasets. For all of the experiments, we compare our model with four other GP-based models: **STL-DGP**: a single-task 2-layer DGP [Damianou and Lawrence, 2013, Salimbeni and Deisenroth, 2017] trained on each task individually; **ICM-DGP**: a formulation of multi-task learning in DGPs applied in Alaa and van der Schaar [2017b] for survival analysis with competing risks. This model uses the Intrinsic Coregionalisation (ICM) kernel in the hidden and output layers of the DGP [3]; **STL-GP**: a single-task variational sparse Gaussian process [Hensman et al., 2013] trained on each task individually; **ICM-GP**: a multi-task variational sparse Gaussian process [Nguyen and Bonilla, 2014, Bonilla et al., 2007]. This model uses the ICM kernel for multi-task learning.

Further, we test three variations of our model. The first variation is the standard 2-layer set-up described in Section 2.2, which we refer to as **MTL-DGP 1**. The second variation is also a 2-layer set-up with the task specific component turned off in the first layer. We refer to this as **MTL-DGP ***. The third variation is a 3-layer set-up where the hidden layers have both task-specific and sharing components. We refer to this set-up as **MTL-DGP 2**.

For all the models, we use the Matérn-5/2 kernel, initialising the kernel parameters to the same values. We also use the same number of inducing inputs for all the models. We jointly learn the variational parameters and hyperparameters (of the kernels and likelihoods) through the maximisation of the models' ELBO. We use the Adam Optimiser [Kingma and Ba, 2015] with learning rate 10^{-3} on the DGP-based models, and L-BFGS [Nocedal and Wright, 2000] for the shallow GP-based models (except when mini-batching was used where we revert back to Adam).

[3]We note that this is a special case formulation of our model where all the shared processes in the first layer are turned off and the task specific processes in the first and second layers are derived from the ICM kernel.

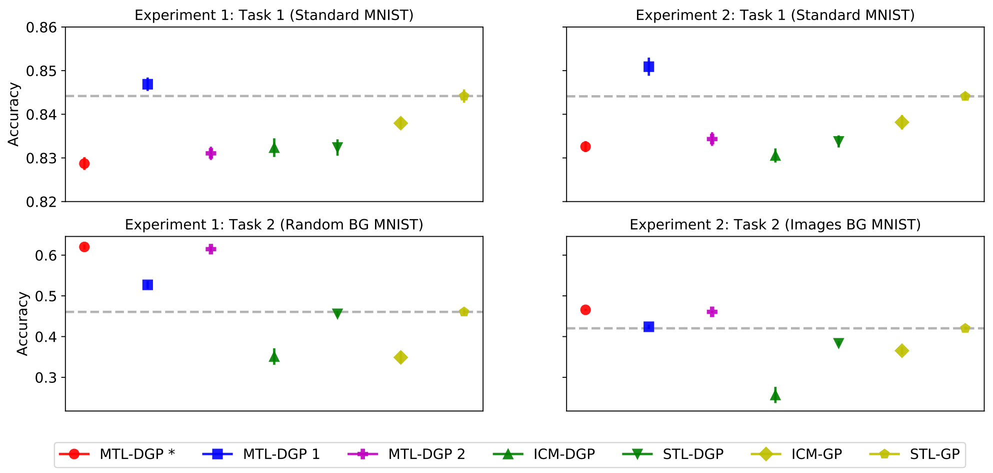

Figure 2: Classification accuracy on the MNIST variations experiments. Experiment 1 (left) is the standard MNIST and Random BG MNIST pair. Experiment 2 (right) is the standard MNIST and Image BG MNIST pair. The dashed line highlights the single task GP baseline. Higher is better.

For the deep models, we use Automatic Relevance Determination (ARD) [Rasmussen and Williams, 2006] in the top layer. This allows the processes in the final layer to automatically determine the amount of information to use from the previous layers, balancing between the shared and task specific latent features in the case of the multi-task models in a similar fashion to Damianou et al. [2012].

4.1 MNIST Variations

We use the MNIST digits dataset [Lecun et al., 1998], as well as two other variations. The first variation, termed **Random Background (BG) MNIST**, replaces the black background for the digits with random noise. The second variation, termed **Images Background (BG) MNIST**, replaces the background with randomly extracted patches from 20 black and white images. [4] These two variations are more challenging for classification due to the presence of non-monochromatic backgrounds.

We sample 500 images from the training split of the standard MNIST dataset and 100 images from the training splits of the two other versions. We conduct two separate experiments: **Experiment 1** pairs standard MNIST (**Task 1**) with the Random BG MNIST (**Task 2**), **Experiment 2** pairs standard MNIST (**Task 1**) and Images BG MNIST (**Task 2**). We repeat the sampling and experimentation procedure 10 times and report the average accuracy and its standard error on the standard test split for the datasets. The results are shown in Figure 2.

The purpose of these two experiments is to test the ability of the model to transfer information from an easy task with a large number of training samples (Task 1) to a harder task with a smaller number of training samples (Task 2). This is sometimes termed as auxiliary multi-task learning. In this instance, we expect our model to use the standard MNIST task as an auxiliary task to boost the performance on the more difficult variations.

Figure 2 suggests that all the proposed multi-task architectures are successfully able to to use the information in the auxiliary tasks to improve the learning on the primary tasks. We observe that for Experiment 1, we get a considerable improvement over the STL-GP baseline. However, although MTL-DGP * and MTL-DGP 2 outperform the other models on Task 2, they suffer a performance hit on Task 1. This might be due to negative transfer in the direction of Task 1. A similar behaviour is seen in Experiment 2 with both the performance gain on Task 2 and the performance hit on Task 1 less pronounced.

Conversely, MTL-DGP 1 achieves a gain in performance over the baseling on all of the tasks, but the gain is less prominent on Task 2 in both experiments than MTL-DGP * and MTL-DGP 2. This suggests that the choice of architecture for the multi-task models plays an important role in the direction and strength of information transfer.

[4] `http://www.iro.umontreal.ca/~lisa/twiki/bin/view.cgi/Public/MnistVariations`

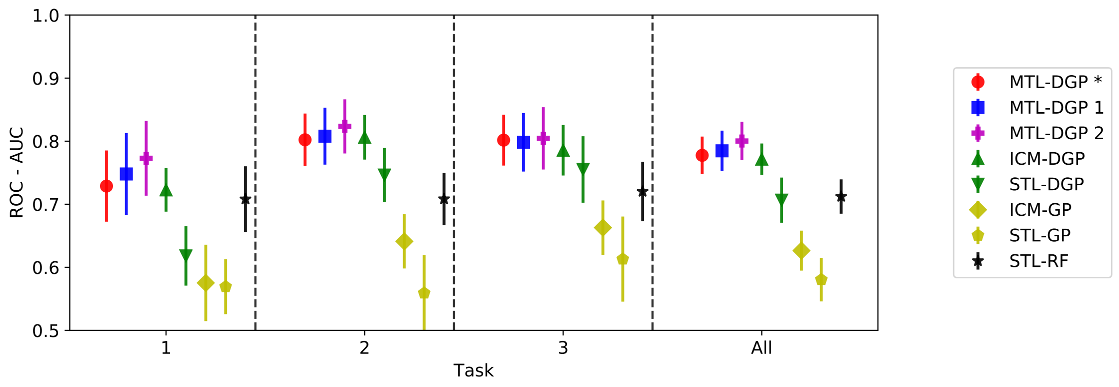

Figure 3: ROC-AUC results on the FAIMS dataset. Higher is better.

4.2 FAIMS Diabetes Diagnosis

Field Asymmetric Ion Mobility Spectrometry (FAIMS) is a method for detecting Volatile Organic Compounds (VOCs) which for example contribute to a given odour. VOCs are known to carry information on a range of disease states in humans, and technologies such as FAIMS can be used to cheaply and non-invasively diagnose a patient's disease from a simple biological sample such as urine [Covington et al., 2015].

We have access to data from a case-control study of 125 patients who have been tested for diabetes. 48 out of 125 have been found to have diabetes (the disease group), while the rest are disease-free (the control group). The data consists of three experimental runs per patient, corresponding to sequential FAIMS analyses on the same urine sample. These are expected to contain similar but not identical signals, as different VOCs evaporate at different rates, meaning the overall VOC signal varies over time. For the purpose of our benchmark, we treat each experimental run as a task. By doing so, we aim to share statistical strength between runs to improve the accuracy of prediction and overcome the scarcity of training data.

Due to the large size of the feature space and its sparsity, we perform Sparse Principal Component Analysis (SPCA) decomposition [Hastie et al., 2009, Mairal et al., 2009] on the features selecting the first 20 principal components. We perform 10-fold cross-validation on 70:30 train-test splits. We report the value of the area under the Receiver Operating Characteristic Curve (ROC-AUC) for each task averaged across the folds, as well as the average value of the ROC-AUC for all tasks and runs. We also present error bars on the ROC-AUC measurements using the standard error obtain from the cross-validation process. Additionally, we include a single task Random Forest classifier [Hastie et al., 2009] as a benchmark, which has been shown to perform well on this type of problems [Martinez-Vernon et al., 2018]. The results of this experiment are shown in Figure 3.

We observe that the multi-task models with deep architectures outperform the other GP-based models. They also significantly outferporm the STL-RF baseline on the second task. Interestingly, both STL-DGP and ICM-GP outperform STL-GP on average, indicating that representation learning as well as information sharing is important for this problem. Hence, our proposed multitask architectures (MTL-DGP */1/2) along with ICM-DGP are able to leverage information transfer and depth to perform well on this type of problems compared to the other tested models.

4.3 SARCOS Robot Inverse Dynamics

We consider the SARCOS regression dataset relating to the inverse dynamics problem for a seven degrees-of-freedom anthropomorphic robot arm [Vijayakumar et al., 2002, Rasmussen and Williams, 2006]. The dataset consists of 44,484 training observations with 21 variables (7 joint positions, 7 joint velocities, 7 joint accelerations) and the corresponding real-valued torques for 7 joints, i.e. 7 tasks corresponding to each joint torque. Additionally, there are 4,449 testing examples with all the 7 joint torques available for each.

The experimental procedure is described as follows: For N in $\{100, 200, 500, 1000, 2000, 5000\}$, sample N training data points from the training set. This constitutes the training set for one experimental run. For each of the N sampled points select 1 of the 7 labels uniformly at random. Split the experimental training set into 7 according to which joint the label corresponds to. These 7 splits constitute our 7 tasks. We train the models on the training set after standardising the features and the targets and test for all 7 labels on the test set.

Table 1: NLPP scores on the SARCOS test set averaged over the 7 tasks. Lower is better.

	Number of Training Inputs					
	100	200	500	1000	2000	5000
MTL-DGP *	1.44 ± 0.14	$\mathbf{0.94 \pm 0.06}$	0.57 ± 0.05	0.18 ± 0.06	$\mathbf{0.01 \pm 0.06}$	$\mathbf{-0.14 \pm 0.06}$
MTL-DGP 1	1.39 ± 0.20	1.07 ± 0.13	$\mathbf{0.40 \pm 0.08}$	$\mathbf{0.13 \pm 0.06}$	0.04 ± 0.06	-0.09 ± 0.06
MTL-DGP 2	1.17 ± 0.06	1.90 ± 0.25	0.48 ± 0.08	0.15 ± 0.06	0.02 ± 0.06	-0.12 ± 0.06
ICM-DGP	1.56 ± 0.16	1.04 ± 0.07	0.72 ± 0.04	0.38 ± 0.06	0.17 ± 0.06	-0.11 ± 0.06
STL-DGP	1.81 ± 0.26	0.98 ± 0.08	0.73 ± 0.03	0.63 ± 0.03	0.42 ± 0.05	0.10 ± 0.05
ICM-GP	1.60 ± 0.20	0.96 ± 0.10	0.73 ± 0.11	0.45 ± 0.11	0.15 ± 0.06	0.04 ± 0.06
STL-GP	$\mathbf{1.05 \pm 0.09}$	0.96 ± 0.10	0.80 ± 0.10	0.69 ± 0.11	0.20 ± 0.05	0.06 ± 0.05

Table 2: RMSE scores on the SARCOS test set averaged over the 7 tasks. Lower is better.

	Number of Training Inputs					
	100	200	500	1000	2000	5000
MTL-DGP *	0.79 ± 0.03	0.64 ± 0.03	0.46 ± 0.02	0.32 ± 0.02	$\mathbf{0.27 \pm 0.01}$	$\mathbf{0.22 \pm 0.01}$
MTL-DGP 1	0.78 ± 0.03	$\mathbf{0.59 \pm 0.03}$	$\mathbf{0.39 \pm 0.02}$	$\mathbf{0.31 \pm 0.02}$	$\mathbf{0.27 \pm 0.02}$	0.23 ± 0.01
MTL-DGP 2	0.78 ± 0.02	0.65 ± 0.03	0.40 ± 0.02	$\mathbf{0.31 \pm 0.02}$	$\mathbf{0.27 \pm 0.01}$	0.23 ± 0.01
ICM-DGP	0.83 ± 0.04	0.67 ± 0.04	0.52 ± 0.02	0.38 ± 0.02	0.31 ± 0.02	0.23 ± 0.01
STL-DGP	0.76 ± 0.03	0.63 ± 0.03	0.53 ± 0.02	0.47 ± 0.01	0.38 ± 0.02	0.27 ± 0.01
ICM-GP	0.74 ± 0.05	0.67 ± 0.05	0.59 ± 0.06	0.46 ± 0.06	0.29 ± 0.02	0.25 ± 0.02
STL-GP	$\mathbf{0.73 \pm 0.03}$	0.71 ± 0.05	0.63 ± 0.06	0.57 ± 0.06	0.30 ± 0.02	0.25 ± 0.01

For each N, we repeat the experimental procedure 10 times to obtain confidence bounds on our metrics, calculated as 2 times the standard error. We report the negative log predictive probability (NLPP) and the root mean squared error (RMSE), averaged across the 7 tasks. We use the NLPP scores to assess the models' ability to quantify uncertainty, and the RMSE scores to measure the mean predictions. The results for this experiment are presented in Tables 1 and 2.

Looking at the NLPP scores in Table 1, we observe that for a small amount of training data (100 inputs), STL-GP achieves considerably better performance in comparison to the other models including the presented multi-task models; however, as the number of training inputs increases MTL-DGP */1/2 are able to outperform the other models with wider margins as the training data grows. A similar observation is made for the RMSE scores.

5 Conclusion

We introduced a new framework for multi-task learning for DGPs, presenting a multi-kernel layer structure that can learn shared information between the tasks, as well as task specific information. Our experimental results show that the multi-task formulation presented in this paper is indeed effective and can improve the learning performance on multiple tasks. We observe that the model can transfer information between multiple tasks of similar sizes; in addition to being successful in the auxiliary multi-task learning setting.

In practice, we found that the presented model incurs a computational burden when the feature space or the training set are large. Therefore, we think that investigating the scalability of this type of models for large problems is a very fruitful direction for future research.

While in all of our experiments we treated the modelling as black-box predictions, our framework allows for incorporating modelling assumptions through the choice of the kernels in the shared and task-specific processes. This could be crucial in some applications where more is known about the task relationships and where there is a risk of negative transfer between the tasks.

We have implemented this model using GPflow [Matthews et al., 2017], an open-source Python library for GPs, and made our implementation publicly available on GitHub. [5]

Acknowledgments

This work was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1. AB gratefully acknowledges his scholarship award from the University of Warwick.

References

Rich Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, 1997.

Yee Teh, Victor Bapst, Wojciech M. Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 1723–1732, 2015.

Christian Widmer and Gunnar Ratsch. Multitask learning in computational biology. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 2012.

I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-Stitch Networks for Multi-task Learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Bernardino Romera-Paredes, Andreas Argyriou, Nadia Berthouze, and Massimiliano Pontil. Exploiting unrelated tasks in multi-task learning. In *International Conference on Artificial Intelligence and Statistics*, 2012.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2006. ISBN 978-0-262-18253-9.

Marc Deisenroth. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, Washington, USA, 2011.

Ahmed M. Alaa and Mihaela van der Schaar. Bayesian Inference of Individualized Treatment Effects using Multi-task Gaussian Processes. In *Advances in Neural Information Processing Systems*, Long Beach, California, 2017a.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, 2012.

Andreas C. Damianou and Neil D. Lawrence. Deep Gaussian Processes. In *Proceedings of 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Scottsdale, USA, 2013. JMLR.

Trung V. Nguyen and Edwin V. Bonilla. Collaborative Multi-output Gaussian Processes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, Quebec City, Canada, 2014.

Andrew Gordon Wilson, David A. Knowles, and Zoubin Ghahramani. Gaussian Process Regresssion Networks. In *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, UK, 2012.

Yee Whye Teh, Matthias Seeger, and Michael Jordan, I. Semiparametric Latent Factor Models. In *Proceedings of 10th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2005.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems*, Long Beach, California, 2017.

Krunoslav Kovač. *Multitask Learning for Bayesian Neural Networks*. PhD thesis, University of Toronto, 2005.

Bart Bakker and Tom Heskes. Task clustering and gating for Bayesian Multitask Learning. *Journal of Machine Learning Research*, 4:83–99, 2003.

Michalis K. Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *Proceedings of 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 5, Clearwater Beach, Florida, USA, 2009.

James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, Bellevue, Washington, USA, 2013.

[5] `https://github.com/aboustati/dgplib`

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Diederik P Kingma, Tim Salimans, and Max Welling. Variational Dropout and the Local Reparameterization Trick. In *Advances in Neural Information Processing Systems*, 2015.

Edwin V. Bonilla, Kian M. Chai, and Christopher Williams. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems*, pages 153–160, Vancouver, Canada, 2007.

G. Skolidis and G. Sanguinetti. Bayesian Multitask Classification With Gaussian Process Priors. *IEEE Transactions on Neural Networks*, 22(12):2011–2021, 2011.

P. Goovaerts. *Geostatistics for Natural Resources Evaluation*. Applied geostatistics series. Oxford University Press, 1997. ISBN 978-0-19-511538-3.

Mauricio Alvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for Vector-Valued Functions: a Review. Technical Report, MIT, 2011.

Michalis K. Titsias and Miguel Lazaro-Gredilla. Spike and slab variational inference for multi-task and multiple kernel learning. In *Advances in Neural Information Processing Systems*, Granada, Spain, 2011.

Trung Nguyen and Edwin Bonilla. Efficient Variational Inference for Gaussian Process Regression Networks. In *Proceedings of 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2013.

Phillip Boyle and Marcus Frean. Dependent Gaussian Processes. In *Advances in Neural Information Processing Systems*, Vancouver, Canada, 2004. MIT Press.

Mauricio Alvarez and Neil D. Lawrence. Sparse convolved Gaussian Process for multi-output regression. In *Advances in Neural Information Processing Systems*, Vancouver, Canada, 2009.

Melih Kandemir. Asymmetric Transfer Learning with Deep Gaussian Processes. In *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 2015.

Ahmed M. Alaa and Mihaela van der Schaar. Deep Multi-task Gaussian Processes for Survival Analysis with Competing Risks. In *Advances in Neural Information Processing Systems*, Long Beach, California, 2017b.

Andreas C. Damianou, Carl Henrik Ek, Michalis K. Titsias, and Neil D. Lawrence. Manifold Relevance Determination. In *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, UK, 2012. URL `http://www.icml.cc/2012/papers/94.pdf`.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, San Diego, California, 2015.

Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer series in operations research. Springer, New York, NY, corr. 2. print edition, 2000. ISBN 978-0-387-98793-4.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

J. A. Covington, M. P. van der Schee, A. S. L. Edge, B. Boyle, R. S. Savage, and R. P. Arasaradnam. The application of FAIMS gas analysis in medical diagnostics. *Analyst*, 140(20):6775–6781, 2015.

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009. ISBN 978-0-387-84884-6.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009. ACM.

Andrea S. Martinez-Vernon, James A. Covington, Ramesh P. Arasaradnam, Siavash Esfahani, Nicola O'Connell, Ioannis Kyrou, and Richard S. Savage. An improved machine learning pipeline for urinary volatiles disease detection: Diagnosing diabetes. *PLOS ONE*, 13(9):e0204425, September 2018. ISSN 1932-6203. doi: 10.1371/journal.pone.0204425. URL `http://dx.plos.org/10.1371/journal.pone.0204425`.

Sethu Vijayakumar, Aaron D'souza, Tomohiro Shibata, Jörg Conradt, and Stefan Schaal. Statistical learning for humanoid robots. *Autonomous Robots*, 12(1):55–69, 2002.

De G. Matthews, G. Alexander, Mark Van Der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *The Journal of Machine Learning Research*, 18(1):1299–1304, 2017.