Fast Transfer Gaussian Process Regression with Large-Scale Sources

Bingshui Da[1,2], Yew-Soon Ong[1], Abhishek Gupta[1],
Liang Feng[3], Haitao Liu[4]
[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2]SAP Innovation Center Network, Machine Learning, Singapore
[3]College of Computer Science, Chongqing University, China
[4]Rolls-Royce@NTU Corporate Lab, Nanyang Technological University, Singapore
DA0002UI@e.ntu.edu.sg, {ASYSONG,ABHISHEKG}@ntu.edu.sg,
LIANGF@cqu.edu.cn, HTLIU@ntu.edu.sg

Abstract

In transfer learning, we aim to improve the predictive modeling of a target output by using the knowledge from some related source outputs. In real-world applications, the data from the target domain is often precious and hard to obtain, while the data from source domains is plentiful. Thus, since the complexity of Gaussian process based multi-task/transfer learning approaches grows cubically with the total number of source+target observations, the method becomes increasingly impractical for large ($> 10^4$) source data inputs even with a small amount of target data. In order to scale known transfer Gaussian processes to large-scale source datasets, we propose an efficient aggregation model in this paper, which combines the predictions from distributed (small-scale) local experts in a principled manner. The proposed model inherits the advantages of single-task aggregation schemes, including efficient computation, analytically tractable inference, and straightforward parallelization during training and prediction. Further, a salient feature of the proposed method is the enhanced expressiveness in transfer learning - as a byproduct of flexible inter-task relationship modelings across different experts. When deploying such models in real-world applications, each local expert corresponds to a lightweight predictor that can be embedded in *edge* devices, thus catering to cases of online on-mote processing in fog computing

settings.

Keywords: Transfer Learning, Large-Scale, Gaussian Process, Aggregation Models, Edge Intelligence

1. Introduction

Transfer learning (or inductive transfer) aims at improving the learning of a particular *target* task by utilizing the knowledge available from related *source* domains [1, 2]. It is widely applied in cases when the data from the target domain is scarce, while the data from the source domain is extensive. In today's times of data democratization, where we have relatively easy access to diverse information streams, the opportunity to harness the knowledge from related examples to enhance performance on a new target task is immense. As a result, the notion of transfer learning has recently attracted significant research attention, with success stories reported in a variety of areas, including, aerospace engineering [3], computer vision [4], natural language processing [5], affective computing [6], general black-box optimization problem-solving [7, 8], fuzzy systems [9, 10], etc.

Gaussian process (GP) [11] is a well-established method for inference on functions, and has received research attention in various scenarios, such as regression [12], classification [13], optimization [14, 15], data visualization [16], etc. It provides a principled probabilistic kernel learning framework, and closed-form inference that can be directly performed by applying Bayes' rule. It is noted that in transfer learning, a naive approach has been to rely on the practitioner to possess some *a priori* understanding of the suitability of its application in a given scenario. However, it may be deceptively difficult to ascertain the relationship between distinct domains, as a result of which *negative transfer* [17] is a major threat for the generalization performance of transfer learning models. However, multi-task/transfer GPs [18, 19, 20, 21] can, in theory, avoid this risk by automatically learning task relations concealed in data from various domains; thus adaptively transferring knowledge from source domain to target domain.

Generally in real-world applications, it is possible for source data to be collected and accumulated over time. For example, with the onset of the Internet of Things (IoT) and cyber-physical systems, environmental data that is collectable via cheap sensors can be abundantly gathered and stored. On the other hand, distinct but associated environmental data that requires high

precision (high cost) sensors may be extremely scarce. In this setting, it may be possible to exploit the knowledge embedded in the cheap source datasets to enhance predictive accuracy for high precision target points. Importantly, with the increasing amount of collected source data, it is noted that existing transfer GP (TGP) approaches become progressively impractical, due to the cubically scaling complexity of model learning. Yet it is found that TGPs catering to large-scale source datasets has received little attention in the literature.

While recent works in the general theme of multi-task/transfer GPs can be found in [18, 22, 23, 24], these methods generally focus on applying low-rank approximation to the full GP covariance matrix by selecting a set of inducing variables. Although the complexity during inference can be reduced to scale linearly to the number of observations, the major drawbacks of these kinds of methods are that: (1) finding the locations for the inducing inputs to best approximate the posterior is challenging, and (2) the predictions do not interpolate the observation points in local regions - specifically, for quick-varying functions with significant local structures, it may be difficult to find any trust-worthy representation more compact than the complete set of the local training observations. In other words, there clearly exists a research gap in the field of adaptive TGPs to decrease the computational complexity with large-scale source data inputs while simultaneously maintaining similar local expressiveness of the original full TGP model.

It is found that in single-task learning, aggregation models, such as Bayesian committee machine [25], product of experts [26, 27], mixture of experts [28, 29], and so on, have been well-studied to reduce the computational burden of large-scale GPs. These methods generally involve partitioning the training inputs into local blocks or clusters, then modeling each block with an independent GP as a local expert. If the blocks are spatially localized, the overall model corresponds to a covariance function that imposes independence between output values in different regions of the input space. In comparison to the sparse approximation methods, the aggregation models (1) do not require any additional inducing or variational parameters, (2) allow straightforward parallelization to distribute the computations on individual experts, and (3) maintain similar local expressiveness as the full GP for functions with notable local structures. However, naively applying these aggregation methods into TGPs may not be practical due to insufficient target inputs. Target data is scarce and precious in transfer learning, and hence partitioning the target inputs will probably cause significant infor-

mation loss globally. Therefore, in this paper, we propose a novel factorized training strategy for transfer learning, in which the target data is fully utilized within each local expert, even as the advantages of lower computational complexity and straightforward parallelization of classical aggregation models is retained. In particular, armed with a set of trained local experts, we propose a principled method, labeled as *transfer Bayesian Committee Machine* (Tr-BCM), to combine their respective predictions. It will be demonstrated that the proposed model fully utilizes the scarce target inputs to ensure the predictive performance of each local expert is superior to the performance of a single-task GP trained on target data only. As far as we know, this is the first paper introducing aggregation models in the setting of transfer learning. The efficacy of Tr-BCM is verified on toy examples as well as real-world applications.

As a notable byproduct of the aggregation model-based approach, we find that Tr-BCM offers enhanced expressiveness in multi-task/transfer GPs by enabling the capture of localized inter-task relationships. According to recent studies [30, 31], the efficacy of enhancing model expressiveness has been well-established. In the context of knowledge transfer in particular, while a given pair of tasks may be resolved as being globally uncorrelated, there may exist local subspaces characterized by strong correlation. Nevertheless, naively extending a full TGP model to learn localized source-target relationships is proved in this paper to have no guarantee to produce a positive semi-definite (PSD) covariance matrix. However, such issues can be easily circumvented by applying aggregation models as shall be illustrated later on. What is more, it is revealed that this salient feature of Tr-BCM applies with little/no modification to the case of multi-source transfer learning problems as well; thereby highlighting the generality of the proposed method in practice.

To summarize, the following salient features make the proposed model an attractive proposition for the domain of transfer learning:

- We propose a new factorized training strategy and principled aggregation model, namely Tr-BCM, for transfer learning, in order to accelerate full TGP with large-scale source inputs. The theoretical behaviors of the proposed Tr-BCM model in comparison to other naive extensions of model aggregation schemes are analyzed in detail.

- Flexible/non-uniform source-target similarity capture is made possible through the proposed Tr-BCM. Therefore, the expressiveness of the

proposed model is increased, and negative transfer is mitigated if the source-target similarity indeed varies drastically in the input space.

- We further propose a hierarchical structure to extend Tr-BCM for dealing with transfer learning problems with multiple sources. Thus, the practical generality of Tr-BCM is greatly increased.

- Finally, when applying Tr-BCM in real-world applications, each local expert corresponds to a lightweight predictor that can be embedded in *edge* devices, thus catering to cases of online on-mote processing [32, 33].

For a detailed exposition about the proposed model and the empirical investigation of its efficacy, the rest of the paper is organized as follows. In Section 2, we briefly review related work in the area of adaptive transfer/multi-task learning dealing with large-scale source inputs. After that, in Section 3, we introduce the concept of *edge intelligence* in the emerging *fog computing* paradigm [33], which serves as one of the key practical motivations for our proposal of aggregation models in the setting of transfer learning. Next, we offer a general introduction of TGP in Section 4, following which we present our proposed Tr-BCM strategy in order to decrease the computational burden of traditional TGP in Section 5. Non-uniform source-target relationship capture and multi-source transfer learning problems are studied in Section 6 and 7, respectively. In the empirical studies of Section 8, numerical experiments on real-world datasets highlight the benefits of the proposed method in comparison to existing transfer learning approaches.

2. Related Work

In this section, we first briefly recap the recent progress towards TGPs, then introduce different acceleration methods proposed in the literature to reduce the computational complexity to scale GP-based methods.

The idea of transfer learning is that information shared between the tasks leads to improved generalization performance on the target task in comparison to learning the target task individually [1, 18]. This idea is closely related to multi-task learning - which aims to improve generalization performance across multiple tasks at once. When using a GP for multiple distinct but related outputs, the problem often reduces to developing a prior (mainly determined by the covariance function) that expresses correlations between

the outputs. A number of different covariance functions for multi-task GPs have been proposed in [34, 18, 22, 35]. For example, in [18, 36], the authors encode the inter-task correlations in a PSD matrix, with the entry in the ith and $j$the column capturing the degree of relatedness between the ith and jth tasks. Detailed reviews on the subject have recently been published in [37, 38].

As opposed to symmetric transfer in multi-task learning, less research attention has been focused on asymmetric transfer via TGP. In [19], Cao et al. proposed TGP to adaptively transfer knowledge from a single source task to improve the performance of the target task by learning source-target similarity. Leen et al. [35] combines the latent decision margins of multiple GPs operating on the source tasks with the latent decision margin of the target task. In [39], Wang et al. proposed to model the source task, the target task, and the offset between using Gaussian process models, based on the assumption that there is some smoothness in the offset over the input domain. Taking advantage of deep GP, Kandemir [40] adopted a two-layer feed-forward deep GP [41] as the task learner of source and target domains. More recently, Wagle and Frew [20] proposed forward adaptive TGP, in which the training of source task is decoupled. In [21], Wei et al. studied multi-source transfer learning problems by *stacking* all the source and target models. A similar stacking procedure was also adopted in [42], with the transfer learning GP model applied to enhance the efficiency of Bayesian optimization. Further, Wistuba et al. [43] proposed to combine source and target Gaussian process models via ensemble techniques, thus the final model is a weighted sum of all surrogates.

However, one of the major problems of the above transfer learning methods is the computational complexity during training and prediction, which scales cubically with the number of observations. Given the huge amount of observations that can be available from source tasks, practical use of such methods becomes problematic, even with a small number of target training data. To reduce the overwhelming computational burden, different acceleration methods have been proposed in the literature, primarily focusing on symmetric transfer in multi-task GP, taking advantage of low-rank approximation to the full covariance matrix. The pioneering work proposed in [18] uses Nyström approximation of the kernel matrix in the joint marginal likelihood. Later, several different low-rank approximation methods have been proposed in [22, 24], which are strongly related to the partially independent training conditional [44] and fully independent training conditional

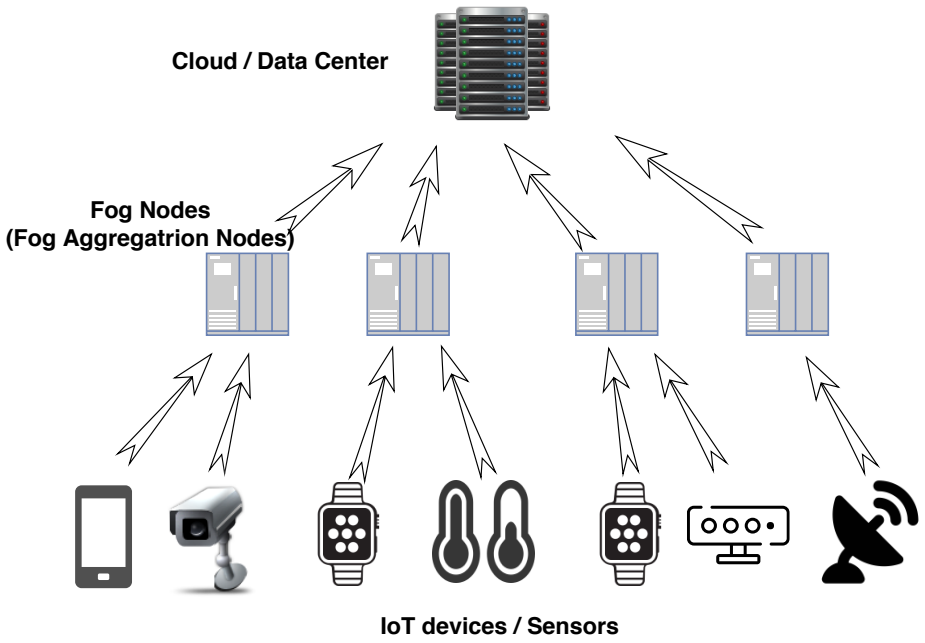

Figure 1: The Fog extends the Cloud closer to (edge) devices producing data.

[45] approximations for a single-task GP. These approximation methods cut the computational complexity to scale linearly with the number of observations. More recently, various methods taking advantage of variational inference [46, 47, 48] are proposed in the literature.

Nevertheless, it is noted that low-rank approximation and variational inference based methods generally have a key limitation: for quick-varying functions with significant local structures, the complete set of local training observations may be the most compact representation than any low-rank approximation. Recent study in [49] shows, in single-task GP models, similar local expressiveness can be maintained by using aggregation models. However, to the best of our knowledge, no aggregation models have been proposed to tackle transfer learning problems. Therefore, in this paper, we will propose a principled aggregation model to deal with transfer learning problems efficiently, especially those with large-scale source data.

3. Practical Motivation in Fog Computing

The IoT is capable of generating an unprecedented volume and variety of data via cheap sensors. In the conventional cloud computing paradigm,

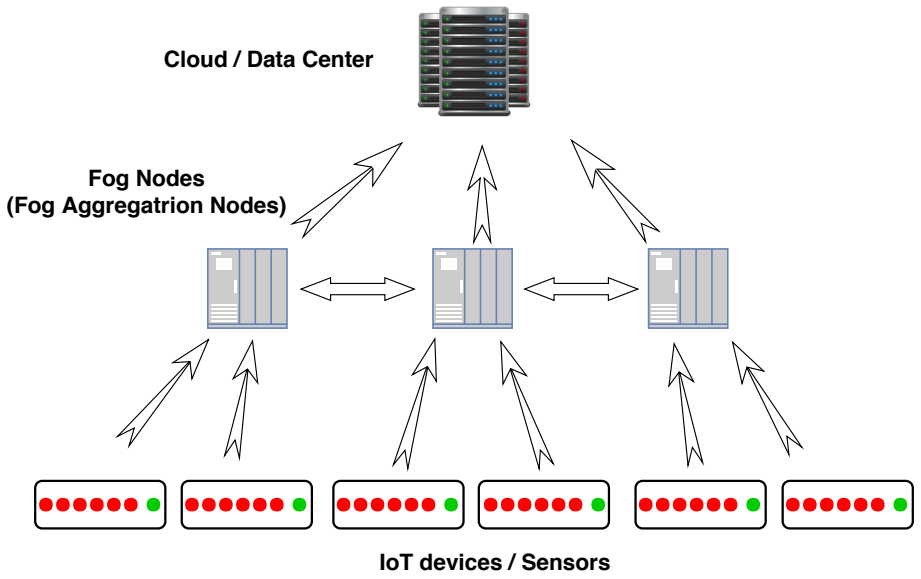

Figure 2: An abstraction of Fog computing paradigm for aggregated transfer learning models. Red dots indicate easily collected source data inputs, while green dots indicate the scarce target data inputs. To apply the proposed Tr-BCM, only small amount of target inputs need be broadcast over the fog (aggregation) nodes.

all the data is transmitted to the cloud/data center for processing, thus requiring huge bandwidth cost. More importantly, by the time the data makes its way to the cloud for analysis, the opportunity to make predictive analysis on it might be gone or is expected to experience observable delay. With this in mind, it is argued that the ideal place to analyze most IoT data is near (edge) devices that produce and act on that data. Consequently, the Fog computing (or Edge computing) paradigm, as shown in Fig. 1, has recently been put forward in order to provide real-time/low-latency services and decrease the bandwidth requirement. The fog nodes (also known as *fog aggregation nodes*), extend the cloud to be closer to the edge by enabling computations to be carried out at the sensors/devices that produce and act on IoT data. In other words, it is possible for lightweight machine learning models to be embedded at the fog nodes, such that the data streaming in from various distributed sensors can be efficiently *aggregated* for high accuracy system-level predictions/decisions. Only data that is not time sensitive need be sent from the fog nodes to be processed in the cloud data center.

Note that in real-world IoT applications, certain types of data might be

abundantly collected and accumulated via distributed cheap sensors (denoted as red dots in Fig. 2), while distinct but associated high-fidelity data for building specific target predictive models of interest may demand scarce high precision/cost sensors (denoted as green dots in Fig. 2). In this case, transfer learning becomes an appealing proposition to boost the generalization performance on the target task. Nevertheless, traditional transfer learning models generally require all the source and target data to be uploaded from edge devices to the cloud, inevitably causing large amount of band-width cost. Clearly, the transmission cost is dominated by the large amount of source data. If a local transfer learning model utilizing only the source data generated nearby can be embedded in each fog (aggregation) node, then data transmission will be reduced dramatically. The framework is shown in Fig. 2, where only small amount of target data need be broadcast. Interestingly, when applying the proposed principled aggregation model, namely Tr-BCM, in this framework, each local TGP expert corresponds to a lightweight predictor that can be embedded in a fog (aggregation) node, thus catering to the case of online on-mote processing.

4. Preliminary

In this section, we present a brief overview of the TGP model proposed in [19].

4.1. Problem Specification

We first consider transfer regression problems with a single source task and a single target task. The dimensionality of the source and target inputs is set to d. Assume that a large source input set $\mathbf{X}_{\mathcal{S}} \in \mathbb{R}^{n_{\mathcal{S}} \times d}$ and the corresponding labels $\mathbf{y}_{\mathcal{S}} \in \mathbb{R}^{n_{\mathcal{S}}}$ are available for the source task $\mathcal{S}$, labeled as $\mathcal{D}_{\mathcal{S}} = \{\mathbf{X}_{\mathcal{S}}, \mathbf{y}_{\mathcal{S}}\}$. In contrast, the inputs $\mathbf{X}_{\mathcal{T}} \in \mathbb{R}^{n_{\mathcal{T}} \times d}$ and the corresponding labels $\mathbf{y}_{\mathcal{T}} \in \mathbb{R}^{n_{\mathcal{T}}}$ available for the target task $\mathcal{T}$ are relatively scarce (i.e., $n_{\mathcal{T}} \ll n_{\mathcal{S}}$). The overall target dataset is denoted as $\mathcal{D}_{\mathcal{T}} = \{\mathbf{X}_{\mathcal{T}}, \mathbf{y}_{\mathcal{T}}\}$. Generally, given the input $\mathbf{x}$, the source and target outputs are modeled as:

$$y_{\mathcal{S}} = f_{\mathcal{S}}(\mathbf{x}) + \epsilon_{\mathcal{S}},$$
$$y_{\mathcal{T}} = f_{\mathcal{T}}(\mathbf{x}) + \epsilon_{\mathcal{T}},$$

where the additive noise terms $\epsilon_{\mathcal{S}}$ and $\epsilon_{\mathcal{T}}$ are assumed to be independent, identically distributed (i.i.d.) Gaussian distributions with zero mean and

variance $\sigma_{\mathcal{S}}^2$ and $\sigma_{\mathcal{T}}^2$, respectively; $f_{\mathcal{S}}$ and $f_{\mathcal{T}}$ are the latent functions of the corresponding tasks. The objective is to transfer knowledge from the source task $\mathcal{S}$, so as to improve the generalization performance of a predictive model over target task $\mathcal{T}$.

4.2. Transfer Gaussian Process

GP is a popular stochastic, nonparametric approach for regression. It describes a distribution over functions, given as $f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, where $\mu(\mathbf{x})$ is the mean function (typically we set $\mu(\mathbf{x}) = 0$) and $k(\cdot, \cdot)$ is some valid covariance function. To be valid, any Gram matrix derived from kernel $k(\mathbf{x}, \mathbf{x}')$ is required to be PSD. Popular kernel functions include squared exponential (SE) and Matérn kernel. GP is a stochastic process wherein any finite subset of random variables follows a joint multivariate Gaussian distribution. Therefore, for a standard single-task GP, given the observations $\mathcal{D}_{\mathcal{T}} = \{\mathbf{X}_{\mathcal{T}}, \mathbf{y}_{\mathcal{T}}\}$ on the target task, the posterior distribution at a particular test point $\mathbf{x}_*$ is efficiently obtained [11].

In order to take advantage of abundant and perhaps correlated source data, Cao *et al.* [19] proposed the TGP model to achieve adaptive knowledge transfer while retaining the advantages of a standard GP model. The key distinguishing feature of the TGP model is the description of the following transfer covariance kernel:

$$\tilde{k}(\mathbf{x}, \mathbf{x}') = \begin{cases} \lambda k(\mathbf{x}, \mathbf{x}'), & \mathbf{x} \in \mathbf{X}_{\mathcal{S}} \ \& \ \mathbf{x}' \in \mathbf{X}_{\mathcal{T}} \\ & \text{or } \mathbf{x} \in \mathbf{X}_{\mathcal{T}} \ \& \ \mathbf{x}' \in \mathbf{X}_{\mathcal{S}} \\ k(\mathbf{x}, \mathbf{x}'), & \text{otherwise.} \end{cases} \tag{1}$$

Here, the additional parameter λ measures the source-target similarity. According to Theorem 1 in [19], $\tilde{k}(\cdot, \cdot)$ is a valid kernel for all $|\lambda| \leq 1$. If $|\lambda|$ is close to 1, it indicates that the source and target tasks are highly correlated.

As for the inference process of TGP, it is very similar to that of standard GP. In particular, the mean and the associated variance at an unknown target input $\mathbf{x}_*$ is given by:

$$\begin{aligned} \mu(\mathbf{x}_*) &= \tilde{\mathbf{k}}_{\mathbf{x}_*}(\widetilde{\mathbf{K}} + \Lambda)^{-1}\mathbf{y}, \\ \sigma^2(\mathbf{x}_*) &= \tilde{k}(\mathbf{x}_*, \mathbf{x}_*) - \tilde{\mathbf{k}}_{\mathbf{x}_*}^{\intercal}(\widetilde{\mathbf{K}} + \Lambda)^{-1}\tilde{\mathbf{k}}_{\mathbf{x}_*}, \end{aligned} \tag{2}$$

where $\tilde{\mathbf{k}}_{\mathbf{x}_*}$ is the kernel vector between $\mathbf{x}_*$ and $\mathbf{X} = \{\mathbf{X}_{\mathcal{S}}, \mathbf{X}_{\mathcal{T}}\}$ using the transfer kernel $\tilde{k}(\cdot, \cdot)$ in Eq.(1), $\Lambda = \begin{bmatrix} \sigma_{\mathcal{S}}^2 \mathbf{I}_{n_{\mathcal{S}}} & \mathbf{0} \\ \mathbf{0} & \sigma_{\mathcal{T}}^2 \mathbf{I}_{n_{\mathcal{T}}} \end{bmatrix}$, and $\widetilde{\mathbf{K}} = \begin{bmatrix} \widetilde{\mathbf{K}}_{\mathcal{SS}} & \widetilde{\mathbf{K}}_{\mathcal{ST}} \\ \widetilde{\mathbf{K}}_{\mathcal{TS}} & \widetilde{\mathbf{K}}_{\mathcal{TT}} \end{bmatrix}$

is the overall covariance matrix. In $\widetilde{\mathbf{K}}$, $\widetilde{\mathbf{K}}_{\mathcal{SS}}$ and $\widetilde{\mathbf{K}}_{\mathcal{TT}}$ are the kernel matrices of the data in the source task and target task, respectively; $\widetilde{\mathbf{K}}_{\mathcal{ST}} (= \widetilde{\mathbf{K}}_{\mathcal{TS}}^{\intercal})$ is the kernel matrix across source and target inputs.

During the training stage, the most commonly used approach for tuning the hyperparameters ($\boldsymbol{\theta}$) of the transfer covariance function is the conjugate gradient algorithm for optimizing the joint likelihood $p(\mathbf{y}_{\mathcal{T}}, \mathbf{y}_{\mathcal{S}} | \mathbf{X}_{\mathcal{T}}, \mathbf{X}_{\mathcal{S}}, \boldsymbol{\theta})$. Notably, training requires the inversion of covariance matrix $\widetilde{\mathbf{K}}$, which requires $\mathcal{O}((n_{\mathcal{S}} + n_{\mathcal{T}})^3)$ computations and $\mathcal{O}((n_{\mathcal{S}} + n_{\mathcal{T}})^2)$ memory. Given that $n_{\mathcal{S}} \gg n_{\mathcal{T}}$, the time and memory complexity can be written as $\mathcal{O}(n_{\mathcal{S}}^3)$ and $\mathcal{O}(n_{\mathcal{S}}^2)$, respectively. Due to the cubically scaling computational complexity and quadratically scaling storage requirements, the practical viability of TGP rapidly diminishes with increasing amount of source data accumulated over time - regardless of the potentially small size of the target dataset. Thus, the need to propose a scalable alternative over the existing TGP model is clear. From here on, we denote the afore-described model as full TGP to avoid possible confusions.

5. Model Aggregation for Fast Transfer Gaussian Processes

5.1. Factorized Training of Transfer Gaussian Processes

To be able to train a TGP model with large-scale source inputs using limited (or distributed) computational resources, a factorized training process is deemed as an efficient strategy. In this regard, a naive approach would be to partition all the source and target inputs $\mathbf{X}$ into M subsets, and then train every local TGP model with the corresponding local subset in parallel. Larger the choice of M, lesser is the computational burden imposed on each of the local TGPs. However, note that, given the scarcity of valuable target data, there tends to be fewer and fewer target inputs in each local subset with increasing M. Since training a good local TGP expert will require the availability of a reasonable amount of target inputs, it immediately follows that a naive extension of single-task model aggregation may not suffice in the transfer learning case.

Thus, considering that the computational complexity of TGP training is primarily dominated by the large amount of source inputs (as $n_{\mathcal{S}} \gg n_{\mathcal{T}}$), we propose to partition only the source data into M spatially disjoint subsets, i.e., $\mathcal{D}_{\mathcal{S}} = \{\mathcal{D}_{\mathcal{S}_1}, \cdots, \mathcal{D}_{\mathcal{S}_M}\}$, with $\mathcal{D}_{\mathcal{S}_i} = \{\mathbf{X}_{\mathcal{S}_i}, \mathbf{y}_{\mathcal{S}_i}\}$ for $i = 1, \cdots, M$, to effectively decrease the computational burden on each local TGP model. In addition, each local expert is provided with the *entire* target dataset -

without partitioning. That is, for the ith expert $\mathcal{M}_i$, the corresponding training inputs are $\mathcal{D}_{\mathcal{S}_i}$ and $\mathcal{D}_{\mathcal{T}}$. With this, the M 'local'[1] experts $\{\mathcal{M}_i\}_{i=1}^{M}$ can be trained in parallel.

During the hyperparameter learning stage of expert $\mathcal{M}_i$, the log marginal likelihood computed with respect to $\mathcal{D}_{\mathcal{S}_i}$ and $\mathcal{D}_{\mathcal{T}}$, i.e., $\log p(\mathbf{y}_{\mathcal{T}}, \mathbf{y}_{\mathcal{S}_i}|\mathbf{X}_{\mathcal{T}}, \mathbf{X}_{\mathcal{S}_i}, \boldsymbol{\theta})$, is optimized. Specifically, the log marginal likelihood of the expert $\mathcal{M}_i$ is given by

$$\begin{aligned}
&\log p\big(\mathbf{y}_{\mathcal{S}_i}, \mathbf{y}_{\mathcal{T}}|\mathbf{X}_{\mathcal{T}}, \mathbf{X}_{\mathcal{S}_i}, \boldsymbol{\theta}\big) \\
&\quad = -\frac{1}{2}[\mathbf{y}_{\mathcal{S}_i}^{\intercal} \quad \mathbf{y}_{\mathcal{T}}^{\intercal}](\widetilde{\mathbf{K}}_i + \Lambda_i)^{-1}\begin{bmatrix}\mathbf{y}_{\mathcal{S}_i} \\ \mathbf{y}_{\mathcal{T}}\end{bmatrix} \\
&\qquad -\frac{1}{2}\log\big(|\widetilde{\mathbf{K}}_i + \Lambda_i|\big) + const, \qquad (3)
\end{aligned}$$

where $\widetilde{\mathbf{K}}_i = \begin{bmatrix}\widetilde{\mathbf{K}}_{\mathcal{S}_i\mathcal{S}_i} & \widetilde{\mathbf{K}}_{\mathcal{S}_i\mathcal{T}} \\ \widetilde{\mathbf{K}}_{\mathcal{T}\mathcal{S}_i} & \widetilde{\mathbf{K}}_{\mathcal{T}\mathcal{T}}\end{bmatrix}$ and $\Lambda_i = \begin{bmatrix}\sigma_{\mathcal{S}}^2\mathbf{I}_{n_{\mathcal{S}_i}} & \mathbf{0} \\ \mathbf{0} & \sigma_{\mathcal{T}}^2\mathbf{I}_{n_{\mathcal{T}}}\end{bmatrix}$. This means that while training the ith expert, the observations on all the other source subsets are considered to be marginalized. A further provision made in the present paper is that the learned hyperparameters of the transfer covariance function are shared across all local TGP experts as a way to prevent individual model overfitting, hence no additional hyperparameters are needed compared to the full TGP model.

5.2. Principled Tr-BCM for Aggregative Model Prediction

Given an unknown target inputs $\mathbf{x}_{\mathcal{T}}^q$, we consider M (Gaussian) predictive distributions from $\{\mathcal{M}_i\}_{i=1}^{M}$ local TGP experts to be combined for the final output. The corresponding unknown response variable is defined as $f_{\mathcal{T}}^q$. Let $p(f_{\mathcal{T}}^q|\mathbf{x}_{\mathcal{T}}^q, \mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}_i})$[2] be the posterior predictive probability density at the query point for expert $\mathcal{M}_i$, and the corresponding predictive mean and variance are denoted as $\mu_i(\mathbf{x}_{\mathcal{T}}^q)$ and $\sigma_i^2(\mathbf{x}_{\mathcal{T}}^q)$, respectively. Therefore, in what follows, we propose an efficient strategy to combine these predictive distributions in a theoretically principled manner.

The idea of the Bayesian Committee Machine (BCM) was first introduced in [25] in the context of single-task learning. The BCM is formally equivalent to an inducing-point model in which the test points are the inducing inputs

[1]Here, the term 'local' is defined in the context of the source data.

[2]Hereafter we shall omit the dependence on $\mathbf{x}_{\mathcal{T}}^q$ in our notation for simplicity.

[49]. It provides a principled strategy to combining local estimators that may have been trained in parallel. Inspired by the mathematical derivations of BCM, we here propose transfer BCM (Tr-BCM), as a principled approach to combining predictions from local TGP experts.

Let $\mathscr{D}_{\mathcal{S}_i} = \{\mathcal{D}_{\mathcal{S}_1}, \cdots, \mathcal{D}_{\mathcal{S}_i}\}$ represent the set of all source datasets with indices smaller or equal to i, with $i = 1, \cdots, M$. For the first i source subsets $\mathscr{D}_{\mathcal{S}_i}$, we have

$$\begin{aligned} p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}_i}, \mathscr{D}_{\mathcal{S}_{i-1}}) \propto\ & p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}) p(\mathscr{D}_{\mathcal{S}_{i-1}}|f_{\mathcal{T}}^q, \mathcal{D}_{\mathcal{T}}) \\ & p(\mathcal{D}_{\mathcal{S}_i}|f_{\mathcal{T}}^q, \mathcal{D}_{\mathcal{T}}, \mathscr{D}_{\mathcal{S}_{i-1}}) \end{aligned} \tag{4}$$

Note that $p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}})$ is the posterior predictive distribution using only target training inputs, and we label the corresponding expert as $\mathcal{M}_{\mathcal{T}}$[3]. To simplify the calculation, we make the following conditional independence assumptions,

$$p(\mathcal{D}_{\mathcal{S}_i}|f_{\mathcal{T}}^q, \mathcal{D}_{\mathcal{T}}, \mathscr{D}_{\mathcal{S}_{i-1}}) \approx p(\mathcal{D}_{\mathcal{S}_i}|f_{\mathcal{T}}^q, \mathcal{D}_{\mathcal{T}}) \tag{5}$$

Iteratively applying Bayes' rule, we obtain

$$\begin{aligned} p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}}) &\approx const \times \frac{p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}_M}) p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}, \mathscr{D}_{\mathcal{S}_{M-1}})}{p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}})} \\ &\approx const \times \frac{\prod_{i=1}^M p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}_i})}{p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}})^{M-1}}. \end{aligned} \tag{6}$$

As a consequence, the predictive distribution is still a Gaussian, with mean and variance listed as follows:

$$\begin{aligned} \mu_{\text{Tr-BCM}}(\mathbf{x}_{\mathcal{T}}^q) &= \sigma_{\text{Tr-BCM}}^2(\mathbf{x}_{\mathcal{T}}^q)\Big(\sum_{i=1}^M \sigma_i^{-2}(\mathbf{x}_{\mathcal{T}}^q)\mu_i(\mathbf{x}_{\mathcal{T}}^q) \\ &\qquad + (1-M)\sigma_{\mathcal{T}}^{-2}(\mathbf{x}_{\mathcal{T}}^q)\mu_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^q)\Big), \\ \sigma_{\text{Tr-BCM}}^2(\mathbf{x}_*) &= 1/\Big(\sum_{i=1}^M \sigma_i^{-2}(\mathbf{x}_{\mathcal{T}}^q) + (1-M)\sigma_{\mathcal{T}}^{-2}(\mathbf{x}_{\mathcal{T}}^q)\Big), \end{aligned} \tag{7}$$

[3]No extra training procedure is necessary for model $\mathcal{M}_{\mathcal{T}}$, since a common target model with shared hyperparameters can be obtained after marginalizing out the corresponding source subsets for every local TGP model.

where $\mu_{\mathcal{T}}$ and $\sigma^2_{\mathcal{T}}$ are the predicted mean and variance of the expert $\mathcal{M}_{\mathcal{T}}$. From the predictive distribution, it is easy to observe that the overall weight assigned to the expert $\mathcal{M}_i$ in the predictive mean is inversely proportional to its variance. This implies that those experts with more confident prediction are automatically assigned higher responsibility in an input dependent manner.

Observe that unlike single-task BCM, where the "correction" term for the predictive variance is the prior variance $k_{**} = k(\mathbf{x}^q_{\mathcal{T}}, \mathbf{x}^q_{\mathcal{T}})$ [27], the posterior $\sigma^2_{\mathcal{T}}$ is used to rectify the predictive variance of Tr-BCM. This is caused by the fact that in Tr-BCM, the whole target data is fully utilized across every local expert, guaranteeing quality predictions from each local expert.

When applying the proposed Tr-BCM in the setting of Fog computing, as illustrated in Fig. 2, each local TGP expert can be embedded in a fog (aggregation) node. It is observed that each local TGP expert only utilizes the target data and a single source subset, which is generated by the nearby sensors. Therefore, the large amount of source data is processed in a distributed manner in relatively lightweight local experts directly at the fog nodes, avoiding the cost of transmission to the cloud. Only the small amount of target data need be broadcast among the fog nodes. When making predictions, the predictive means and variances of each local expert are shared or transmitted to the cloud. Thereafter, the proposed Tr-BCM is invoked to combine the predictions in a theoretically principled manner.

5.3. Alternative Heuristic Model Aggregations

Apart from the principled Tr-BCM, it is possible to construct alternative model aggregation schemes based on (related) heuristically defined procedures that have recently been developed for large-scale single-task GPs. A prominent example among them is the *product of experts* (PoE) [26]. In the PoE, all the predictive distributions at $\mathbf{x}^q_{\mathcal{T}}$ from a set of local estimators are directly multiplied, and the product is proportional to a Gaussian distribution if every local predictive distribution is a Gaussian. Similarly, in the case of transfer learning, the final output distribution can be directly set as proportional to the product of all the predictive distributions from the M local TGP experts, with the resultant mean and variance listed as follows:

$$\mu_{\text{PoE}}(\mathbf{x}_{\mathcal{T}}^q) = \sigma_{\text{PoE}}^2 \sum_{i=1}^{M} \beta_i \sigma_i^{-2}(\mathbf{x}_{\mathcal{T}}^q)\mu_i(\mathbf{x}_*),$$
$$\sigma_{\text{PoE}}^2(\mathbf{x}_{\mathcal{T}}^q) = 1/\Big(\sum_{i=1}^{M} \beta_i \sigma_i^{-2}(\mathbf{x}_{\mathcal{T}}^q))\Big), \tag{8}$$

where the heuristically incorporated tunable parameter β_i is set to 1 for $i = 1, \cdots, M$.

From Eq.(8), observe the familiar property that experts which are uncertain about their predictions are automatically weighted less than those which are more confident about their predictions. However, with an increasing number of TGP experts, Eq.(8) implies that the predictive variance $\sigma_{\text{PoE}}^2(\mathbf{x}_{\mathcal{T}}^q)$ monotonically decreases, leading to unreasonably overconfident predictions. Therefore, the PoE model is inconsistent in the sense that it does not fall back to the prior outside the regime of the training dataset [27]. To overcome the evident issue of the PoE aggregation approach, it has been proposed in the literature to simply set $\sum_{i=1}^{M} \beta_i = 1$. The corresponding model is labeled as *generalized PoE* (gPoE). Accordingly, in this paper, we set $\beta_i = 1/M$, so that the predictive means of PoE and gPoE are identical, and only the predictive variances are adjusted.

5.4. Empirical Analysis of Various Aggregation Models

We analyze and compare the Tr-BCM, PoE, and gPoE methods using a 1-D toy example. In this toy example, a single source and target inputs are sampled according to a full TGP model with source-target similarity $\lambda = 0.5$ and squared exponential kernel with pre-specified hyperparameters. There are a large number (1,000) of source inputs sampled uniformly at random from the range $[-1, 1]$, and 5 target inputs sampled from the range of $[0, 1]$. We partition the source training inputs into two disjoint subsets. Thus, the predictive distributions of the two local TGPs, $\mathcal{M}_1$ and $\mathcal{M}_2$ are displayed in 3(a) and 3(b). The resultant predictions from the three different model aggregation schemes are presented in 3(c-e), with each compared against the predictions made by the full TGP model. Our goal is to test how closely the proposed lightweight aggregation schemes can replicate the full TGP model.

Clearly, the predictive performance of Tr-BCM as shown in Fig. 3(c) is the best approximation of the full TGP model among all the aggregation

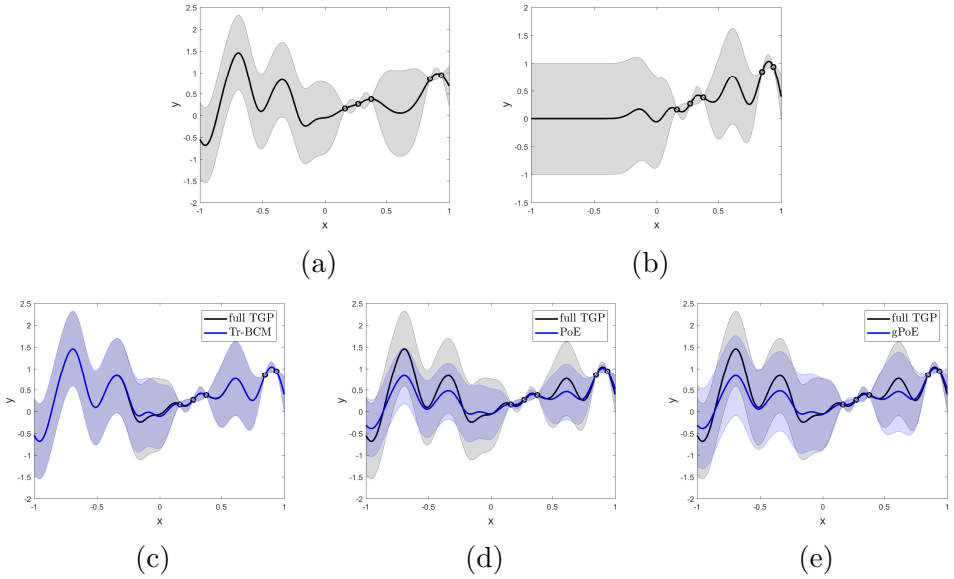

Figure 3: Toy example of aggregation of two local TGP experts. In (a) and (b), for each model, we present the predictive mean (black curve) while the gray shaded region denotes the standard deviation. The "○" symbols represent the 5 target training samples. **c-e** aggregated predictions (in blue) from Tr-BCM, PoE, and gPoE compared against the full TGP model prediction (in gray black).

models. For the PoE in Fig. 3(d), the problematic overconfident prediction (with unreasonably low variance) is verified. What is more, the predictive mean of PoE does not align well with the full TGP either. In contrast to PoE, gPoE seems to make more consistent predictions of the predictive variance, as displayed in Fig. 3(e).

5.5. Theoretical Analysis of Various Aggregation Models

We further analyze and compare the theoretical behavior of the proposed Tr-BCM against PoE and gPoE. To simplify our analysis, we assume that stationary and monotonic kernel is applied. All source subsets are spatially disjoint, such that $k(\mathbf{x}, \mathbf{x}') \approx 0$, for $\mathbf{x} \in \mathcal{D}_{\mathcal{S}_i}$ and $\mathbf{x}' \in \mathcal{D}_{\mathcal{S}_j}$, when $i \neq j$. In the following, we will analyze the predictive behavior on an unknown query points $\mathbf{x}_{\mathcal{T}}^q$ under two circumstances. One case is that $\mathbf{x}_{\mathcal{T}}^q$ is distant from all the source subsets. The other case is that $\mathbf{x}_{\mathcal{T}}^q$ falls within the regime of a specific source subset. The proposed Tr-BCM model is proved to output consistent predictive distributions with the ones made by the full TGP model.

In the first case, $\mathbf{x}_{\mathcal{T}}^q$ is distant from all source subsets, meaning that $k(\mathbf{x}_{\mathcal{T}}^q, \mathbf{x}) \approx 0$, for all $\mathbf{x} \in \mathcal{D}_{\mathcal{S}_i}$, $i = 1, \cdots, M$. Thus, the predictive distributions of all the local TGP experts $p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}_i})$ and the full TGP model $p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}})$ fall back to the prediction of $\mathcal{M}_{\mathcal{T}}$, i.e.,

$$\begin{aligned} p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}}) &\approx p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}_i}) \approx p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}) \\ &= \mathcal{N}(\mu_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^q), \sigma_{\mathcal{T}}^2(\mathbf{x}_{\mathcal{T}}^q)),\ i = 1, \cdots, M. \end{aligned} \tag{9}$$

According to Eq.(7) and Eq.(8), Tr-BCM and gPoE can produce the same predictive distributions as the one made by the full TGP model, while PoE makes unreasonably overconfident predictions as $\lim_{M\to\infty} \sigma_{\text{PoE}}^2(\mathbf{x}_{\mathcal{T}}^q) = 0$.

In the second circumstance, $\mathbf{x}_{\mathcal{T}}^q$ falls within the regime of a specific source subset - say the ith expert. Therefore, we have:

$$\begin{aligned} &p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}}) \approx p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}_i}) = \mathcal{N}(\mu_i(\mathbf{x}_{\mathcal{T}}^q), \sigma_i^2(\mathbf{x}_{\mathcal{T}}^q)), \\ &p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}_j}) \approx p(f_{\mathcal{T}}^q|\mathcal{D}_{\mathcal{T}}) = \mathcal{N}(\mu_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^q), \sigma_{\mathcal{T}}^2(\mathbf{x}_{\mathcal{T}}^q)), \quad j \neq i. \end{aligned} \tag{10}$$

According to Eq.(7), the aggregated predictive distributions of Tr-BCM are equivalent to the ones made by the full TGP model. The PoE aggregation scheme continues to make characteristic overconfident predictions. For the gPoE, from Eq.(8) - with $\beta_i = 1/M$ - we find that with increasing number of experts, the predictive variance can be written as $\lim_{M\to\infty} \sigma_{\text{gPoE}}^2(\mathbf{x}_{\mathcal{T}}^q) = \sigma_{\mathcal{T}}^2(\mathbf{x}_{\mathcal{T}}^q)$. In addition, according to Proposition 1 in [36], given the availability of related source data ($|\lambda| > 0$), we have $\sigma_i^2(\mathbf{x}_{\mathcal{T}}^q) < \sigma_T^2(\mathbf{x}_{\mathcal{T}}^q)$. These facts imply that since the aggregated predictive distribution of gPoE falls back to the prediction of single-task expert $\mathcal{M}_{\mathcal{T}}$, the predictive behavior of gPoE theoretically tends to be over conservative compared to full TGP.

In summary, the theoretical behavior of the proposed model, namely Tr-BCM, is consistent with the full TGP model. In contrast, the heuristically defined PoE and gPoE aggregation schemes tend to make overconfident and over-conservative predictions, respectively.

5.6. Computational Complexity and Memory Consumption

As elaborated earlier, the $\mathcal{O}(n_{\mathcal{S}}^3)$ computational complexity and $\mathcal{O}(n_{\mathcal{S}}^2)$ storage requirement are bottlenecks to scale a full TGP model to tackle problems with large-scale sources. To overcome this issue, we have put forward a theoretically principled Tr-BCM model that partitions the source data into disjoint subsets so as to build lightweight local TGP experts. Here, we analyze the complexity of the Tr-BCM approach. We refer to Eq.(3) which points

to the inversion of matrix $(\widetilde{\mathbf{K}}_i + \Lambda_i)$, for $i = 1, \ldots, M$. Expert $\mathcal{M}_i$ will require $\mathcal{O}((n_{\mathcal{S}_i} + n_{\mathcal{T}})^3)$ computations and $\mathcal{O}((n_{\mathcal{S}_i} + n_{\mathcal{T}})^2)$ memory space. Therefore, the overall factorized training process requires $\mathcal{O}(M \times (n_{\mathcal{S}}/M + n_{\mathcal{T}})^3)$ computations and $\mathcal{O}(M \times (n_{\mathcal{S}}/M + n_{\mathcal{T}})^2)$ memory, assuming uniform source data partitioning. In the present paper, we set $M \approx n_{\mathcal{S}}/(2n_{\mathcal{T}})$ in the experimental study to enable sufficient source-target knowledge transfer. Accordingly the overall training complexity decreases to $\mathcal{O}(\frac{27}{2} n_{\mathcal{S}} n_{\mathcal{T}}^2)$, and the memory cost reduces to $\mathcal{O}(\frac{9}{2} n_{\mathcal{S}} n_{\mathcal{T}})$. In other words, both quantities scale linearly with the number of source observations; thereby making Tr-BCM a viable option for lightweight online on-mote processing on edge devices.

6. Enhanced Expressiveness with Tr-BCM: Local Inter-Task Similarity Capture

Recent progress towards adaptive multi-task/transfer GP has shown that the expressiveness of a model can be enhanced by exploiting *spatially adaptive* inter-task relationship [30, 31]. However, most existing approaches for adaptive multi-task/transfer learning have been focused on *fixed* correlations among output variables. In other words, it has been assumed that the source-target relationship can be captured by a single scalar parameter, and is uniform everywhere in the input space. The same is seen to be true for the full TGP model, where a single parameter (λ) is used to capture source-target similarity. However, this assumption is often found to be too strict for real-world applications. Therefore, in this section, we explore the possibility of equipping traditional transfer learning with the ability to learn non-uniform inter-task relationship through a simple adjustment to Tr-BCM.

Taking advantage of the partition of source dataset into M disjoint subsets, a straightforward approach to capture localized inter-task similarity would be to apply the following localized transfer covariance function:

$$\tilde{k}(\mathbf{x}, \mathbf{x}') = \begin{cases} \lambda_i k(\mathbf{x}, \mathbf{x}'), & \mathbf{x} \in \mathbf{X}_{\mathcal{S}_i} \;\&\; \mathbf{x}' \in \mathbf{X}_{\mathcal{T}} \\ & \text{or } \mathbf{x} \in \mathbf{X}_{\mathcal{T}} \;\&\; \mathbf{x}' \in \mathbf{X}_{\mathcal{S}_i} \\ k(\mathbf{x}, \mathbf{x}'), & \text{otherwise,} \end{cases} \tag{11}$$

where λ_i indicates the localized inter-task similarity between the ith source subset $\mathcal{S}_i$ and the target task $\mathcal{T}$. Using this transfer covariance function, localized inter-task relationship is learned between the different source subsets and the target data.

Let $K^f \in \mathbb{R}^{(M+1)\times(M+1)}$ represent a matrix capturing the inter-task (between source and target) and intra-task (between different source subsets) similarities. Naturally, the similarity across data subsets belonging to the same source task can be assumed to be 1. Accordingly, K^f is expressed in the following form:

$$K^f = \begin{pmatrix} 1 & 1 & \cdots & 1 & \lambda_1 \\ 1 & 1 & \cdots & 1 & \lambda_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & \lambda_M \\ \lambda_1 & \lambda_2 & \cdots & \lambda_M & 1 \end{pmatrix}. \tag{12}$$

In order to guarantee that the localized transfer covariance function in Eq.(11) is always PSD given a valid kernel $k(\cdot,\cdot)$, it suffices for us to show that K^f is a PSD matrix [18]. The following theorem gives the necessary and sufficient condition for a PSD K^f.

Theorem 1. *The matrix K^f is PSD if and only if $\lambda_1 = \lambda_2 = \cdots = \lambda_M$, and $|\lambda_i| \leq 1$, for all $i = 1, 2, \cdots, M$.*

Proof. Necessary condition: A principal minor of any matrix A is defined as the determinant of a principal submatrix of matrix A. Let A be an symmetric matrix. Then A is PSD if and only if every principal minor of A is nonnegative [50]. Therefore, given K^f is PSD, for a 2×2 principal submatrix $K_i^f = \begin{pmatrix} 1 & \lambda_i \\ \lambda_i & 1 \end{pmatrix}$, we have $|K_i^f| = 1 - \lambda_i^2 \geq 0$. Therefore, $|\lambda_i| \leq 1$, for $i = 1, \cdots, M$.

Further, for a 3×3 principal submatrix $K_{ij}^f = \begin{pmatrix} 1 & 1 & \lambda_i \\ 1 & 1 & \lambda_j \\ \lambda_i & \lambda_j & 1 \end{pmatrix}$, we have $|K_{ij}^f| = -(\lambda_i - \lambda_j)^2 \geq 0$. Thus, we have $\lambda_i = \lambda_j$.

Sufficiency condition: Let $\lambda_1 = \lambda_2 = \cdots = \lambda_M = \lambda$ and $|\lambda| \leq 1$. According to Theorem 1 in Cao et al. [19], it follows that K^f is PSD since a matrix of all ones is PSD. □

According to the above theorem, all the local source-target similarities take the same value in order to guarantee the validity of the transfer covariance function of Eq.(11). However, such a condition hampers the original intention of partitioning the source data into local subsets to learn localized

inter-task relationship between each source subset and the target task. Thus, training a full TGP with the localized transfer kernel is not guaranteed to be feasible.

In contrast, Tr-BCM can easily avoid this issue by neglecting the correlations between different source subsets, as only the correlations between the individual source subsets and the target task are considered separately under a conditional independence assumption. Taking this cue, we slightly relax the provision for sharing a single set of hyperparameters across all local TGP experts in Tr-BCM, and allow localized λ_i's for each local model to be learned. All other hyperparameters of the covariance function continue to be shared. After the factorized training stage, if we marginalize out the source subsets, a common target expert $\mathcal{M}_{\mathcal{T}}$ will still be obtained. Therefore, during prediction, Eq.(7) for Tr-BCM can be directly applied.

It is observed that if the ith local expert learns a high source-target correlation, i.e., $|\lambda_i| \rightarrow 1$, then predictions within its local region will be highly supported by the local subset of the source data. On the contrary, if λ_i is learned to be close to 0, then there is little knowledge transferred from the corresponding source subset to the target task. As there is no restriction placed on the λ_i's to be uniform across the M subsets, the non-uniformity of the source-target similarity distribution is practically addressed.

To provide insights on the behavior of the *Tr-BCM model with localized inter-task similarity capture* (labeled as Tr-BCM-ls), we consider a toy example. The generation of the synthetic dataset is carried out as follows. 100 data points are randomly sampled from each of the two 1-D functions $f_{\mathcal{S}} = \sin(|x|)$ and $f_{\mathcal{T}} = \sin(x)$, $-5 \leq x \leq 5$, both corrupted by a zero-mean Gaussian noise with variance equal to 0.1. The first function is taken as source task, and the second function is taken as target task. 5% of the target data points are used for training, and the rest are used for testing. After training the conventional full TGP model, we obtain that the source-task similarity is $\lambda \approx 0$, implying that the source and target tasks are nearly uncorrelated globally. As a result, the performance of TGP is somewhat similar to that of single-task GP, showing that there is nearly no knowledge transfer from source task to target task.

Nevertheless, it is apparent from the function forms of $f_{\mathcal{S}}$ and $f_{\mathcal{T}}$ that there naturally exist two regions in the input space where the source and target tasks are indeed correlated. In particular, the task pairs are strongly *positively* correlated when $x >= 0$, and strongly *negatively* correlated when $x < 0$. Accordingly, we partition the source data into 2 subsets, i.e., $\mathcal{D}_{\mathcal{S}_1} =$

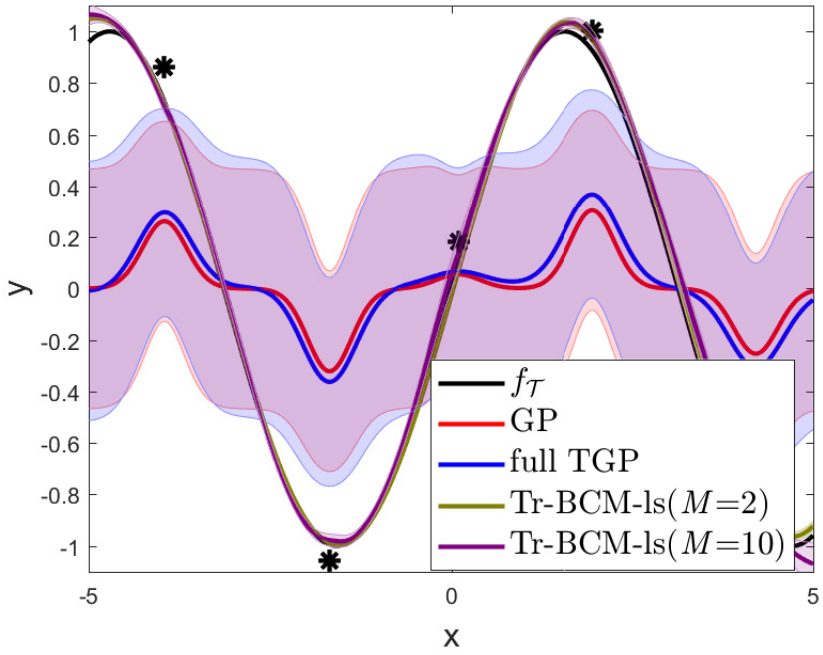

Figure 4: Predictive distribution of GP, full TGP and the proposed Tr-BCM. Shaded area denotes the predicted standard derivation of the corresponding probabilistic output. The starred points are the training data of the target task.

$\{x_i >= 0 : x_i \in \mathcal{D}_{\mathcal{S}}\}$ and $\mathcal{D}_{\mathcal{S}_2} = \{x_i < 0 : x_i \in \mathcal{D}_{\mathcal{S}}\}$. By applying the proposed factorized training with localized inter-task similarity capture, we find that two different source-target similarities (-0.99 and 1.00) are indeed learned, closely matching the true underlying distribution of inter-task similarity. The predicted distribution of Tr-BCM-ls is shown in Fig. 4. Using only 5% of the target data, the proposed method can almost exactly recover the target function $f_{\mathcal{T}}$ by adaptively taking advantage of the knowledge concealed in the source task, highlighting the efficacy of the proposed method.

In order to quantify the averaged generalization performance on the test set over 10 trial runs, we present root mean square error (RMSE[4]) results in Table 1. The results in Table 1 also contain the case of Tr-BCM-ls with

[4]RMSE is computed as $\sqrt{\frac{\sum_{i=1}^{n_{\text{test}}}(y_i-\mu(\mathbf{x}_i))^2}{n_{\text{test}}}}$ on test samples, where y_i is the label of $\mathbf{x}_i$, and $\mu(\mathbf{x}_i)$ is the predicted mean of $\mathbf{x}_i$ on target task $\mathcal{T}$.

Table 1: Averaged RMSE on toy example

Methods	RMSE
GP	0.42623 ± 0.0084
full TGP	0.45456 ± 0.0004
Tr-BCM-ls ($M = 2$)	**0.1050** ± 0.0040
Tr-BCM-ls ($M = 10$)	0.12681 ± 0.0176

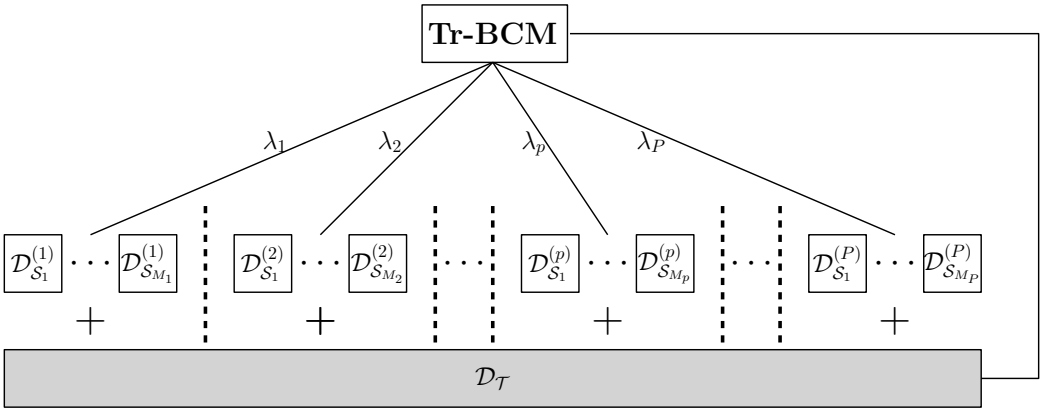

Figure 5: Hierarchical structure of multi-source Tr-BCM.

$M = n_{\mathcal{S}}/2n_{\mathcal{S}} = 10$ (k-means is used to partition the source dataset). Tr-BCM-ls with $M = 10$ outperforms full TGP and GP with a large margin. However, Tr-BCM-ls with $M = 2$ slightly outperforms Tr-BCM-ls with $M = 10$, as in the former case prior knowledge about the underlying distribution of source-target correlation was utilized while partitioning the source datasets.

7. Extensions of Tr-BCM to Multi-Source Transfer Learning

Given the proposed relaxation of Tr-BCM with localized inter-task relationship capture, a similar scheme can be immediately used to deal with multi-source transfer learning problems as well.

With research efforts largely confined to the single-source setting [19, 1], an increasing amount of studies are contributing to a realistic applicability of transfer learning by addressing the multi-source scenario - where different sources have differing degree of inter-task relationship with the target [21]. By ignoring the interactions among the different source tasks, the relaxed Tr-BCM formulation can be directly applied to tackle multi-source transfer

Table 2: Results on the real-world datasets. The averaged RMSEs of different approaches for Wine and error distance (in meter) for UJIIndoorLoc are reported. Superior performance are highlighted using bold characters.

Methods	Wine	UJIIndoorLoc
Tr-BCM	**0.7074**±0.0022	**6.8197**±0.0711
(g)PoE	0.7562±0.0008	7.4277±0.0111
full TGP	0.7739±0.0207	6.9279±0.0164
GP	0.7619±0.0019	7.6545±0.0001
TSGP	0.7675±0.0056	7.5631±0.0001
TrAdaBoost.R2	0.8053±0.0111	39.448±0.0003

learning problems. Accordingly, in the following, we propose a hierarchical structure of Tr-BCM to tackle these kinds of problems.

Say there are P source tasks $\mathcal{S}^{(1)}, \cdots, \mathcal{S}^{(P)}$ and one target task $\mathcal{T}$. We assume all the tasks are defined in a common input space with dimensionality d. For the pth source task, the corresponding training data is labeled as $\mathcal{D}_{\mathcal{S}}^{(p)} = \{\mathbf{X}_{\mathcal{S}}^{(p)}, \mathbf{y}_{\mathcal{S}}^{(p)}\}$, where $\mathbf{X}_{\mathcal{S}}^{(p)} \in \mathbb{R}^{n_{\mathcal{S}}^{(p)} \times d}$ and $\mathbf{y}_{\mathcal{S}}^{(p)} \in \mathbb{R}^{n_{\mathcal{S}}^{(p)}}$. To accelerate the computational process, $\mathcal{D}_{\mathcal{S}}^{(p)}$ is partitioned into $M_p = n_{\mathcal{S}}^{(p)}/2n_{\mathcal{T}}$ local blocks. Hence, $M = \sum_{p=1}^{P} M_p$ TGP experts undergo factorized training in parallel. Notice that each source task possesses a unique noise level and source-target similarity, while all the other hyperparameters are shared across all the experts. Fig. 5 shows the hierarchical structure of the proposed model. The aggregated predictive distributions are directly calculated using Eq.(7).

8. Experimental Study

8.1. Medium-scale Datasets

In the following, we conduct experiments on two UCI datasets with a single source task with medium-sized source training inputs. In addition to the proposed Tr-BCM and the direct extension of (g)PoE, we present results obtained from standard GP, the full TGP model, implementations of Transfer Stacking GP (TSGP) [42], and TrAdaBoost.R2 [51] for regression transfer. For all the aggregation models, the number of experts is set as $M = n_{\mathcal{S}}/(2n_{\mathcal{T}})$, and we use k-means to partition the source data. In the following, only predictive mean is used to measure the generalization performance, therefore, PoE and gPoE serve as referred as one model.

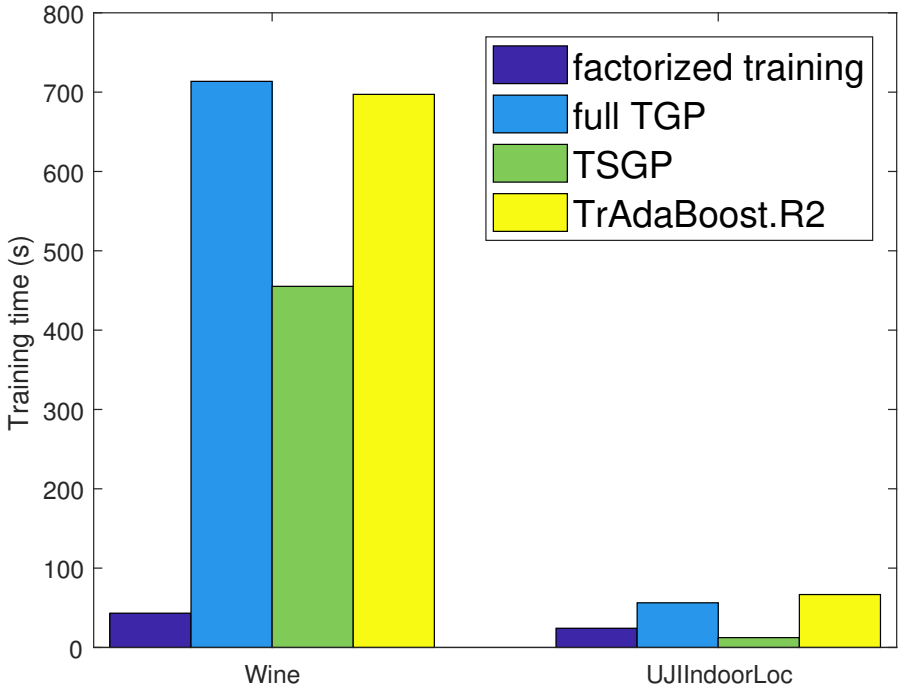

Figure 6: The averaged training time for each method.

8.1.1. Wine Quality Dataset

The wine dataset [52] is related to red and white wine samples, and the goal is to model wine quality based on physicochemical tests including PH values, etc (in total 11 features). The labels are given by experts with grades between 0 (very bad) and 10 (very good). There are in total 4898 records, among which 1599 are for the red wine, and 4898 are for the white wine. In the experimental study, the quality prediction for the white wine is used as the source task, and the quality prediction for red wine is taken as the target task. 5% of the available target data is used for training, and the remaining is used for evaluation.

8.1.2. WiFi-based Indoor Localization

The WiFi-based indoor localization system aims to detect the location of a client device given the signals received from various access points. Given the ever-expanding scale of WiFi deployments in metropolitan areas, WiFi-based

localization gains its importance and popularity due to the many AI and ubiquitous computing applications. However, most localization techniques require a training set of signal strength readings labeled against a ground truth location map. Training data of the target task is precious due to the heavy reliance on the ground truth calibration. Therefore, transfer learning becomes more appealing as fruitful knowledge from some source data can be utilized to decrease the workload of calibrating the target data. In the experimental study, we randomly choose two floors in one building as source and target task. Therefore, there are 1137 source inputs, 78 target training inputs and 1486 test samples.

All the experiments are conducted over 10 repetitions. The results are presented in Table 2. Note that the proposed Tr-BCM performs the best over the other compared methods. What is more, single-task GP outperforms full TGP. The reason is probably that optimizing the joint likelihood over source and target inputs may bias the TGP model towards the source task since $n_{\mathcal{T}} \ll n_{\mathcal{S}}$. On the other hand, in the proposed factorized training scheme, the training data for each expert is more balanced since within each local expert, the number of source inputs is limited to be twice that of target inputs. TrAdaBoost.R2 is always found to perform the worst over all the methods. This is consistent with the experimental results reported in [21].

Further, we record the training time of all the transfer learning methods, which are reported in Fig. 6. Note that there are more source inputs for Wine data than in UJIIndoorLoc. As a result, the proposed factorized training for transfer learning shows its advantages with larger source inputs. Comparing the number of source inputs for the two datasets, the runtime for the proposed factorized training does not increase drastically with the increased number of source inputs (scales linearly), while training time for other methods increases drastically (scales cubically).

Table 3: Results on the large-scale datasets. The RMSEs of different approaches are reported for SARCOS. Superior performance are highlighted using bold characters.

Methods	RMSE
Tr-BCM	**5.8715**±0.9077
(g)PoE	6.1557±1.1485
GP	14.8993±5.7632

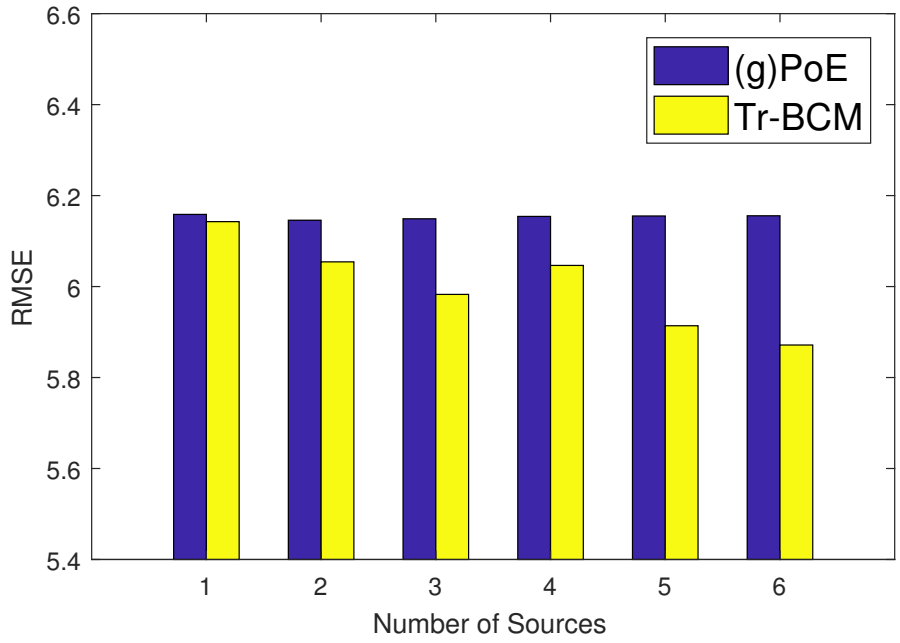

Figure 7: The averaged RMSE over 10 runs for (g)PoE, and Tr-BCM with increasing number of source tasks.

8.2. Large-scale Dataset

The SARCOS dataset [19] relates to an inverse dynamics problem for a seven degrees-of-freedom anthropomorphic robot arm. The task is to map from a 21-dimensional input space (7 joint position, 7 joint velocities, 7 joint accelerations) to the corresponding 7 joint torques. Therefore, the input has 21 dimensions and there are 7 tasks for each input. The original problem is of multi-output regression. In this experiment, we use the first joint torque as the target task, and the remaining six joint torques as six different source tasks. 5% of the available target data is used for training, and the remaining is used for evaluation. For the source tasks, we randomly choose 30,000 points in total. With huge amount of source inputs, most traditional GP-based methods become impractical. As we compare the aggregation models Tr-BCM and gPoE to a standard single-task GP, a huge performance enhancement is observed as shown in Table 3. Notably, Tr-BCM outperforms (g)PoE.

We have also analyzed the effect of increasing number of sources. Using

factorized training, we first jointly train the six source tasks and target task. During prediction, we consider different cases in which the number of sources is gradually increased. The predictive performance of Tr-BCM and (g)PoE is shown in Fig. 7, averaged over 10 repetitions. With increasing number of source tasks, Tr-BCM is found to significantly improve its performance with the availability of more source inputs, while the performance enhancement for (g)PoE is only marginal.

8.3. Conceptualized Fog Computing Application

We conceive a real-world scenario in air quality prediction[5], where we want to predict the concentration of PM2.5 (considered to be scarcely available target data represented by green dots in Fig. 2), which is a crucial standard for clean air quality. However, the corresponding data is relatively hard to collect. The concentration level of PM2.5 is probably closely related to wind speed (considered to be widely accessible source data represented by red dots in Fig. 2). The data for wind speed is easy to collect by say multiple unmanned aerial vehicles (UAVs) flying over disjoint local areas. In other words, the UAVs may be seen as distributed fog nodes collecting source data with which the target predictions can be augmented. A local TGP model can thus be embedded into each UAV for online on-mote processing 30 UAVs (i.e., 30 fog nodes), and only 34 air quality stations constantly monitoring the concentration level of PM2.5.

Among all the target samples, 10 of them are randomly picked for training, and the remaining inputs are used for evaluation in our experimental study. We also explore the possibly non-uniform inter-task similarity in different local areas as a consequence of different localized geographical landscape characteristics. To this end, the Tr-BCM-ls model, as put forward in Section 6, is applied. The experimental results are shown in Table 4. Note that Tr-BCM-ls outperforms Tr-BCM, verifying our conjecture that the inter-task relationship probably varies with different geo-locations.

9. Conclusion

In this paper, we have introduced a theoretically principled aggregation model, namely transfer Bayesian Committee Machine (Tr-BCM), for transfer

[5] `https://biendata.com/competition/kdd_2018/`

Table 4: Results on the conceived fog computing application. The averaged RMSEs of different approaches are reported. Superior performance are highlighted using bold characters.

Methods	RMSE
Tr-BCM	30.7251±6.4189
Tr-BCM-ls	**28.5832**±7.2159
GP	31.5839±4.0998

learning with large-scale source inputs. The salient features of Tr-BCM are three-fold: (1) it offers a distributed lightweight alternative that is capable of replicating the full (heavyweight) TGP model; (2) by relaxing the uniformity condition on inter-task similarity capture, Tr-BCM can even enhance model expressiveness compared to TGP; (3) the relaxed Tr-BCM formulation directly applies to the multi-source transfer learning scenario where different sources can have differing inter-task relationship with the target.

The proposed aggregation model has been applied to synthetic as well as real-world datasets, with the experimental results verifying its efficacy over existing state-of-art transfer learning methods. Compared to traditional transfer learning methods, the accuracy and scalability of Tr-BCM are both theoretically and empirically shown to be superior with increasing amounts of source data, i.e., Wine, UJIIndoorLoc, and SARCOS. Interestingly, when deploying the proposed aggregation approach in certain distributed practical settings, each local expert serves as a lightweight predictor that can be embedded in edge devices, thus potentially catering to cases of online on-mote processing in fog computing environments.

Finally, with regard to future work, we note that the performance of the proposed Tr-BCM (and aggregation models in general) is at times sensitive to the data partitioning. In this regard, one promising direction is to incorporate sparse approaches and variational inference into our aggregation models in order to dynamically allocate data points to each local model in a more principled manner.

References

[1] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (10) (2010) 1345–1359.

[2] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, G. Zhang, Transfer learning using computational intelligence: A survey, Knowl.-Based Syst. 80 (2015) 14–23.

[3] A. W. M. Tan, R. Almandoz, A. Gupta, Y. S. Ong, Coping with data scarcity in aircraft engine design.

[4] H. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. J. Mollura, R. M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, IEEE Trans. Med. Imaging 35 (5) (2016) 1285–1298.

[5] J.-T. Huang, J. Li, D. Yu, L. Deng, Y. Gong, Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers, in: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, IEEE, 2013, pp. 7304–7308.

[6] N. Jaques, S. Taylor, E. Nosakhare, A. Sano, R. Picard, Multi-task learning for predicting health, stress, and happiness, in: NIPS Machine Learning for Health Care Workshop, 2016.

[7] K. Swersky, J. Snoek, R. P. Adams, Multi-task bayesian optimization, in: NIPS, 2013, pp. 2004–2012.

[8] L. Feng, Y. S. Ong, S. Jiang, A. Gupta, Autoencoding evolutionary search with learning across heterogeneous problems, IEEE Transactions on Evolutionary Computation PP (99) (2017) 1–1.

[9] H. Zuo, J. Lu, G. Zhang, F. Liu, Fuzzy transfer learning using an infinite gaussian mixture model and active learning, IEEE Transactions on Fuzzy Systemsdoi:10.1109/TFUZZ.2018.2857725.

[10] H. Zuo, J. Lu, G. Zhang, W. Pedrycz, Fuzzy rule-based domain adaptation in homogeneous and heterogeneous spaces, IEEE Transactions on Fuzzy Systemsdoi:10.1109/TFUZZ.2018.2853720.

[11] C. E. Rasmussen, Gaussian processes for machine learning.

[12] Y. Wang, B. Chaib-draa, Knn-based kalman filter: An efficient and non-stationary method for gaussian process regression, Knowl.-Based Syst. 114 (2016) 148–155.

[13] J. Hensman, A. G. de G. Matthews, Z. Ghahramani, Scalable variational gaussian process classification, in: AISTATS, Vol. 38 of JMLR Workshop and Conference Proceedings, JMLR.org, 2015.

[14] J. Snoek, H. Larochelle, R. P. Adams, Practical bayesian optimization of machine learning algorithms, in: NIPS, 2012, pp. 2960–2968.

[15] H. Liu, Y. Ong, J. Cai, Y. Wang, Cope with diverse data structures in multi-fidelity modeling: A gaussian process method, Eng. Appl. of AI 67 (2018) 211–225.

[16] N. D. Lawrence, Gaussian process latent variable models for visualisation of high dimensional data, in: NIPS, MIT Press, 2003, pp. 329–336.

[17] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, T. G. Dietterich, To transfer or not to transfer, in: NIPS 2005 Workshop on Transfer Learning, Vol. 898, 2005.

[18] E. V. Bonilla, K. M. Chai, C. Williams, Multi-task gaussian process prediction, in: Advances in neural information processing systems, 2007, pp. 153–160.

[19] B. Cao, S. J. Pan, Y. Zhang, D.-Y. Yeung, Q. Yang, Adaptive transfer learning., in: AAAI, Vol. 2, 2010, p. 7.

[20] N. Wagle, E. W. Frew, Forward adaptive transfer of gaussian process regression, J. Aerospace Inf. Sys. 14 (4) (2017) 214–231.

[21] P. Wei, R. Sagarna, Y. Ke, Y. Ong, C. Goh, Source-target similarity modelings for multi-source transfer gaussian process regression, in: ICML, Vol. 70 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 3722–3731.

[22] M. A. Álvarez, N. D. Lawrence, Sparse convolved gaussian processes for multi-output regression, in: NIPS, Curran Associates, Inc., 2008, pp. 57–64.

[23] M. A. Álvarez, D. Luengo, M. K. Titsias, N. D. Lawrence, Efficient multioutput gaussian processes through variational inducing kernels, in: AISTATS, Vol. 9 of JMLR Proceedings, JMLR.org, 2010, pp. 25–32.

[24] M. A. Álvarez, N. D. Lawrence, Computationally efficient convolved multiple output gaussian processes, Journal of Machine Learning Research 12 (2011) 1459–1500.

[25] V. Tresp, A bayesian committee machine, Neural Computation 12 (11) (2000) 2719–2741.

[26] G. E. Hinton, Products of experts.

[27] M. P. Deisenroth, J. W. Ng, Distributed gaussian processes, in: ICML, Vol. 37 of JMLR Workshop and Conference Proceedings, JMLR.org, 2015, pp. 1481–1490.

[28] C. E. Rasmussen, Z. Ghahramani, Infinite mixtures of gaussian process experts, in: NIPS, MIT Press, 2001, pp. 881–888.

[29] C. Yuan, C. Neubauer, Variational mixture of gaussian process experts, in: NIPS, Curran Associates, Inc., 2008, pp. 1897–1904.

[30] A. G. Wilson, D. A. Knowles, Z. Ghahramani, Gaussian process regression networks, in: ICML, icml.cc / Omnipress, 2012.

[31] T. V. Nguyen, E. V. Bonilla, Efficient variational inference for gaussian process regression networks, in: AISTATS, Vol. 31 of JMLR Workshop and Conference Proceedings, JMLR.org, 2013, pp. 472–480.

[32] P. G. López, A. Montresor, D. H. J. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. P. Barcellos, P. Felber, E. Rivière, Edge-centric computing: Vision and challenges, Computer Communication Review 45 (5) (2015) 37–42.

[33] F. Computing, the internet of things: Extend the cloud to where the things are, Cisco White Paper.

[34] E. V. Bonilla, F. V. Agakov, C. K. I. Williams, Kernel multi-task learning using task-specific features, in: AISTATS, Vol. 2 of JMLR Proceedings, JMLR.org, 2007, pp. 43–50.

[35] G. Leen, J. Peltonen, S. Kaski, Focused multi-task learning using gaussian processes, in: ECML/PKDD (2), Vol. 6912 of Lecture Notes in Computer Science, Springer, 2011, pp. 310–325.

[36] K. M. A. Chai, Generalization errors and learning curves for regression with multi-task gaussian processes, in: NIPS, Curran Associates, Inc., 2009, pp. 279–287.

[37] M. A. Álvarez, L. Rosasco, N. D. Lawrence, Kernels for vector-valued functions: A review, Foundations and Trends in Machine Learning 4 (3) (2012) 195–266.

[38] H. Liu, J. Cai, Y. Ong, Remarks on multi-output gaussian process regression, Knowl.-Based Syst. 144 (2018) 102–121.

[39] X. Wang, T. Huang, J. G. Schneider, Active transfer learning under model shift, in: ICML, Vol. 32 of JMLR Workshop and Conference Proceedings, JMLR.org, 2014, pp. 1305–1313.

[40] M. Kandemir, Asymmetric transfer learning with deep gaussian processes, in: ICML, Vol. 37 of JMLR Workshop and Conference Proceedings, JMLR.org, 2015, pp. 730–738.

[41] A. C. Damianou, N. D. Lawrence, Deep gaussian processes, in: AISTATS, Vol. 31 of JMLR Workshop and Conference Proceedings, JMLR.org, 2013, pp. 207–215.

[42] A. T. W. Min, Y. Ong, A. Gupta, C. Goh, Multi-problem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems, IEEE Transactions on Evolutionary Computation (2017) 1–1.

[43] M. Wistuba, N. Schilling, L. Schmidt-Thieme, Scalable gaussian process-based transfer surrogates for hyperparameter optimization, Machine Learning 107 (1) (2018) 43–78.

[44] J. Q. Candela, C. E. Rasmussen, A unifying view of sparse approximate gaussian process regression, Journal of Machine Learning Research 6 (2005) 1939–1959.

[45] E. Snelson, Z. Ghahramani, Sparse gaussian processes using pseudo-inputs, in: NIPS, 2005, pp. 1257–1264.

[46] J. Hensman, N. Fusi, N. D. Lawrence, Gaussian processes for big data, in: UAI, AUAI Press, 2013.

[47] Y. Gal, M. van der Wilk, C. E. Rasmussen, Distributed variational inference in sparse gaussian process regression and latent variable models, in: NIPS, 2014, pp. 3257–3265.

[48] Z. Dai, M. A. Álvarez, N. D. Lawrence, Efficient modeling of latent information in supervised learning using gaussian processes, in: NIPS, 2017, pp. 5137–5145.

[49] D. A. Moore, S. J. Russell, Gaussian process random fields, in: NIPS, 2015, pp. 3357–3365.

[50] L. Hogben, Handbook of linear algebra, CRC Press, 2006.

[51] D. Pardoe, P. Stone, Boosting for regression transfer, in: ICML, Omnipress, 2010, pp. 863–870.

[52] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis, Modeling wine preferences by data mining from physicochemical properties, Decision Support Systems 47 (4) (2009) 547–553.