Uncluttered domain sub-similarity modeling for transfer regression

Wei, Pengfei; Sagarna, Ramon; Ke, Yiping; Ong, Yew-Soon

2018

Wei, P., Sagarna, R., Ke, Y., & Ong, Y.-S. (2018). Uncluttered domain sub-similarity modeling for transfer regression. Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), 1314-1319. doi:10.1109/icdm.2018.00178

<https://hdl.handle.net/10356/143654>

<https://doi.org/10.1109/ICDM.2018.00178>

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: <https://doi.org/10.1109/ICDM.2018.00178>.

Downloaded on 26 Apr 2023 17:24:46 SGT

Uncluttered Domain Sub-similarity Modeling for Transfer Regression

Pengfei Wei, Ramon Sagarna, Yiping Ke, Yew-Soon Ong

School of Computer Science and Engineering, Rolls-Royce@NTU Corporate Lab

Nanyang Technological University, Singapore

pwei001@e.ntu.edu.sg, saramon@ntu.edu.sg, ypke@ntu.edu.sg, asysong@ntu.edu.sg

Abstract—Transfer covariance functions, which can model domain similarities and adaptively control the knowledge transfer across domains, are widely used in Gaussian process (GP) based transfer learning. We focus on regression problems in a black-box learning scenario, and study a family of rather general transfer covariance functions, T_* , that can model the similarity heterogeneity of domains through multiple kernel learning. A necessary and sufficient condition that (i) validates GPs using T_* for any data and (ii) provides semantic interpretations is given. Moreover, building on this condition, we propose a computationally inexpensive model learning rule that can explicitly capture different sub-similarities of domains. Extensive experiments on one synthetic dataset and four real-world datasets demonstrate the effectiveness of the learned GP on the sub-similarity capture and the transfer performance.

I. INTRODUCTION

Transfer learning has been proposed to deal with the problem in which train and test data are differently distributed. The idea is to learn a model that works well in a target domain by leveraging upon the knowledge from a different but related source domain. Although classification has received more research [1], an increasing attention is being paid to regression. In this setting, various real-world applications arise, e.g. Wi-Fi localization [2], sentiment analysis [3], and aerospace systems design [4]. In this work, we focus on transfer regression based on *Gaussian process* (GP) models.

A brute-force transfer assuming that domains are mutually relevant is inadvisable as it may well lead to negative transfer. Thus, the crucial point is to adapt the capture and transfer of relevant source knowledge to help the target tasks. Generally, this is complicated in real-world problems as the data from distinct domains are diverse and heterogeneous. So, two domains may share similarities in some characteristics but conflict with each other in the others. For example, in sentiment analysis, we may seek to transfer reviews knowledge across products, e.g., books and electronic devices. Both products exhibit the same sentiment on some words, e.g., “good” and “excellent”, but they perform inconsistently on some others, e.g., “noisy” and “comprehensive”. The similarity heterogeneity of domains, which refers to existing consistences and divergences among them, is widespread in real-world applications [5]. As such, methods that are capable of tuning the strength of knowledge transfer to the similarity heterogeneity are desirable.

A popular way of capturing inter-domain similarity in GP is to build a *transfer covariance function* (TCF). Instead of treat-

ing all instance pairs equally, a TCF differentiates the distance calculation of two instances based on the domains they come from. In this work, we focus on a TCF for problems with one source and one target domain. A general strategy is to use two distinct covariance functions for the distance calculation, i.e., k_{dd} if the instances are from the same domain, and $k_{dd'}$ otherwise. For example, [6], [7] set $k_{dd'} = \lambda k_{dd}$. Such a TCF, T_λ , uses one coefficient λ to model the similarity of domains. This is simple and computationally cheap but penalizes the ability to express the similarity heterogeneity of domains, as it can only reflect a compromise relation. By contrast, the more flexible *linear model of coregionalization* from geostatistics [8] adopts a TCF, T_* , using multiple kernels. Specifically, it is assumed that k_{dd} and $k_{dd'}$ lie in the same kernel function space and share the same base kernels. However, k_{dd} and $k_{dd'}$ may perform differently on the base kernels. They may present a divergence on the same base kernel; i.e., the coefficients of that kernel may differ dramatically in k_{dd} and $k_{dd'}$. They may also have a consistence on the same base kernel; i.e., k_{dd} and $k_{dd'}$ share the kernel coefficient. Such coefficients diversities on the base kernels play a role in modeling the similarity heterogeneity of domains.

Notwithstanding the above, any TCF must be valid so that the associated GP model is applicable for any data. This is equivalent to ensuring that the corresponding covariance matrix is always positive semi-definite (PSD). In some scenarios, this can be made easier by exploiting prior problem information to confine the form of the TCF, e.g. [9]. However, in many real-world applications, such knowledge is unavailable or cannot be used in a feasible way. These cases demand a black-box learning approach that (i) accounts for the complete set of valid covariance functions (to avoid overlooking the adequate ones), (ii) enables interpretable domain similarities (so that problem insights can be inferred), and (iii) is computationally feasible (as the learning of GP is prone to computational issues). Partial solutions exist in the literature, however methods spanning all the three aspects are not well studied (cf. Section II). For example, [10], [11] concentrate on enforcing PSD but “free-form” similarity matrices for T_λ and T_* by using the Cholesky decomposition or an EM strategy. [7] proposes an interpretable though only sufficient condition for the similarity coefficient to make T_λ PSD.

In this paper, we propose a condition for T_* with which the PSD problem is efficiently solved and T_* is semantically

regularized. The coefficient vectors of k_{dd} and of $k_{dd'}$ fully characterize T_* , and their element-wise distance indicates the similarity of the source and target domains on the corresponding base kernel (we call it *sub-similarity of two domains*). With the black-box learning, our aim is to precisely capture the heterogeneous sub-similarities of two domains. The proposed condition embraces all the three aspects described above in a simple way. It describes a necessary and sufficient (and hence complete) relationship between the coefficient vectors of k_{dd} and $k_{dd'}$ which readily delivers an interpretation in terms of sub-similarities. Thus, we can exploit this relationship as a computationally inexpensive constraint in the model learning. In the empirical evaluation, we first utilize an artificial dataset to demonstrate the capability of the resulting *GP* in the modeling of domains sub-similarities. We then show the transfer effectiveness of the *GP* on four real-world datasets as compared to other state-of-the-art baselines.

II. RELATED WORK

A covariance matrix derived from T_* (T_λ) can be written as a sum of Schur products, each involving a matrix for domains similarities, $\mathbf{K}_d$, and a matrix from a covariance function for input locations, $k_{\mathbf{x}}$. Once the model is learned, all $\mathbf{K}_d$ s stay fixed. Many approaches take advantage of this formulation.

A number of works follow a constraint handling strategy which is explicitly applied during the optimization. In [12], a T_* -like function is sought with simulated annealing. To check the feasibility of a candidate solution, each $\mathbf{K}_d$ term is tested for positive semidefiniteness. Provided that the $k_{\mathbf{x}}$ is PSD, this procedure guarantees the validity of the resulting *TCF*. However, the computation of the PSD tests can be cumbersome, e.g. typical approaches based on eigenvalues are subject to ill-conditioning issues. Also, a usual concern with metaheuristics is that the solution quality or runtime are highly sensitive to the tuning of the algorithm. This is evinced in [12] in the disparity of fittings obtained with three cooling schedules which govern the stopping condition of the search. An alternative direction is taken in [6], [10], [11], where the constraint handling is embedded in the search by means of an Expectation Maximization algorithm. At each step, the estimates for each base kernel in T_* (T_λ) can be decoupled: first for the hyperparameters of the $k_{\mathbf{x}}$, and then for the $\mathbf{K}_d$ matrix given the former. Further, the $\mathbf{K}_d$ is given in closed form and is PSD. Thus, the formulation is quite convenient, although the implementation of the expectations is somewhat involved. Another concern is that this method does not account for the interpretability of the $\mathbf{K}_d$ matrices being learned. Neither do the metaheuristic-based approaches, though they might be augmented with explicitly defined constraints for this.

Other works define an unconstrained parameterization of the $\mathbf{K}_d$ matrices enforcing them to be PSD, and subsequently apply the usual methods for optimization (e.g. maximizing log-likelihood). Since by using the Cholesky factorization, any PSD $\mathbf{K}_d$ can be written as $\mathbf{K}_d = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is lower triangular, a popular choice is to re-parameterize $\mathbf{K}_d$ on $\mathbf{L}$ [10], [13]. This ensures the resulting *TCF* is valid, however

it does not explain us whether all such valid functions are being considered or not (i.e. $\mathbf{K}_d$ could be non-PSD and the Schur product of $\mathbf{K}_d$ and a PSD matrix still be PSD). Also, the Cholesky parameterization lacks a clear-cut relationship between a given $\mathbf{L}$ and the elements in $\mathbf{K}_d$. This hampers any control on the semantic meaning of the similarities during learning. Moreover, the problem is exacerbated by the fact that the factorization may not be unique. Cao et al. [7], by contrast, enable an unconstrained parameterization for T_λ with $\lambda \in [-1, 1]$, which facilitates its interpretation in terms of correlations between the source and target domains. However, only a sufficient condition for PSD fulfillment is provided, leaving the behavior of the model for other λ values unclear. Further, as opposed to T_* , T_λ cannot deal with the similarity heterogeneity of domains, which may jeopardize its capability of handling complex real-world transfer tasks.

III. PROBLEM SETTING AND PRELIMINARIES

We consider the transfer regression setting with one source domain and one target domain. We denote the source by $\mathcal{S}$ and assign to it a set of n_S data points $\mathbf{X}^S \in \mathbb{R}^{n_S \times d}$ with corresponding labels $\mathbf{y}^S \in \mathbb{R}^{n_S}$. Likewise, we represent the target domain as $\mathcal{T}$, and attach to it a data matrix $\mathbf{X}^T \in \mathbb{R}^{n_T \times d}$ and a label vector $\mathbf{y}^T \in \mathbb{R}^{n_T}$. Typically, $n_T \ll n_S$. Our objective is to utilize $\mathbf{X} = [\mathbf{X}^S; \mathbf{X}^T] \in \mathbb{R}^{n \times d}$ and $\mathbf{y} = [\mathbf{y}^S; \mathbf{y}^T] \in \mathbb{R}^n$, with $n = n_S + n_T$, to build a model for $\mathcal{T}$.

In standard regression, we assume an underlying latent function between the input $\mathbf{x}$ and the output y , and a Gaussian noise term with variance σ^2 . This latent function can be represented as an arbitrarily long vector $\mathbf{f}$ of values at given input data locations. A *GP* model defines a Gaussian distribution over the functions, $\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$, with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{K}$. $\mathbf{K}$ must be PSD (also denoted $\mathbf{K} \succeq \mathbf{0}$), and is given by a parameterized covariance function over the input space. Usually, $\boldsymbol{\mu} = \mathbf{0}$ is assumed, and so the *GP* model is completely specified by the covariance function.

A *GP* model for the transfer learning setting defines a joint Gaussian distribution over the space of source and target functions. Thus, the representation of the dependencies between the functions is ruled by the form of the *TCF*. In [7], the following *TCF*, T_λ , is proposed for two domains:

$$k_\lambda(\mathbf{x}, \mathbf{x}') = \begin{cases} \lambda k(\mathbf{x}, \mathbf{x}'), & \mathbf{x} \in \mathbf{X}^S \text{ \& } \mathbf{x}' \in \mathbf{X}^T \\ & \text{or } \mathbf{x} \in \mathbf{X}^T \text{ \& } \mathbf{x}' \in \mathbf{X}^S, \\ k(\mathbf{x}, \mathbf{x}'), & \text{otherwise,} \end{cases} \quad (1)$$

where λ is a coefficient measuring the similarity of different domains. Note that k_λ fixes the similarity (to be 1) when points come from the same domain. Further, $\lambda \in [-1, 1]$ enables a PSD T_λ while exposing more rationales. For $|\lambda| = 1$, T_λ can model the extreme scenario in which the source and target domains are consistently related, and so all the source data is fully used in transfer. For $|\lambda| = 0$, T_λ models another extreme case in which source and target are completely unrelated, and thus only target data is used to avoid negative transfer. For $|\lambda| < 1$, T_λ fine-tunes the impact the source domain has on the target domain.

Even if semantically engaging, relying on such λ hampers the capture of the similarity heterogeneity of domains. To overcome this, we may consider using two different covariance functions that account for the heterogeneity. Here, we assume that both functions are in the same kernel function space expanded by a set of independent base kernels $k_i, i = 1, 2, \dots, N$ where each k_i is a valid covariance function and N can be infinite. Such a TCF, T_* , corresponds to the *linear model of coregionalization* in geostatistics [8]:

$$k_*(\mathbf{x}, \mathbf{x}') = \begin{cases} \sum_i \beta_i k_i(\mathbf{x}, \mathbf{x}'), & \mathbf{x} \in \mathbf{X}^S \text{ \& } \mathbf{x}' \in \mathbf{X}^T \\ & \text{or } \mathbf{x} \in \mathbf{X}^T \text{ \& } \mathbf{x}' \in \mathbf{X}^S, \\ \sum_i \alpha_i k_i(\mathbf{x}, \mathbf{x}'), & \text{otherwise.} \end{cases} \quad (2)$$

We set $\alpha_i, i = 1, \dots, N$ to be non-negative as they are the similarity coefficients for the case that domains are the same. We derive the condition that the β_i s should satisfy later on.

The intuition behind T_* is that the two different sets of coefficient variables, α_i s and β_i s, allow diverse similarities on the base kernels, and thus represent the heterogeneous sub-similarities between the source and target domains. Specifically, if β_i is close to α_i , the source domain is closely related to the target domain on the base kernel k_i . On the contrary, if β_i is very dissimilar to α_i , it means that the two domains differ considerably on the base kernel k_i . The similarity of β_i and α_i indicates how much knowledge can be transferred across domains on k_i .

IV. TRANSFER MODEL USING T_*

With T_* we can model the similarity heterogeneity of domains. However, the learned GP_{T_*} should be a valid GP model for any data. This prevents β_i s and α_i s from taking arbitrary values. Next, we provide a necessary and sufficient condition on the relationship of β_i s and α_i s that makes T_* (and hence GP_{T_*}) valid. We then introduce a learning rule for GP_{T_*} that exploits this relationship.

A. On the Validity of GP_{T_*} for Any Data

To make T_* valid we must guarantee that a covariance matrix derived from it will be PSD. Theorem 1 facilitates this.

Theorem 1 A Gram matrix

$$\mathbf{K}_* = \begin{bmatrix} \sum_{i=1}^N \alpha_i \mathbf{K}_{SS}^i & \sum_{i=1}^N \beta_i \mathbf{K}_{ST}^i \\ \sum_{i=1}^N \beta_i \mathbf{K}_{TS}^i & \sum_{i=1}^N \alpha_i \mathbf{K}_{TT}^i \end{bmatrix}$$

with $\alpha_i > 0$ is PSD for any covariance matrices $\mathbf{K}^i$ in the form

$$\mathbf{K}^i = \begin{bmatrix} \mathbf{K}_{SS}^i & \mathbf{K}_{ST}^i \\ \mathbf{K}_{TS}^i & \mathbf{K}_{TT}^i \end{bmatrix},$$

if and only if $|\beta_i| \leq \alpha_i$ for $i = 1, 2, \dots, N$.

Proof 1 We first rewrite $\mathbf{K}_* = \sum_{i=1}^N \alpha_i \mathbf{K}_*^i$, where

$$\mathbf{K}_*^i = \begin{bmatrix} \mathbf{K}_{SS}^i & \beta_i/\alpha_i \mathbf{K}_{ST}^i \\ \beta_i/\alpha_i \mathbf{K}_{TS}^i & \mathbf{K}_{TT}^i \end{bmatrix}. \quad (3)$$

Necessary condition: Since $\mathbf{K}_*$ is PSD for **any** covariance matrices $\mathbf{K}^i, i = 1, 2, \dots, N$ and $\alpha_i > 0$, we can derive

$$\mathbf{K}_*^i \succeq \mathbf{0}, i = 1, 2, \dots, N.$$

According to the Schur complement theorem [14], we derive:

$$\mathbf{K}_{TT}^i - (\beta_i/\alpha_i)^2 (\mathbf{K}_{ST}^i)^\top (\mathbf{K}_{SS}^i)^{-1} \mathbf{K}_{ST}^i \succeq \mathbf{0}. \quad (4)$$

Since $\mathbf{K}^i \succeq \mathbf{0}, i = 1, 2, \dots, N$, we have:

$$\mathbf{K}_{TT}^i - (\mathbf{K}_{ST}^i)^\top (\mathbf{K}_{SS}^i)^{-1} \mathbf{K}_{ST}^i \succeq \mathbf{0}.$$

We then rewrite Eq. (4):

$$\begin{aligned} & \mathbf{K}_{TT}^i - (\beta_i/\alpha_i)^2 \mathbf{K}_{TT}^i + (\beta_i/\alpha_i)^2 \mathbf{K}_{TT}^i \\ & - (\beta_i/\alpha_i)^2 (\mathbf{K}_{ST}^i)^\top (\mathbf{K}_{SS}^i)^{-1} \mathbf{K}_{ST}^i \succeq \mathbf{0}. \end{aligned}$$

By denoting $\mathbf{M} = \mathbf{K}_{TT}^i - (\mathbf{K}_{ST}^i)^\top (\mathbf{K}_{SS}^i)^{-1} \mathbf{K}_{ST}^i$, we have:

$$\mathbf{M} \succeq \mathbf{0}, \quad (5)$$

$$(1 - (\beta_i/\alpha_i)^2) \mathbf{K}_{TT}^i + (\beta_i/\alpha_i)^2 \mathbf{M} \succeq \mathbf{0}, \quad (6)$$

Since $\mathbf{K}_{TT}^i \succeq \mathbf{0}$ and Eq. (6) must hold for all $\mathbf{K}^i$, we induce:

$$(1 - (\beta_i/\alpha_i)^2) \geq 0, i = 1, 2, \dots, N,$$

and thus we have: $|\beta_i| \leq \alpha_i$ for $i = 1, 2, \dots, N$.

Sufficient condition: We consider:

$$\begin{aligned} \mathbf{N} &= \mathbf{K}_{TT}^i - (\beta_i/\alpha_i)^2 (\mathbf{K}_{ST}^i)^\top (\mathbf{K}_{SS}^i)^{-1} \mathbf{K}_{ST}^i \\ &= \mathbf{K}_{TT}^i - (\mathbf{K}_{ST}^i)^\top (\mathbf{K}_{SS}^i)^{-1} \mathbf{K}_{ST}^i \\ &+ (1 - (\beta_i/\alpha_i)^2) (\mathbf{K}_{ST}^i)^\top (\mathbf{K}_{SS}^i)^{-1} \mathbf{K}_{ST}^i \end{aligned}$$

Since $\mathbf{M} \succeq \mathbf{0}$ and $|\beta_i| \leq \alpha_i$ for $i = 1, 2, \dots, N$, we obtain $\mathbf{N} \succeq \mathbf{0}$. Considering $\mathbf{K}_{SS}^i \succeq \mathbf{0}$, we obtain $\mathbf{K}_*^i \succeq \mathbf{0}$ based on the Schur complement theorem. As the conical sum of PSD matrices is also PSD, we obtain that $\mathbf{K}_*$ is PSD.

In summary, for $i = 1, 2, \dots, N$, if $\mathbf{K}_*$ is PSD with $\alpha_i > 0$ for any $\mathbf{K}^i$, the coefficients β_i s should satisfy $|\beta_i| \leq \alpha_i$.

Note that the proof of PSD $\mathbf{K}_*^i$ in the sufficient condition can also be obtained by using Theorem 1 in [7]. To be complete, we give the full proof. Moreover, when $\alpha_i = 0$, the corresponding base kernel k_i does not contribute to k_{dd} , and thus no knowledge on k_i can be transferred across domains. In this case, the coefficient for $k_{dd'}$, β_i , must be 0 too.

When applying Theorem 1, the semantics of T_* are as follows. Looking at each base kernel k_i , Theorem 1 states that $|\beta_i|$ never goes beyond α_i . This is reasonable as α_i alludes to k_{dd} , which gives the distance of two points from the same domain, and thus binds the maximum achievable transfer when the domains are different. The extreme cases $|\beta_i| = \alpha_i$ and $\beta_i = 0$ indicate full knowledge and no knowledge transfer on kernel k_i . When considering all the base kernels, T_* enables rich and diverse sub-similarities between the source and target domains. The case $|\beta_i| = \alpha_i, i = 1, \dots, N$, only happens when both domains are identical, while $|\beta_i| = 0, i = 1, \dots, N$, only happens when domains are completely unrelated. For other cases $|\beta_i| < \alpha_i, i = 1, \dots, N$, T_* can give a fine modeling of the nuances between the two domains.

B. Learning of GP_{T_*}

We can use numerical methods such as conjugate gradient for the learning of $\theta := \{\alpha_1 \dots \alpha_N, \beta_1 \dots \beta_N, \delta_S^2, \delta_T^2\}$, where δ_S^2 and δ_T^2 are the noise variance for source and target. In particular, here we follow [7] and optimize the log-likelihood of the conditional distribution of the target $p(\mathbf{y}^T | \mathbf{X}, \mathbf{y}^S)^1$.

Importantly, Theorem 1 can be easily exploited to account for the relationship of α_i s and β_i s in the optimization. In the l -th iteration during learning, we calculate the updated α_i^l and use $\max(0, \alpha_i^l)$ to avoid negative values. Next, we calculate the updated β_i^l and if the constraint $|\beta_i| \leq \alpha_i$ is violated, we project β_i^l to a value close to the boundary as follows:

$$\beta_i^l = \begin{cases} \alpha_i^l - \epsilon_i^l, & \text{if } \beta_i^l > \alpha_i^l, \\ -\alpha_i^l + \epsilon_i^l, & \text{if } \beta_i^l < -\alpha_i^l, \end{cases}$$

where ϵ_i^l is a variable with tiny random value guaranteed to be in $[0, 2\alpha_i^l]$. We use ϵ_i^l to explore more diverse values of β_i^l in the search. After obtaining the (sub)optimal θ^{opt} the output for a new test target point can be inferred as usual in GPs .

V. EXPERIMENTAL STUDIES

Next, we aim at empirically assessing the capability of GP_{T_*} to capture the sub-similarities between domains, and evaluating its predictive effectiveness on new target test data. All the GP models herein are optimized using the conjugate gradient implementation from the *gpml* package [15]. For each search, we allow a maximum of 200 iterations. The reported results correspond to the model providing the best objective function value over 10 independent runs. Root mean squared error (RMSE) is used as the prediction evaluation metric. We use one synthetic dataset and four real-world datasets¹.

Synthetic data. We build on a T_* with predefined base kernels k_i s, and ground-truth combination coefficients α_i^{gt} s and β_i^{gt} s for k_{dd} and $k_{dd'}$, respectively. For a set of inputs $\mathbf{X}$, we obtain target function values $\mathbf{f}_{tar}$ from a GP with zero mean and covariance matrix given by $k_{dd} = \sum_i \alpha_i^{gt} k_i$. We then sample source function values $\mathbf{f}_{src}$ from $p(\mathbf{f}_{src} | \mathbf{f}_{tar}, \mathbf{X})$. By varying α_i^{gt} s and β_i^{gt} s, we can control the ground-truth sub-similarity $sim_i = \beta_i^{gt} / \alpha_i^{gt}$ of domains on each base kernel k_i .

Wine Quality. This dataset includes red (R) and white (W) wine samples [16]. Physicochemical variables, e.g. volatile acidity, are used as features and the sensory variable is used as label. We take R and W as the domains, and construct two transfer tasks with the form $\mathcal{S}-\mathcal{T}$: R-W and W-R.

Appliance Energy Prediction. This dataset includes the appliance energy use in different months [17]. House conditions data monitored by the sensors (e.g. humidity) give the features, and the label is the amount of appliance energy use. We select January and May as two domains, and then form two transfer tasks J-M and M-J.

Air Quality. This dataset contains responses of gas multisensor devices from different seasons [18]. The averaged

Figure 1. (a) Data points generated from GPs with distinct exponential kernels; red ‘o’s from $k_1 = 0.1 * \exp(-\frac{x^2}{2*1^2})$; black ‘△’s from $k_2 = 0.01 * \exp(-\frac{x^2}{2*0.1^2})$; blue ‘□’s from $k = k_1 + k_2$. (b) Source points from four special cases $(\beta_1^{gt}, \beta_2^{gt}) \in \{(1, 1), (1, 0), (0, 1), (0, 0)\}$. Notice that for (1,1) the points precisely follow the target function values in (a).

responses from the metal oxide chemical sensors form the features, and the ground-truth provided by a co-located reference certified analyzer is the label. We use Summer and Winter as domains, and then form two transfer tasks S-W and W-S.

UJIIndoorLoc. This dataset covers three buildings of Universitat Jaume I [19]. The received signal strength intensity from 520 wireless access points is used as the features, and the location given by the latitude and longitude is taken as label. By taking the location data from each building B_0, B_1, B_2 as a domain, we obtain 6 transfer tasks B_i-B_j , $i, j \in \{0, 1, 2\}$.

For the real-world datasets, each domain has two sets of points, set1 and set2. When a domain is selected as source, set1 is used for training; and when selected as target, set2 is used for training and set1 for testing. In UJIIndoorLoc, set1 and set2 are of size 1000 and 25 respectively. In the rest of the datasets, set1 and set2 are of size 500 and 10 respectively.

A. Domain Sub-Similarity Capture

We construct a broad spectrum of transfer problems upon the synthetic data in order to analyze the distance between the learned sub-similarity $\beta_i^{opt} / \alpha_i^{opt}$ and the ground-truth similarity sim_i for all the base kernels. Specifically, we generate 500 inputs $\mathbf{X} \in \mathbb{R}^{500 \times 1}$ and predefine two base kernels (k_1 and k_2) using two exponential covariance functions with different amplitudes and length scales. As shown in Fig. 1(a), k_1 draws ‘big trends’ while k_2 depicts ‘small perturbations’. Next, we generate target and source function values based on k_* in Eq. (2). We fix $\alpha_1^{gt} = 1$ and $\alpha_2^{gt} = 1$, but vary β_1^{gt} and β_2^{gt} in a range set $\mathcal{V} = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. Thus, the corresponding sim_i s also vary in $\mathcal{V}$. We then generate $\mathbf{f}_{tar}$ from a GP using $k_1 + k_2$. Fig. 1(a) shows that the sampled target points possess the characteristics of both kernels. Given $\mathbf{f}_{tar}$, for each combination $(\beta_1^{gt}, \beta_2^{gt})$ we generate source points using $p(\mathbf{f}_{src} | \mathbf{f}_{tar}, \mathbf{X})$. Fig. 1(b) shows four special cases $(\beta_1^{gt}, \beta_2^{gt}) \in \{(1, 1), (1, 0), (0, 1), (0, 0)\}$. In total, we can generate $C_6^1 \times C_6^1 = 36$ different source functions, and as many transfer problems. For each of them, we use 380 points as the source train data, 20 as the target train data, and the rest 100 as the target test data. Note that the function values of source train data vary across the 36 problems function, while the values of the target train and test stay the same.

¹An appendix with details on the likelihood and its derivatives, as well as the codes and datasets used in the experiments, are available at: <https://sites.google.com/site/weipengfeishomepage/>.

Table I
RESULTS OF GP_{T_*} , $GP_{T_{\lambda mk}}$ AND $GP_{T_{ar}}$.

Problem (sim_1, sim_2)	Learned Similarity		RMSE	
	GP_{T_*}	$GP_{T_{\lambda mk}}$	GP_{T_*}	$GP_{T_{\lambda mk}}$
(0,0)	(0.057,0.008)	0.042	0.279	0.280
(0,0.2)	(0.087,0.174)	0.114	0.264	0.266
(0,0.4)	(0.015,0.443)	0.080	0.254	0.280
(0,0.6)	(0.080,0.623)	0.155	0.216	0.300
(0,0.8)	(0.079,0.805)	0.227	0.171	0.293
(0,1)	(0.015,1)	0.838	0.122	0.152
(0.2,0)	(0.247,0.056)	0.136	0.247	0.271
(0.2,0.2)	(0.190,0.231)	0.198	0.237	0.237
(0.2,0.4)	(0.195,0.364)	0.210	0.221	0.225
(0.2,0.6)	(0.214,0.616)	0.080	0.181	0.302
(0.2,0.8)	(0.2,0.763)	0.516	0.162	0.202
(0.2,1)	(0.184,1)	0.875	0.112	0.188
(0.4,0)	(0.375,0.039)	0.261	0.235	0.267
(0.4,0.2)	(0.438,0.172)	0.301	0.220	0.236
(0.4,0.4)	(0.420,0.446)	0.433	0.175	0.176
(0.4,0.6)	(0.383,0.644)	0.539	0.124	0.146
(0.4,0.8)	(0.410,0.763)	0.515	0.117	0.151
(0.4,1)	(0.686,1)	0.907	0.108	0.137
(0.6,0)	(0.644,0.078)	0.498	0.212	0.247
(0.6,0.2)	(0.611,0.173)	0.488	0.199	0.230
(0.6,0.4)	(0.580,0.413)	0.537	0.164	0.173
(0.6,0.6)	(0.578,0.565)	0.580	0.111	0.110
(0.6,0.8)	(0.583,0.760)	0.849	0.109	0.153
(0.6,1)	(0.741,1)	0.969	0.025	0.039
(0.8,0)	(0.793,0.032)	0.468	0.191	0.208
(0.8,0.2)	(0.807,0.254)	0.543	0.152	0.206
(0.8,0.4)	(0.757,0.426)	0.556	0.125	0.156
(0.8,0.6)	(0.792,0.611)	0.678	0.093	0.105
(0.8,0.8)	(0.798,0.800)	0.806	0.060	0.061
(0.8,1)	(0.768,1)	0.951	0.037	0.083
(1,0)	(0.999,0.062)	0.826	0.074	0.120
(1,0.2)	(1,0.094)	0.801	0.076	0.153
(1,0.4)	(0.999,0.364)	0.984	0.066	0.081
(1,0.6)	(1,0.457)	0.937	0.068	0.075
(1,0.8)	(1,0.812)	0.981	0.041	0.048
(1,1)	(0.995,0.999)	1.000	0.001	0.001
$GP_{T_{ar}}$ RMSE		0.296		

Table I shows the learned sub-similarity ($\beta_1^{opt}/\alpha_1^{opt}$, $\beta_2^{opt}/\alpha_2^{opt}$) and the transfer performance. The results of two other GP models with multiple kernels are also included: $GP_{T_{ar}}$, which uses target train data only, and $GP_{T_{\lambda mk}}$, which adopts a variant of T_λ where $k(\mathbf{x}, \mathbf{x}') = \sum_i \eta_i k_i(\mathbf{x}, \mathbf{x}')$ (see Eq. (1)). Notice that the latter TCF becomes a special case of T_* with $\beta_i = \lambda \alpha_i$. That is, the same similarity coefficient λ is used on all the base kernels, which implies the model does not account for sub-similarity heterogeneities.

Paying attention to the domain sub-similarities capture, Table I reveals that, overall, GP_{T_*} successfully recovers the sim_i s on different problems. We show the distance of ($\beta_1^{opt}/\alpha_1^{opt}$, $\beta_2^{opt}/\alpha_2^{opt}$) to (sim_1, sim_2) more clearly in Fig. 2. By setting a distance threshold as half of the step size of $\mathcal{V}$, i.e. 0.1, we obtain that in 32 out of 36 problems, the distance is below the threshold. In the figure, the other four cases are pinpointed using the red arrow. This outcome indicates the effectiveness of GP_{T_*} in capturing the heterogeneous sub-

Figure 2. The red ‘o’s are (sim_1, sim_2) on k_1 and k_2 , and the blue ‘□’s are ($\beta_1^{opt}/\alpha_1^{opt}, \beta_2^{opt}/\alpha_2^{opt}$) learned by GP_{T_*} .

similarities of domains. On the other side, the single coefficient λ learned by $GP_{T_{\lambda mk}}$ can accurately reflect the similarity value for the problems where $sim_1 = sim_2$ (in bold in Table I). For the problems $sim_1 \neq sim_2$, however, the λ is usually between sim_1 and sim_2 , and so can only describe a compromised similarity.

We now turn to the transfer performance. Firstly, from Table I we observe that both $GP_{T_{\lambda mk}}$ and GP_{T_*} can achieve adaptive transfer as they are not worse than the no-transfer method $GP_{T_{ar}}$. By browsing the table it can be noticed that the RMSE of both models tends to decrease as the two domains similarities increase, which is aligned with the cases in which the models are transferring more knowledge. This further supports the adaptiveness of the two methods. Also, we find that both models perform similar in problems where $sim_1 = sim_2$. However, for the other cases GP_{T_*} consistently outperforms $GP_{T_{\lambda mk}}$, suggesting that the former can better deal with more complex transfer tasks than the latter. We believe that the performance improvement of GP_{T_*} to $GP_{T_{\lambda mk}}$ is due to its capability of precisely capturing the sub-similarities of domains. Conceptually, each of GP_{T_*} and $GP_{T_{\lambda mk}}$ are equivalent to a linear combination of GPs, one for each base kernel. For the i th term, the coefficient refers to the weight of the base function component i on the function space, and the GP models the sub-similarity between source and target for this component. For $GP_{T_*}^i$ and $GP_{T_{\lambda mk}}^i$, the respective coefficients are α_i and η_i , and the sub-similarities are given by β_i/α_i and λ . Provided both models approximate the linear coefficients similarly, if β_i/α_i is the ground-truth similarity on k_i , $GP_{T_*}^i$ should be no worse than $GP_{T_{\lambda mk}}^i$ on the transfer performance. Thus, we would expect GP_{T_*} to be also no worse than $GP_{T_{\lambda mk}}$.

B. Performance on Real-World Datasets

We compare the performance of GP_{T_*} with baselines $GP_{T_{\lambda mk}}$, $GP_{T_{ar}}$ and GP_{All} . The latter fully utilizes all the source and target train data in the model learning. All the baselines are implemented with multiple kernels. Specifically, we predefine 10 base kernels: one linear, two polynomial with

Figure 3. Comparison results on four real-world datasets.

degrees 2 and 3, and seven exponential with length scale taking values in $\{0.01, 0.1, 1, 5, 15, 20\}$. Fig. 3 shows the results on the 12 transfer tasks derived from the four real-world datasets.

We can observe that GP_{T_*} is a clear winner among all the baselines across all the tasks. More precisely, GP_{Tar} is the worst in most cases as it does not transfer any knowledge from source domain. However, we note that GP_{All} is even worse in tasks **W-S** and **S-W**. This indicates us such a brute-force way of transferring all the source knowledge may even cause negative transfer. In contrast, both $GP_{T_{\lambda mk}}$ and GP_{T_*} consistently achieve positive transfer in all the transfer tasks compared to GP_{Tar} . This is because these two models take the domain similarity into account in transfer, and thus can adaptively transfer the shared knowledge and avoid negative transfer. Also, GP_{T_*} outperforms $GP_{T_{\lambda mk}}$ in all the tasks. This is aligned with the test on the artificial dataset, and further supports the capability of finely modeling different sub-similarities between domains by GP_{T_*} , as opposed to the compromised similarity heterogeneity by $GP_{T_{\lambda mk}}$.

VI. CONCLUSIONS

We focus on GPs using a TCF for regression in a black-box learning scenario. We investigate a family of general $TCFs$, T_* , that are capable of modeling the similarity heterogeneity of domains through multiple kernel learning. A necessary and sufficient condition that encodes the relationship of the coefficients for the base kernels is presented. Based on this condition, T_* is guaranteed to be a valid covariance function and is semantically regularized. We then propose a computationally inexpensive constraint for GP_{T_*} learning with which we can explicitly capture the sub-similarities between domains. Extensive experiments on one synthetic and four real-world datasets show the effectiveness of GP_{T_*} on capturing sub-similarities and on the transfer adaptiveness.

ACKNOWLEDGMENT

This work was conducted within the Rolls-Royce@NTU Corporate Lab with support from the National Research

Foundation (NRF) Singapore under the Corp Lab@University Scheme. This work is supported in part by the AcRF Tier-1 Grant (RG135/14) from Ministry of Education of Singapore. This work is partially supported by the Data Science and Artificial Intelligence Research Centre (DSAIR) and the School of Computer Science and Engineering at Nanyang Technological University.

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] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *Neural Networks, IEEE Transactions on*, vol. 22, no. 2, pp. 199–210, 2011.
- [3] P. Wei, Y. Ke, and C.-K. Goh, "Deep nonlinear feature coding for unsupervised domain adaptation," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI'16. AAAI Press, 2016, pp. 2189–2195.
- [4] A. Tan Wei Min, R. Sagarna, A. Gupta, Y.-S. Ong, and C.-K. Goh, "Knowledge transfer through machine learning in aircraft design," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 48–60, 2017.
- [5] S. Sun, "A survey of multi-view machine learning," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2031–2038, 2013.
- [6] K. M. Chai, C. Williams, S. Klankke, and S. Vijayakumar, "Multi-task gaussian process learning of robot inverse dynamics," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. MIT Press, 2009, pp. 265–272.
- [7] B. Cao, S. J. Pan, Y. Zhang, D.-Y. Yeung, and Q. Yang, "Adaptive transfer learning," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*. AAAI Press, 2010, pp. 407–712.
- [8] H. Wackernagel, *Multivariate Geostatistics: An Introduction with Applications*. Springer Science & Business Media, 1998.
- [9] T. Cohn and L. Specia, "Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, ser. ACL, 2013, pp. 32–42.
- [10] E. V. Bonilla, K. M. Chai, and C. Williams, "Multi-task gaussian process prediction," in *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2008, pp. 153–160.
- [11] H. Zhang, "Maximum-likelihood estimation for multivariate spatial linear coregionalization models," *Environmetrics*, vol. 18, no. 2, pp. 125–139, 2007.
- [12] R. M. Lark and A. Papritz, "Fitting a linear model of coregionalization for soil properties using simulated annealing," *Geoderma*, vol. 115, pp. 245–260, 2003.
- [13] K. M. Chai, "Multi-task learning with gaussian processes," Ph.D. dissertation, The University of Edinburgh, 2010.
- [14] F. Zhang, *The Schur Complement and Its Applications*. Springer Science & Business Media, 2005, vol. 4.
- [15] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (GPML) toolbox," *Journal of Machine Learning Research*, vol. 11, pp. 3011–3015, Dec. 2010.
- [16] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems*, vol. 47, no. 4, pp. 547–553, 2009.
- [17] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy and Buildings*, vol. 140, pp. 81–97, 2017.
- [18] S. De Vito, G. Fattoruso, M. Pardo, F. Tortorella, and G. Di Francia, "Semi-supervised learning techniques in artificial olfaction: A novel approach to classification problems and drift counteraction," *IEEE Sensors Journal*, vol. 12, no. 11, pp. 3215–3224, 2012.
- [19] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta, "UJIIndoorloc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2014, pp. 261–270.