Bayesian transfer learning between Gaussian process regression tasks

Milan Papež [a] and Anthony Quinn [a,b]
[a] Institute of Information Theory and Automation, Czech Academy of Sciences, Czech Republic
[b] Department of Electronic and Electrical Engineering, Trinity College Dublin, the University of Dublin, Ireland

***Abstract*—Bayesian knowledge transfer in supervised learning scenarios often relies on a complete specification and optimization of the stochastic dependence between source and target tasks. This is a critical requirement of completely modelled settings, which can often be difficult to justify. We propose a strategy to overcome this. The methodology relies on fully probabilistic design to develop a target algorithm which accepts source knowledge in the form of a probability distribution. We present this incompletely modelled setting in the supervised learning context where the source and target tasks are to perform Gaussian process regression. Experimental evaluation demonstrates that the transfer of the source distribution substantially improves prediction performance of the target learner when recovering a distorted nonparametric function realization from noisy data.**

***Index Terms*—Bayesian transfer learning, supervised learning, fully probabilistic design, incomplete modelling, Gaussian process regression.**

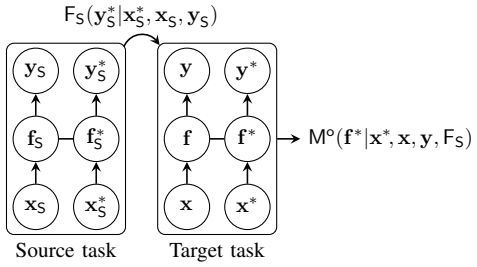

Fig. 1: Supervised source and target tasks process their source $(\mathbf{x}_\mathsf{S}, \mathbf{y}_\mathsf{S}, \mathbf{x}_\mathsf{S}^*)$ and target $(\mathbf{x}, \mathbf{y}, \mathbf{x}^*)$ known values to learn source $(\mathbf{f}_\mathsf{S}, \mathbf{f}_\mathsf{S}^*, \mathbf{y}_\mathsf{S}^*)$ and target $(\mathbf{f}, \mathbf{f}^*, \mathbf{y}^*)$ unknown values. Here, $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{f}$ are labelled inputs, outputs, and function evaluations, respectively, and the superscript $*$ denotes their unlabelled variants. The source task transfers the posterior predictive distribution of unlabelled outputs, F_S. The target task improves its performance by using an FPD-optimal, F_S-conditioned, posterior predictive distribution of unlabelled function values, M°.

I. Introduction

Transfer learning [1]—also known as multi-task learning [2]—has become a fundamental part of machine learning [3] and artificial intelligence [4]. The root principle behind transfer learning is to use knowledge provided by one (or more) source task(s) in order to improve performance and generalization capabilities of one (or more) related target task(s). In this paper, we restrict to Bayesian transfer learning [5] and propose an algorithm for transferring knowledge between supervised learning tasks. The central aim of supervised learning is to learn behaviour of a latent function from labelled pairs of input-output data, and then make predictions about its future values based on unlabelled input data [6], [7]. We specify our framework in the case where source and target tasks rely on a Gaussian process (GP) [8] to model their latent functions.

Bayesian transfer learning with GP priors has been successfully applied in diverse areas, including Bayesian optimization [9], medical time-series analysis [10], terrain modelling [11], and robot pose estimation [12]. The central question of transfer learning is how to specify the dependence structure between the tasks. GP-based approaches define a GP prior for each of the tasks and then model their correlations based on a linear dependence structure with fixed coefficients [13]–[15] or varying, input-dependent, coefficients [16], [17], which can be generalized under the framework of convolving processes [18]. To capture a nonlinear dependence structure, and foster a richer representation of relationships between the tasks, an additional GP layer can be applied to model the correlations [19].

The aforementioned techniques are all instances of *complete modelling*, where the inference of the target unknown quantities is improved by conditioning on knowledge given by crude source data. This requires a dependence structure between the source and target tasks to be specified. In contrast, this paper presents an approach where the target unknown quantities are additionally conditioned on knowledge in the form of a source probability distribution. Therefore, the challenge is to find a probabilistic model over a source probability distribution, which is assumed to be *unavailable*. This is an instance of *incomplete modelling*. The fully probabilistic design (FPD) [20], [21] is an optimal model-completion strategy rooted in the minimum cross-entropy principle [22]. Its primary purpose is to condition unknown quantities on knowledge—a source probability distribution in this paper—for which there is no probabilistic model [23]. The conditioning on a source probability distribution via the FPD approach is the key feature that releases the requirement of any dependence structure between source and target tasks to be specified.

The FPD-optimal model completion was originally used to design the target prior of unknown parameters conditioned on the source output predictor [24] and was later extended to various signal processing problems [25]–[28]. The present paper casts this methodology into the supervised learning context and designs the target posterior predictor of unlabelled function evaluations conditioned on the source posterior predictor of unlabelled outputs. We evaluate the prediction performance and the ability to reject imprecise source knowledge of different methods. The experimental results demonstrate that the proposed approach provides competitive performance and lower computational requirements compared to an alternative strategy relying on explicit dependence assumptions.

The research has been supported by GAČR grant 18-15970S.

II. FPD KNOWLEDGE TRANSFER BETWEEN SUPERVISED SOURCE AND TARGET TASKS

Let us consider a supervised learning problem specified by a conditional likelihood function in the following form:

$$\mathsf{F}(\mathbf{y}|\mathbf{f},\mathbf{x}), \tag{1}$$

where $\mathbf{y} \equiv (y_i \in \mathbb{R})_{i=1}^n$ are output data, $\mathbf{x} \equiv (x_i \in \mathbb{R}^{n_x})_{i=1}^n$ are input data, and $\mathbf{f} \equiv (f(x_i))_{i=1}^n$ are evaluations of a latent function, $f : \mathbb{R}^{n_x} \to \mathbb{R}$, at the inputs, $\mathbf{x}$. One of the primary objectives of supervised learning is to predict the unlabelled function values, $\mathbf{f}^*$, and outputs, $\mathbf{y}^*$, based on the labelled input-output data, $(\mathbf{x},\mathbf{y})$, and the unlabelled inputs, $\mathbf{x}^*$. That is, we seek the joint posterior predictor,

$$\mathsf{F}(\mathbf{y}^*,\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y}). \tag{2}$$

In this paper, we consider source and target tasks as shown by Fig. 1, where S denotes the source variables. Note that, to simplify notation, we do not use any decorator to distinguish the target variables. We assume that the target task does not have any direct knowledge about the quantities processed by the source task. The target task only receives the posterior predictor from the source task, $\mathsf{F}_\mathsf{S}(\mathbf{y}_\mathsf{S}^*|\mathbf{x}_\mathsf{S}^*,\mathbf{x}_\mathsf{S},\mathbf{y}_\mathsf{S})$. The central inference aim is to extend the joint posterior model of the target task, (2), to accommodate additional knowledge in the form of the source posterior predictor,

$$\mathsf{M}(\mathbf{y}^*,\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S}). \tag{3}$$

Nevertheless, a probability distribution over this distributional knowledge, F_S, is assumed to be unavailable. Therefore, the functional form of (3) is unknown, and our goal is to find a mechanism to condition (2) on F_S. In this paper, F and M denote specified (fixed-form) and unspecified (variational-form) distributions, respectively.

To transfer the source posterior predictor of unlabelled outputs, F_S, we constrain the functional form of the unknown joint model (3) as follows:

$$\begin{aligned}\mathsf{M}(\mathbf{y}^*,\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S}) \equiv \\ \mathsf{F}_\mathsf{S}(\mathbf{y}^*|\mathbf{x}_\mathsf{S}^*,\mathbf{x}_\mathsf{S},\mathbf{y}_\mathsf{S})\mathsf{M}(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S}),\end{aligned} \tag{4}$$

where we apply the conditional independence, and we restrict the unknown F_S-conditioned target posterior predictor of unlabelled outputs, $\mathbf{y}^*$, to be the source posterior predictor of unlabelled outputs computed at $\mathbf{y}_\mathsf{S}^* \equiv \mathbf{y}^*$, that is,

$$\mathsf{M}(\mathbf{y}^*|\mathbf{f}^*,\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S}) \equiv \mathsf{F}_\mathsf{S}(\mathbf{y}_\mathsf{S}^*|\mathbf{x}_\mathsf{S}^*,\mathbf{x}_\mathsf{S},\mathbf{y}_\mathsf{S})\big|_{\mathbf{y}_\mathsf{S}^*\equiv\mathbf{y}^*}. \tag{5}$$

This specification leaves $\mathsf{M}(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S})$ as the only variational quantity in (4), and enables us to delineate the knowledge-constrained set of possible models, as follows:

$$\begin{aligned}\mathsf{M} \in \mathbf{M} \equiv \{&\text{models (4) with } \mathsf{F}_\mathsf{S}(\mathbf{f}^*|\mathbf{x}_\mathsf{S}^*,\mathbf{x}_\mathsf{S},\mathbf{y}_\mathsf{S}) \\ &\text{fixed and } \mathsf{M}(\mathbf{f}|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S}) \text{ variational}\}.\end{aligned} \tag{6}$$

The joint posterior model (2) reflects behaviour of a supervised learning algorithm in the absence of any source knowledge. Therefore, we use it as the ideal (prescriptive) design

$$\begin{aligned}\mathsf{M}_\mathsf{I}(\mathbf{y}^*,\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y}) &\equiv \mathsf{F}(\mathbf{y}^*,\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y}) \\ &= \mathsf{F}(\mathbf{y}^*|\mathbf{f}^*,\mathbf{x}^*)\mathsf{F}(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y}).\end{aligned} \tag{7}$$

Here, we rely on the fact that $(\mathbf{x},\mathbf{y})$ does not provide more information about $\mathbf{y}^*$ when $(\mathbf{f}^*,\mathbf{x}^*)$ is known.

FPD is an optimal tool to condition a probability distribution on another probability distribution. FPD makes this possible by seeking an optimal model, M°, that belongs to the knowledge-constrained set, $\mathsf{M} \in \mathbf{M}$ (6), and incorporates preferences about M defined by an ideal model M_I (7). The distribution that is closest to M_I in the minimum Kullback-Leibler divergence (KLD) [29] sense,

$$\mathsf{M}^\circ(\mathbf{y}^*,\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S}) \equiv \operatorname*{argmin}_{\mathsf{M}\in\mathbf{M}} \mathcal{D}(\mathsf{M}||\mathsf{M}_\mathsf{I}), \tag{8}$$

is the FPD-optimal design, $\mathsf{M}^\circ \in \mathbf{M}$, which we seek. Here, the KLD from M to M_I is

$$\mathcal{D}(\mathsf{M}||\mathsf{M}_\mathsf{I}) = \mathsf{E}_\mathsf{M}\left[\log\left(\frac{\mathsf{M}}{\mathsf{M}_\mathsf{I}}\right)\right],$$

and the expected value under M is E_M.

Proposition 1. *The unknown model is a member of the knowledge constrained set,* $\mathsf{M} \in \mathbf{M}$ (6)*, and the ideal model* M_I *is* (7)*. Then, the solution of* (8)*—the FPD-optimal model—is*

$$\begin{aligned}\mathsf{M}^\circ(\mathbf{y}^*,\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S}) = \\ \mathsf{F}_\mathsf{S}(\mathbf{y}^*|\mathbf{x}_\mathsf{S}^*,\mathbf{x}_\mathsf{S},\mathbf{y}_\mathsf{S})\mathsf{M}^\circ(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S}),\end{aligned} \tag{9}$$

where

$$\begin{aligned}\mathsf{M}^\circ(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S}) &\propto \mathsf{F}(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y}) \\ &\times \exp\left\{\int \log \mathsf{F}(\mathbf{y}^*|\mathbf{f}^*,\mathbf{x}^*)\mathsf{F}_\mathsf{S}(\mathbf{y}^*|\mathbf{x}_\mathsf{S}^*,\mathbf{x}_\mathsf{S},\mathbf{y}_\mathsf{S})d\mathbf{y}^*\right\}.\end{aligned} \tag{10}$$

Proof. See Appendix A. □

Proposition 1 defines the approach to incorporate the source posterior predictor, F_S, into the target learning procedure. Note that (10) uses $\mathsf{F}_\mathsf{S}(\mathbf{y}^*|\mathbf{x}_\mathsf{S}^*,\mathbf{x}_\mathsf{S},\mathbf{y}_\mathsf{S})$ to perform the update from the posterior $\mathsf{F}(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y})$ to the FPD-optimal, F_S-conditioned, posterior $\mathsf{M}^\circ(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S})$. A specific computational flow for producing $\mathsf{F}(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y})$ is known as the *data learning* step. From this perspective, (10) can be seen as an additional step after the data learning step, which we refer to as the *transfer learning* step.

III. FPD KNOWLEDGE TRANSFER BETWEEN GAUSSIAN PROCESS TASKS

This section is concerned with the transfer learning scenario where the source and target task is to perform the GP regression. We specify the conditional likelihood function (1) as

$$\mathsf{F}(\mathbf{y}|\mathbf{f},\mathbf{x}) \equiv \mathcal{N}(\mathbf{y};\mathbf{f},\sigma^2 I_n), \tag{11}$$

where σ^2 is the variance of the noise corrupting the (independent and identically distributed) outputs, and I_n is the n-dimensional identity matrix. To find the joint posterior model

Algorithm 1: The FPD-optimal GP regression transfer

Input: $m(x)$, $k(x,x')$, σ^2, $\mathbf{x}$, $\mathbf{x}^*$, $\mathbf{y}$, $m_{\mathsf{S},n}(\mathbf{x}_\mathsf{S}^*)$

1 Data learning step:

2 Use m, k, σ^2, $\mathbf{x}$, $\mathbf{x}^*$, and $\mathbf{y}$ in (15) to compute $m_n(\mathbf{x}^*)$ and $k_n(\mathbf{x}^*,\mathbf{x}^*)$.

3 Transfer learning step:

4 Use $\mathbf{x}^*$, $m_n(\mathbf{x}^*)$, $k_n(\mathbf{x}^*,\mathbf{x}^*)$, and $m_{\mathsf{S},n}(\mathbf{x}_\mathsf{S}^*)$ in (18) to compute $m_n^\circ(\mathbf{x}^*)$ and $k_n^\circ(\mathbf{x}^*,\mathbf{x}^*)$.

Output: $m_n^\circ(\mathbf{x}^*)$, $k_n^\circ(\mathbf{x}^*,\mathbf{x}^*)$

(2), we need to specify a prior distribution over the latent function values, $\mathsf{F}(\mathbf{f}|\mathbf{x})$. We choose the Gaussian process prior for this purpose.

A stochastic process over a random function, $f:\mathbb{R}^{n_x}\to\mathbb{R}$, defines a probability distribution, $f\sim\mathsf{F}$, such that any finite collection of evaluation points, $\mathbf{x}$, induces a joint probability distribution over the function values, $\mathbf{f}$. In the case of Gaussian process, we define $f(x)\sim\mathcal{GP}(m(x),k(x,x'))$, where $m(x)$ and $k(x,x')$ are the mean and covariance functions, respectively, and the induced joint probability distribution is the multivariate Gaussian distribution,

$$\mathsf{F}(\mathbf{f}|\mathbf{x})\equiv\mathcal{N}(\mathbf{f};m(\mathbf{x}),k(\mathbf{x},\mathbf{x})). \tag{12}$$

Here, $m(\mathbf{x})$ is an n-dimensional vector, and $k(\mathbf{x},\mathbf{x})$ is an $n\times n$-dimensional matrix. Specific cases of m and k allow us to express prior beliefs in the behaviour of f, including smoothness and periodicity assumptions.

The data learning step of the source and target algorithms is based on the joint posterior model (2). We will require its marginals, which we now recall.

Lemma 1. *The probability distribution of output data is* (11)*, and the prior distribution of latent function evaluations is* (12)*. Then, the joint posterior model* (2) *admits the following marginal distributions:*

$$\mathsf{F}(\mathbf{y}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y})=\mathcal{N}(\mathbf{y}^*;m_n(\mathbf{x}^*),k_n(\mathbf{x}^*,\mathbf{x}^*)+\sigma^2 I_{n_*}), \tag{13}$$

$$\mathsf{F}(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y})=\mathcal{N}(\mathbf{f}^*;m_n(\mathbf{x}^*),k_n(\mathbf{x}^*,\mathbf{x}^*)), \tag{14}$$

where

$$m_n(x)=m(x)+k(x,\mathbf{x})\times\big(k(\mathbf{x},\mathbf{x})+\sigma^2 I_n\big)^{-1}\big(\mathbf{y}-m(\mathbf{x})\big), \tag{15a}$$

$$k_n(x,x')=k(x,x')-k(x,\mathbf{x})\times\big(k(\mathbf{x},\mathbf{x})+\sigma^2 I_n\big)^{-1}k(\mathbf{x},x'), \tag{15b}$$

with $k(x,\mathbf{x})$ denoting an n-dimensional vector.

Proof. The proof follows from standard calculus with Gaussian distributions, see, e.g. [6]. □

The target algorithm then additionally proceeds through the transfer learning step given by Proposition 1. We instantiate this in the present context in Proposition 2.

Proposition 2. *The target posterior predictor of unlabelled function values is* (14)*, and the source posterior predictor of unlabelled outputs has the same functional form as* (13)*, i.e.,*

$$\mathsf{F}_\mathsf{S}(\mathbf{y}^*|\mathbf{x}_\mathsf{S}^*,\mathbf{x}_\mathsf{S},\mathbf{y}_\mathsf{S})\equiv\mathcal{N}(\mathbf{y}^*;m_{\mathsf{S},n}(\mathbf{x}_\mathsf{S}^*),k_{\mathsf{S},n}(\mathbf{x}_\mathsf{S}^*,\mathbf{x}_\mathsf{S}^*)+\sigma_\mathsf{S}^2 I_{n_*}). \tag{16}$$

Then, the FPD-optimal posterior predictor (10) *is*

$$\mathsf{M}^\circ(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S})=\mathcal{N}(\mathbf{f}^*;m_n^\circ(\mathbf{x}^*),k_n^\circ(\mathbf{x}^*,\mathbf{x}^*)), \tag{17}$$

where

$$m_n^\circ(x)=m_n(x)+k_n(x,\mathbf{x}^*)\times\big(k_n(\mathbf{x}^*,\mathbf{x}^*)+\sigma^2 I_{n_*}\big)^{-1}\big(m_{\mathsf{S},n}(\mathbf{x}_\mathsf{S}^*)-m_n(\mathbf{x}^*)\big), \tag{18a}$$

$$k_n^\circ(x,x')=k_n(x,x')-k_n(x,\mathbf{x}^*)\times\big(k_n(\mathbf{x}^*,\mathbf{x}^*)+\sigma^2 I_{n_*}\big)^{-1}k_n(\mathbf{x}^*,x'). \tag{18b}$$

Proof. See Appendix B. □

Proposition 2 implies a computational procedure for the FPD-optimal *target* task as summarized in Algorithm 1. The *source* procedure is based only on the data step, i.e., line 2 of Algorithm 1, with the associated quantities and the GP shaping functions decorated by S.

IV. Experiments

To illustrate the key features of the proposed FPD-optimal algorithm, we focus on a simple example where the source and target tasks are to regress a nonlinear function observed in Gaussian noise. We compare the following methods: *S*ource algorithm with *N*o *T*ransfer (*SNT*); target algorithm with *N*o *T*ransfer (*NT*); target algorithm based on the *L*inear *M*odel of *C*oregionalization (*LMC*) method [14], where the correlation coefficient, ρ, between the source and target tasks is specified, see [30] for details; and target algorithm using the *FPD*-optimal transfer (*FPD*) given in Algorithm 1. We consider that the source and target output data are generated according to

$$\mathsf{F}(\mathbf{y}_\mathsf{S}|\mathbf{f}_\mathsf{S},\mathbf{x}_\mathsf{S})\equiv\textstyle\prod_{i=1}^{n_\mathsf{S}}\mathcal{N}(y_{\mathsf{S},i};f_\mathsf{S}(x_{\mathsf{S},i}),\sigma_\mathsf{S}^2),$$
$$\mathsf{F}(\mathbf{y}|\mathbf{f},\mathbf{x})\equiv\textstyle\prod_{i=1}^{n}\mathcal{N}(y_i;f(x_i),\sigma^2),$$

where $f_\mathsf{S}(x)=\frac{x}{2}\sin(x)$, $\sigma_\mathsf{S}^2=0.01$, $f(x)=\frac{x^{0.95}}{1.9}\sin(x)$, and $\sigma^2=1$. The source and target input data are simulated by

$$\mathbf{x}\sim\mathcal{U}_n(-10,10),$$
$$\mathbf{x}_\mathsf{S}\sim\mathcal{U}_{n_\mathsf{S}}(-10,10).$$

Here, $\mathcal{U}_n(a,b)$ is the uniform distribution on the n-fold open interval (a,b). This scenario reflects a situation where f is a moderately distorted version of f_S, which is observed in a higher noise. In such a case, we would like to know if the source procedure with high-quality data can assist the target procedure with low-quality data. The prior mean functions, m, m_S, are given by the zero vector, and the covariance functions, k, k_S, are both specified as the squared exponential kernel [8],

$$k(x,x')=\sigma_f^2\exp\Big\{-\tfrac{1}{2}\tfrac{||x-x'||^2}{l^2}\Big\},$$

where σ_f^2 is the signal variance, l is the length-scale, and $||\cdot||$ denotes the euclidean norm. We set $\sigma_\mathsf{f}^2=\sigma_f^2=2$ and $l_\mathsf{S}=l=0.5$ for all algorithms. We are interested in the prediction

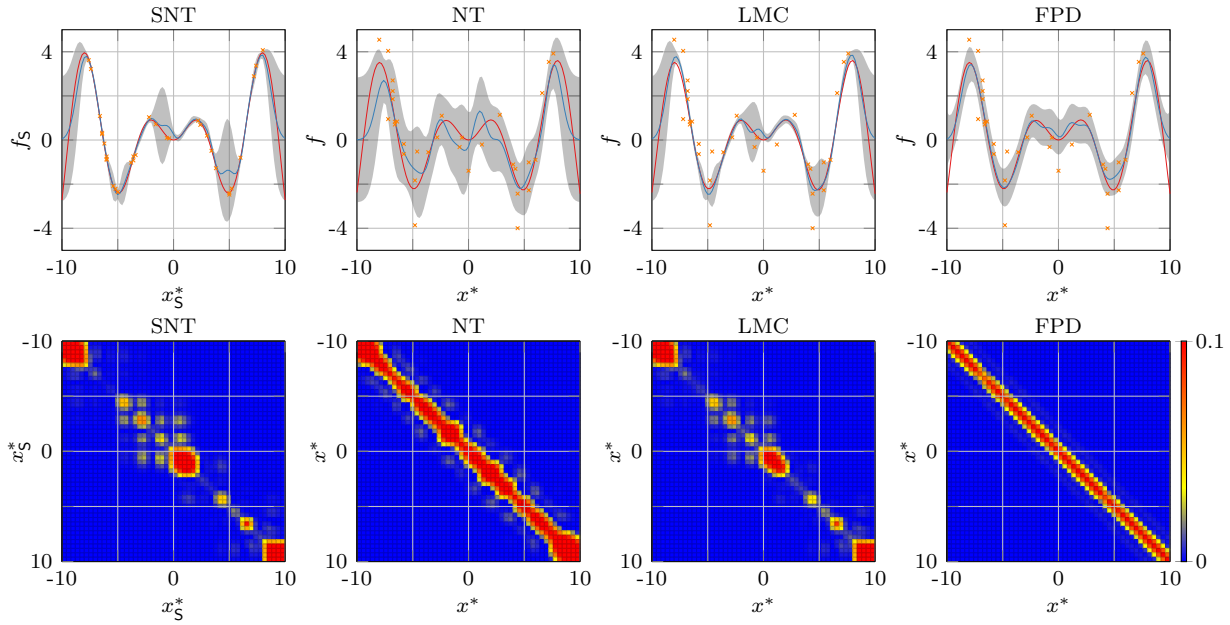

Fig. 2: Top: the prediction performance at the unlabelled input data $\mathbf{x}^*$. Here, (——) is the true function, $f(x^*)$, (×) are the labelled (noisy) outputs, y, (——) is the posterior predictive mean function, $m_n(x^*)$, and ( ) is the posterior predictive 2σ-region, $\pm 2\sqrt{k_n(x^*, x^*)}$. Bottom: the covariance function evaluated at the unlabelled inputs for the SNT algorithm, $k_{\mathsf{S},n}(x^*_{\mathsf{S}}, x^*_{\mathsf{S}})$, the NT and LMC algorithms, $k_n(x^*, x^*)$, and the FPD algorithm, $k^{\circ}_n(x^*, x^*)$.

performance of the various methods for unlabelled input data $\mathbf{x}^*$ generated on the closed interval $[-10, 10]$ with the step-size 0.2. We evaluate the error norm (EN) between the true function values at $\mathbf{x}^*$, $\mathbf{f}^* \equiv (f(x^*_i))_{i=1}^{n_*}$, and their posterior predictive mean estimate, $m_n(\mathbf{x}^*)$,

$$\text{EN} \equiv ||\mathbf{f}^* - m_n(\mathbf{x}^*)||.$$

In the first experiment, whose results are depicted in Fig. 2, we illustrate the prediction performance of the compared methods for $n \equiv n_{\mathsf{S}} \equiv 32$. The top row of Fig. 2 reveals that—despite processing the imprecise target output data—the FPD algorithm takes advantage of the source posterior predictive distribution, F_{S}, and—compared to the NT algorithm—it significantly recovers the shape of the latent function and reduces the 2σ-region. The LMC algorithm—with its correlation coefficient set to $\rho = 1$ (extreme value)—performs moderately better than the FPD algorithm. The bottom row of Fig. 2 presents the covariance matrices associated with the results above, providing more details of the uncertainty representation.

Transfer learning is useful mainly when a source algorithm processes either more precise data or simply more data of the same precision, compared to the target algorithm. We demonstrate this with two experiments depicted in the top row of Fig. 3, where the EN of various methods changes as a function of the variance of the source output data, σ^2_{S} (top-left), and the number of source data, n_{S} (top-right). The NT algorithm does not depend on these parameters and thus defines a reference EN level for comparing the remaining methods. If any algorithm produces an EN below or above this reference level, we say that it provides *positive* or *negative* transfer, respectively. When any algorithm saturates at the EN level of the NT algorithm, we say that it achieves *robust* transfer. The FPD algorithm switches its behaviour near the intersection points $\sigma^2_{\mathsf{S}} = \sigma^2$ and $n_{\mathsf{S}} = n$, yielding positive transfer for (approximately) $\sigma^2_{\mathsf{S}} < \sigma^2$ and $n_{\mathsf{S}} > n$, and negative transfer for $\sigma^2_{\mathsf{S}} > \sigma^2$ and $n_{\mathsf{S}} < n$. We explain this behaviour in Section V. The LMC algorithm gives positive transfer for $\sigma^2_{\mathsf{S}} < 10^2$ and all n_{S}, achieving robust transfer for $\sigma^2_{\mathsf{S}} > 10^2$.

The next experiment (bottom-left of Fig. 3) evaluates the computational time of the compared methods by changing the number of target data according to $n = 2^i$ for $i = 1, \ldots, 9$. We see that the proposed FPD algorithm is computationally more efficient and offers an improved prediction performance compared to the LMC algorithm (again, with $\rho = 1$) when $i = 1, \ldots, 7$. For $i = 8$ and $i = 9$, the FPD and LMC algorithms perform similarly.

The final experiment (bottom-right of Fig. 3) demonstrates the impact of the correlation coefficient, ρ, on the EN of the LMC algorithm for source knowledge of varying quality. We observe that values of ρ other than $\rho \to 1$ cause the EN of the LMC algorithm to be worse compared to the FPD algorithm. More importantly, we see that the LMC algorithm yields negative transfer for $\rho < 0$ even for high-quality source knowledge. Note that if $\rho = 0$ (uncorrelated source and target tasks), the LMC algorithm recovers the EN of the NT algorithm. The proposed FPD algorithm does not depend on any structural (ρ-like) assumptions, and yet it is able to deliver very competitive performance compared to the LMC algorithm when the source knowledge is of high quality.

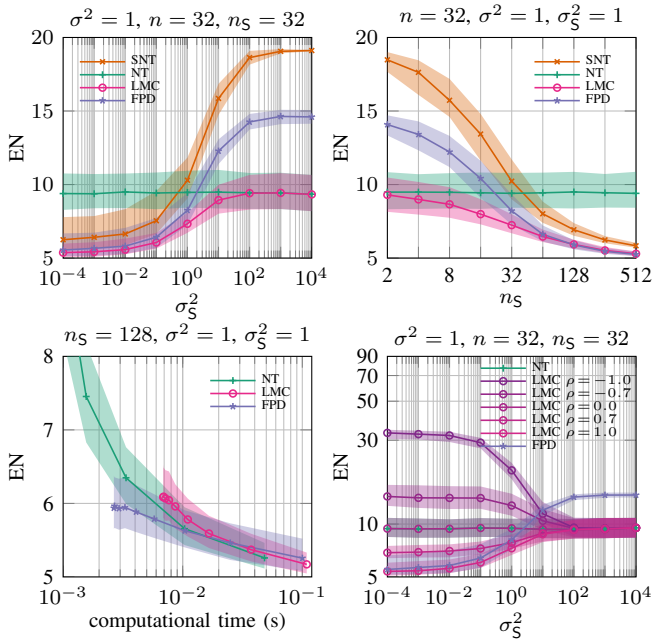

Fig. 3: Top-left: the error norm (EN) versus the source output data variance, σ^2_S. Top-right: the EN versus the number of source data, n_S. Bottom-left: the EN versus the computational time in seconds (influenced by the number of target data). Bottom-right: the EN versus the source output data variance, σ^2_S, for various correlation coefficients of the LMC algorithm, ρ. The results are averaged over 1000 independent simulation epochs, with the solid line and shaded area being the median and interquartile range, respectively.

V. Discussion

In the first part of this section, we would like to discuss the following question: if the true function lies in the 2σ-region for the SNT and NT algorithms—as shown on top of Fig. 2—why is this behaviour not fully preserved by the LMC and FPD algorithms? In the case of the LMC algorithm, surprisingly, this happens close to the points where both the SNT and NT algorithms process output data, i.e., close to $x^* = -8$ and $x^* = 8$. Although, such behaviour occurred only occasionally during our experiments—and is still consistent due to the fact that the true function is certainly in the 3σ-region—this situation demonstrates that the LMC algorithm can deliver overconfident predictions compared to the SNT and NT algorithms. We observed that this is often the case when $\rho \to 1$, which is, however, the value where the LMC algorithm provides the best EN, see the bottom-right of Fig. 3. Importantly, this issue of overconfident predictions diminishes for lower ρ, but the FPD algorithm then outperforms the LMC algorithm (Fig. 3). In the case of the FPD algorithm, the true function lies outside the 2σ-interval where neither the SNT nor NT algorithm processes any output data, i.e., close to $x^* = -10$ and $x^* = 10$. This is more sensible behaviour than in the LMC case. Moreover, a closer look at bottom of Fig. 2 reveals that the uncertainty representation of the FPD algorithm is rather uniform compared to the LMC algorithm which adjusts its uncertainty according to the number of data processed. This behaviour of the FPD algorithm is a result of the loss of the covariance function of the source posterior predictor, $k_{\mathsf{S},n}$, during the transfer learning step. The source covariance function is lost when computing the exponential structure of (10), see also (19) in Appendix B. The consequence of this can best be seen in (18), where only the source mean function, $m_{\mathsf{S},n}$, is present but the source covariance function, $k_{\mathsf{S},n}$, is missing. The same reasoning explains the fact that the FPD algorithm suffers from negative transfer for (approximately) $\sigma^2_\mathsf{S} > \sigma^2$ and $n_\mathsf{S} < n$, as illustrated in the top row of Fig. 3. The kernel hyperparameters were set to fixed values for all the investigated methods in order to ensure a fair comparison. However, we expect that optimizing these hyperparameters can address the problems associated with uncertainty representation of both the LMC and FPD algorithms. This will form the agenda of future work.

Next, we comment on the choice of the correlation coefficient, ρ, of the LMC method, as illustrated in bottom-right of Fig. 2. We see that this choice is critical, leading not only to a slightly worse EN compared to the FPD algorithm, but, crucially, to negative transfer even for high-quality source knowledge. In practice, ρ can be treated as an additional hyperparameter which needs to be optimized [14]. There is no such hyperparameter dependence in the proposed FPD algorithm. Notwithstanding this, the FPD algorithm is competitive with the LMC algorithm. Recall that the LMC algorithm requires complete modelling of the source and target dependence, and directly processes the raw source data. In contrast, our FPD algorithm (resulting from the Bayesian transfer in Fig. 1) only transfers the sufficient statistics of F_S. Furthermore, the essence of this approach is to avoid explicit joint modelling assumptions between the source and target tasks, which, anyway, are very hard to propose in most cases. In addition, ρ will increase the computational cost of marginalizing the hyperparameters in the LMC algorithm via Markov chain Monte Carlo methods. This added complexity is avoided in the FPD algorithm.

VI. Conclusion

This paper has presented FPD-optimal Bayesian transfer learning in the context of Gaussian process regression. The proposed FPD algorithm is a consequence of optimal model completion in the context where the stochastic dependence between the source and target tasks is not specified. As illustrated by the experimental evidence, the FPD algorithm can offer competitive (and sometimes even better) prediction performance, lower computational requirements, and fewer hyperparameters to tune compared to the LMC approach. The latter relies on explicit—and often brittle—dependence assumptions. However, the FPD algorithm suffers from negative transfer when the source knowledge is imprecise compared to the target knowledge. We encountered similar behaviour when developing transfer learning algorithms in Bayesian filtering applications [25], [26]. There, we were successful in resolving this issue by introducing an auxiliary variable augmentation [27]. Our preliminary investigations suggest that the same solution can also be successful in the present context. We will provide more details in a future paper.

APPENDIX

A. Proof of Proposition 1

After substituting (4) and (7) into (8), we obtain:

$$\begin{aligned}\mathcal{D}(\mathsf{M}||\mathsf{M}_\mathsf{I}) &= \int \mathsf{F}_\mathsf{S}(\mathbf{y}^*|\mathbf{x}^*_\mathsf{S},\mathbf{x}_\mathsf{S},\mathbf{y}_\mathsf{S})\mathsf{M}(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S}) \\ &\times \log\left(\frac{\mathsf{F}_\mathsf{S}(\mathbf{y}^*|\mathbf{x}^*_\mathsf{S},\mathbf{x}_\mathsf{S},\mathbf{y}_\mathsf{S})\mathsf{M}(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S})}{\mathsf{F}(\mathbf{y}^*|\mathbf{f}^*,\mathbf{x}^*)\mathsf{F}(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y})}\right) d\mathbf{y}^* d\mathbf{f}^* \\ &= \int \mathsf{M}(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S}) \\ &\times \log\left(\frac{\mathsf{M}(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S})}{\mathsf{M}^\circ(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y},\mathsf{F}_\mathsf{S})}\right) d\mathbf{f}^* - \mathcal{H}_{\mathsf{F}_\mathsf{S}} - \log c_{\mathsf{M}^\circ},\end{aligned}$$

where $\mathcal{H}_{\mathsf{F}_\mathsf{S}}$ is the differential entropy of F_S,

$$\mathcal{H}_{\mathsf{F}_\mathsf{S}} = -\int \mathsf{F}_\mathsf{S}(\mathbf{y}^*|\mathbf{x}^*_\mathsf{S},\mathbf{x}_\mathsf{S},\mathbf{y}_\mathsf{S}) \log \mathsf{F}_\mathsf{S}(\mathbf{y}^*|\mathbf{x}^*_\mathsf{S},\mathbf{x}_\mathsf{S},\mathbf{y}_\mathsf{S}) d\mathbf{y}^*,$$

and c_{M° is the normalizing constant,

$$\begin{aligned}c_{\mathsf{M}^\circ} &= \int \mathsf{F}(\mathbf{f}^*|\mathbf{x}^*,\mathbf{x},\mathbf{y}) \\ &\times \exp\left\{\int \log \mathsf{F}(\mathbf{y}^*|\mathbf{f}^*,\mathbf{x}^*)\mathsf{F}_\mathsf{S}(\mathbf{y}^*|\mathbf{x}^*_\mathsf{S},\mathbf{x}_\mathsf{S},\mathbf{y}_\mathsf{S}) d\mathbf{y}^*\right\} d\mathbf{f}^*. \quad \square\end{aligned}$$

B. Proof of Proposition 2

Substituting (11) and (16) into the exponential term of (10) leads to

$$\exp\left\{\mathsf{E}_{\mathsf{F}_\mathsf{S}}[\log \mathsf{F}(\mathbf{y}^*|\mathbf{f}^*,\mathbf{x}^*)]\right\} \propto \mathcal{N}\big(m_{\mathsf{S},n}; \mathsf{f}^*, \sigma^2 I_{n_*}\big). \quad (19)$$

Then, the product of (14) and (19) yields

$$\mathcal{N}\left(\begin{bmatrix} m_{\mathsf{S},n} \\ \mathbf{f}^* \end{bmatrix}; \begin{bmatrix} m_n(\mathbf{x}^*) \\ m_n(\mathbf{x}^*) \end{bmatrix}, \begin{bmatrix} k_n(\mathbf{x}^*,\mathbf{x}^*) + \sigma^2 I_{n_*} & k_n(\mathbf{x}^*,\mathbf{x}^*) \\ k_n(\mathbf{x}^*,\mathbf{x}^*) & k_n(\mathbf{x}^*,\mathbf{x}^*) \end{bmatrix}\right),$$

which admits the conditional distribution (17) with the moments (18). $\square$

REFERENCES

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[2] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114*, 2018.

[3] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, pp. 9, 2016.

[4] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowledge-Based Systems*, vol. 80, pp. 14–23, 2015.

[5] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pp. 242–264. IGI Global, 2010.

[6] K. P. Murphy, *Machine learning: A probabilistic perspective*, MIT Press, 2012.

[7] C. M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.

[8] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*, MIT Press, 2006.

[9] K. Swersky, J. Snoek, and R. P. Adams, "Multi-task Bayesian optimization," in *Advances in neural information processing systems*, 2013, pp. 2004–2012.

[10] L.-F. Cheng, G. Darnell, B. Dumitrascu, C. Chivers, M. E. Draugelis, K. Li, and B. E. Engelhardt, "Sparse multi-output Gaussian processes for medical time series prediction," *arXiv preprint arXiv:1703.09112*, 2017.

[11] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Non-stationary dependent Gaussian processes for data fusion in large-scale terrain modeling," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1875–1882.

[12] K. M. A. Chai, Christopher K. I. Williams, S. Klanke, and S. Vijayakumar, "Multi-task Gaussian process learning of robot inverse dynamics," in *Advances in neural information processing systems*, 2008, pp. 265–272.

[13] Y. W. Teh, M. Seeger, and M. I. Jordan, "Semiparametric latent factor models," in *AISTATS 2005-Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[14] E. V. Bonilla, K. M. Chai, and C. Williams, "Multi-task Gaussian process prediction," in *Advances in neural information processing systems*, 2008, pp. 153–160.

[15] T. V. Nguyen and E. V. Bonilla, "Collaborative multi-output Gaussian processes," in *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2014, pp. 643–652.

[16] A. G. Wilson, D. A. Knowles, and Z. Ghahramani, "Gaussian process regression networks," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*. Omnipress, 2012, pp. 1139–1146.

[17] T. Nguyen and E. V. Bonilla, "Efficient variational inference for Gaussian process regression networks," in *Artificial Intelligence and Statistics*, 2013, pp. 472–480.

[18] M. A. Álvarez and N. D. Lawrence, "Computationally efficient convolved multiple output Gaussian processes," *Journal of Machine Learning Research*, vol. 12, no. May, pp. 1459–1500, 2011.

[19] A. Boustati and R. S. Savage, "Multi-task learning in deep Gaussian processes with multi-kernel layers," 2019.

[20] M. Kárný, "Towards fully probabilistic control design," *Automatica*, vol. 32, no. 12, pp. 1719–1722, 1996.

[21] M. Kárný and T. Kroupa, "Axiomatisation of fully probabilistic design," *Information Sciences*, vol. 186, no. 1, pp. 105–113, 2012.

[22] J. Shore and R. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Transactions on Information Theory*, vol. 26, no. 1, pp. 26–37, 1980.

[23] A. Quinn, M. Kárný, and T. V. Guy, "Fully probabilistic design of hierarchical Bayesian models," *Information Sciences*, vol. 369, pp. 532–547, 2016.

[24] A. Quinn, M. Kárný, and T. V. Guy, "Optimal design of priors constrained by external predictors," *International Journal of Approximate Reasoning*, vol. 84, pp. 150–158, 2017.

[25] C. Foley and A. Quinn, "Fully probabilistic design for knowledge transfer in a pair of Kalman filters," *IEEE Signal Processing Letters*, vol. 25, no. 4, pp. 487–490, 2018.

[26] M. Papež and A. Quinn, "Dynamic Bayesian knowledge transfer between a pair of Kalman filters," in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2018.

[27] M. Papež and A. Quinn, "Robust Bayesian transfer learning between Kalman filters," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2019.

[28] L. Jirsa, L. Pavelkova, and A. Quinn, "Knowledge transfer in a pair of uniformly modelled Bayesian filters," in *2019 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. IEEE, 2019.

[29] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[30] K. M. Chai, "Generalization errors and learning curves for regression with multi-task Gaussian processes," in *Advances in neural information processing systems*, 2009, pp. 279–287.