Source-Target Similarity Modelings for Multi-Source Transfer Gaussian Process Regression

Pengfei Wei [1 2] **Ramon Sagarna** [1 2] **Yiping Ke** [1 2] **Yew-Soon Ong** [1 2] **Chi-Keong Goh** [3 2]

Abstract

A key challenge in multi-source transfer learning is to capture the diverse inter-domain similarities. In this paper, we study different approaches based on Gaussian process models to solve the multi-source transfer regression problem. Precisely, we first investigate the feasibility and performance of a family of transfer covariance functions that represent the pairwise similarity of each source and the target domain. We theoretically show that using such a transfer covariance function for general Gaussian process modelling can only capture the same similarity coefficient for all the sources, and thus may result in unsatisfactory transfer performance. This leads us to propose $TC_{MS}Stack$, an integrated strategy incorporating the benefits of the transfer covariance function and stacking. Extensive experiments on one synthetic and two real-world datasets, with learning settings of up to 11 sources for the latter, demonstrate the effectiveness of our proposed $TC_{MS}Stack$.

1. Introduction

Transfer learning (*TL*) methods show specially appealing for real-world applications where the data from the *target* domain is scarce but a good amount of data from another *source* domain is available. With research efforts largely confined to the single-source setting (Pan et al., 2011; Wei et al., 2016; Zhou et al., 2016), an increasing amount of studies are contributing to a realistic applicability of *TL* by addressing the multi-source scenario, mainly for classification (Tommasi et al., 2014; Fang et al., 2015; Bhatt et al., 2016). The problem of regression, however, has been much less studied, despite of the variety of real-world domains in which it arises; for instance wifi or indoor signal location (Pan et al., 2008), biological data analysis (Lam et al., 2016), or mechanical system design (Ghosh et al., 2015). In this work, we concentrate on *multi-source transfer regression* (*MSTR*) based on *Gaussian process* (*GP*) models.

All the way through, the *TL* community has been paying attention to modeling the similarity between different domains so that only the source knowledge that is helpful for the target domain is transferred. This is because designing a *TL* method based on the assumption that domains are mutually relevant may lead to negative transfer (Pan & Yang, 2010). Similarity capture is particularly crucial in multi-source *TL* as the transfer capacity to the target task may differ considerably across the diverse source domains. Thus, *TL* methods that are capable of tuning the strength of the knowledge transfer to the similarity of the domains are attracting increasing interest (Luo et al., 2008; Wang et al., 2014; Al-Stouhi & Reddy, 2011).

As regards to *MSTR*, a key issue is to capture the diverse *Source-Target* (*S-T*) similarities. The relatively few efforts to date have focused on ensemble methods. Particularly, an amount of works rely on the boosting strategy due to its capability to capture fine-grained *S-T* similarities be weighting the contribution of train instances individually (Dai et al., 2007; Pardoe & Stone, 2010; Yao & Doretto, 2010). However, as outlined in (Al-Stouhi & Reddy, 2011), such an instance-based similarity strategy in boosting has shown issues with slow/premature weights convergence that have seriously penalized the computational cost or the transfer performance. Another type of ensemble strategy for multi-source transfer is stacking (Wolpert, 1992). Pardoe and Stone propose a meta-model that aggregates the predictions of several base models previously learned with each source in isolation (Pardoe & Stone, 2010) . The aggregation is done by assigning each base model a model importance. In this case, the *S-T* similarities can be captured through the model importance. However, in such stacking-based methods, the base models suffer from a lack of consideration of the dependencies between the different source domains.

[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore [2]Rolls-Royce@Nanyang Technological University Corporate Lab [3]Rolls-Royce Advanced Technology Centre, Singapore. Correspondence to: Pengfei Wei <Pwei001@e.ntu.edu.sg>, Ramon Sagarna <saramon@ntu.edu.sg>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

Another popular idea to model the *S-T* similarity is to construct a transfer covariance function that relates two data points from distinct domains through the similarity coefficients (Bonilla et al., 2008; Williams et al., 2009). Such idea has been proposed in multi-task learning (Bonilla et al., 2008), where each task pair is assigned a particular similarity coefficient. Note, however, that multi-task learning differs from the *TL* problem in that the former aims at improving performance across all the domains while the objective of the latter focuses on the target domain only. Nevertheless, the idea of transfer covariance function is referential for the *TL* problem. In (Cao et al., 2010), a *single source transfer covariance function* (TC_{SS}) was proposed. In the corresponding transfer covariance matrix, one similarity coefficient was assigned to the *S-T* block to model the inter-domain similarity. A *GP* with such TC_{SS} (called *GP-*TC_{SS}) was then trained for the transfer task.

When generalizing to *MSTR*, one may naturally consider a *multi-source transfer covariance function* (TC_{MS}) with different similarity coefficients attached to distinct *S-T* blocks in the corresponding transfer covariance matrix. In this work, we investigate the feasibility of such covariance function. We theoretically prove that a general *GP* with TC_{MS} (*GP-*TC_{MS}) fails to capture the similarity diversities of various *S-T* domain pairs. Although TC_{MS} intends to utilize different similarity coefficients, the learnt *GP-*TC_{MS} would give the same similarity coefficient for all the *S-T* domain pairs. The generalization error bounds of the learnt *GP-*TC_{MS} show that this coefficient is taking effect in every source domain. Considering the diverse *S-T* similarities between the sources and the target, this may jeopardize the transfer performance, especially when the number of sources increases. Moreover, the learning of *GP-*TC_{MS} rapidly poses a computational issue with increasing amounts of source domains as usually $\mathcal{O}(m^3)$ computations are required to evaluate a model for m data points.

The unsatisfactory performance of *GP-*TC_{MS} leads us to exploit the transfer covariance function in another way. Considering that both the stacking strategy and the transfer covariance function can model the *S-T* similarity and using the transfer covariance function at the base models would therefore add flexibility to the similarity capture capability of the stacking approach, we propose to integrate them into one unified model. Specifically, we first discuss TC_{SS}*Stack*, a method that simply stacks *GP-*TC_{SS} base models. TC_{SS}*Stack* alleviates the computational issue of *GP* since it allows to stretch the number of sources due to its $\mathcal{O}(Nn^3)$ cost for N sources with n points each. However, TC_{SS}*Stack* still suffers from the aforementioned limitation of conventional stacking. Thus, we propose a more involved TC_{MS}*Stack*. Two salient features make TC_{MS}*Stack* significantly different from TC_{SS}*Stack*: (i) it associates the similarity coefficients in the base *GP-*TC_{SS} with the model importance during learning, and (ii) it learns the model importance and the base *GP-*TC_{SS} jointly. By doing so, on the one hand, TC_{MS}*Stack* further reduces the computational cost by lowering the number of optimization variables. On the other hand, although the similarity coefficient in TC_{MS}*Stack* represents bivariate *S-T* similarity relations, they are elicited by pondering all the inter-domain dependencies. In the experiments, we show the superiority of TC_{MS}*Stack* on the transfer performance compared to TC_{SS}*Stack*, *GP-*TC_{MS}, and other *MSTR* methods.

2. Related Work

A main challenge in *MSTR* is to precisely capture the diverse *S-T* transfer capacities across the different sources. Ensemble approaches (Dai et al., 2007), which can provide an explicit, fine-grained similarity capture, are widely used to handle the *MSTR* problems. In (Pardoe & Stone, 2010), TrAdaBoost.R2 was proposed, a boosting based algorithm that weights the contribution of train instances individually, and thus delivers a model accounting for the *S-T* similarities for every instance. However, such boosting-like methods suffer from slow/premature convergence issues that tremendously jeopardize the transfer performance (Al-Stouhi & Reddy, 2011). Pardoe and Stone also introduced a multi-source transfer stacking in which base models are pretrained in different source domains separately, and a meta-model is trained by aggregating the outputs of the base models (Pardoe & Stone, 2010). By doing so, the *S-T* similarities are captured at meta-model level by the learnt model importance. Although the stacking methods show success in some *MSTR* problems, they have the limitation that inter-domain dependencies between sources are ignored at the base models.

At the other end of the spectrum are transfer covariance function representing a multivariate similarity relation over sources and target domains. A popular representative of this family is the work by Bonilla et al. (Bonilla et al., 2008) on multi-task learning, where a free-form kernel relates each pair of tasks. Apart from the difference of the application domain (multi-task learning versus *TL*), this kind of models often imply fitting an increasingly large number of hyperparameters; e.g. in the free-form kernel, this number grows as $(N^2 - N)/2$, where N is the number of sources. Motivated by (Bonilla et al., 2008), (Cao et al., 2010) develops another transfer covariance function for the single source transfer.

In this work, we first describe a family of transfer covariance functions, and investigate their feasibility for *MSTR*. With the theoretical analysis showing the unsatisfactory performance of such transfer covariance function, we propose to unify the *S-T* similarity capture of stacking and the transfer covariance function. To the best of our knowledge,

this is the first work that analyzes the feasibility and performance of such family of transfer covariance functions for *MSTR*, and further combines them with stacking.

3. Problem Statement

We denote a domain set for *MSTR* as $\mathcal{D} = \mathcal{S} \cup \mathcal{T}$ where $\mathcal{S} = \{\mathcal{S}_i : 1 \leq i \leq N\}$ is a set of source domains and $\mathcal{T}$ is the target domain. All source domain data and few target domain data are labeled. Denote the data matrix and its corresponding label vector in each source domain $\mathcal{S}_i$ as $\mathbf{X}^{(\mathcal{S}_i)} \in \mathbb{R}^{n_{\mathcal{S}_i} \times d}$ and $\mathbf{y}^{(\mathcal{S}_i)} \in \mathbb{R}^{n_{\mathcal{S}_i}}$. Likewise, we represent the target data set with $\mathbf{X}^{(\mathcal{T})} = \{\mathbf{X}^{(\mathcal{T}_l)}, \mathbf{X}^{(\mathcal{T}_u)}\}$ where $\mathbf{X}^{(\mathcal{T}_l)} \in \mathbb{R}^{n_{\mathcal{T}_l} \times d}$ is the labeled target data matrix and $\mathbf{X}^{(\mathcal{T}_u)} \in \mathbb{R}^{n_{\mathcal{T}_u} \times d}$ is the unlabeled one. We further define $\mathbf{y}^{(\mathcal{T}_l)} \in \mathbb{R}^{n_{\mathcal{T}_l}}$ as the label vector for $\mathbf{X}^{(\mathcal{T}_l)}$. Moreover, we assume $n_{\mathcal{T}_l} \ll min(n_{\mathcal{S}_1}, ..., n_{\mathcal{S}_N}, n_{\mathcal{T}_u})$. Our objective is to utilize $\{\mathbf{X}^{(\mathcal{S}_i)}, \mathbf{y}^{(\mathcal{S}_i)}\}_{i=1}^{N}$ and $\{\mathbf{X}^{(\mathcal{T}_l)}, \mathbf{y}^{(\mathcal{T}_l)}\}$ to predict labels for $\mathbf{X}^{(\mathcal{T}_u)}$.

We use the *GP* model for this regression task. We denote the underlying latent function between the inputs $\mathbf{x}$ and the outputs y as f, and the noise variance as σ^2. Thus, $\mathbf{f}$ denotes the function vector over $\mathbf{X}$. A *GP* model defines a Gaussian distribution over the functions, $\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$ in which $\boldsymbol{\mu}$ is the mean vector and $\mathbf{K}$ is the covariance matrix which is *positive semi-definite* (*PSD*, or equivalently denoted as $\mathbf{K} \succeq \mathbf{0}$). Usually $\boldsymbol{\mu}$ is assumed to be $\mathbf{0}$, and thus the *GP* model is completely specified by $\mathbf{K}$ given a covariance function which is parameterized by $\boldsymbol{\Omega}$.

4. *GP* with Transfer Covariance Function

In this section, we analyze the transfer performance of GP using a specific family of transfer covariance function.

4.1. Transfer Covariance Function for Multi-Source

Since the *GP* model is specified by $\mathbf{K}$, one straightforward way to achieve the knowledge transfer across multiple source domains and the target domain is to design a transfer covariance function for multi-source. Different from a classical *GP* which uses a fixed covariance function for the data from different domains, we focus on the covariance function of the form (*TC*$_{MS}$):

$$k_*(\mathbf{x}, \mathbf{x}') = \begin{cases} \lambda_i k(\mathbf{x}, \mathbf{x}'), \ \mathbf{x} \in \mathbf{X}^{(\mathcal{S}_i)} \ \& \ \mathbf{x}' \in \mathbf{X}^{(\mathcal{T})} \\ \qquad or \ \mathbf{x} \in \mathbf{X}^{(\mathcal{T})} \& \ \mathbf{x}' \in \mathbf{X}^{(\mathcal{S}_i)}, \\ k(\mathbf{x}, \mathbf{x}'), \quad otherwise. \end{cases} \tag{1}$$

where $k(\cdot, \cdot)$ is any valid covariance function, and λ_i is the metric measuring the similarity between the source $\mathcal{S}_i$ and the target $\mathcal{T}$. Through the learning, λ_i is expected to capture the different transfer strengths in different *S*-*T* domain pairs. Those highly target-related sources will play a more important role in transfer, while those completely target-unrelated sources will not be considered. However, to guarantee the *GP* model is always valid, any covariance matrix $\mathbf{K}_*$ constructed by $k_*(\cdot, \cdot)$ should be *PSD*. Theorem 1 gives the sufficient and necessary condition for a *PSD* $\mathbf{K}_*$.

Theorem 1. *Let* $\mathbf{K}_{\mathcal{D}_i\mathcal{D}_j}$ $(\mathcal{D}_i, \mathcal{D}_j \in \mathcal{D})$ *denote a covariance matrix for points in* $\mathcal{D}_i$ *and* $\mathcal{D}_j$*. A Gram matrix*

$$\mathbf{K}_* = \begin{bmatrix} \mathbf{K}_{\mathcal{S}_1\mathcal{S}_1} & ... & \mathbf{K}_{\mathcal{S}_1\mathcal{S}_N} & \lambda_1\mathbf{K}_{\mathcal{S}_1\mathcal{T}} \\ ... & ... & ... & ... \\ \mathbf{K}_{\mathcal{S}_N\mathcal{S}_1} & ... & \mathbf{K}_{\mathcal{S}_N\mathcal{S}_N} & \lambda_N\mathbf{K}_{\mathcal{S}_N\mathcal{T}} \\ \lambda_1\mathbf{K}_{\mathcal{T}\mathcal{S}_1} & ... & \lambda_N\mathbf{K}_{\mathcal{T}\mathcal{S}_N} & \mathbf{K}_{\mathcal{T}\mathcal{T}} \end{bmatrix}$$

is PSD for any covariance matrix $\mathbf{K}$ *in the form*

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathcal{S}_1\mathcal{S}_1} & ... & \mathbf{K}_{\mathcal{S}_1\mathcal{S}_N} & \mathbf{K}_{\mathcal{S}_1\mathcal{T}} \\ ... & ... & ... & ... \\ \mathbf{K}_{\mathcal{S}_N\mathcal{S}_1} & ... & \mathbf{K}_{\mathcal{S}_N\mathcal{S}_N} & \mathbf{K}_{\mathcal{S}_N\mathcal{T}} \\ \mathbf{K}_{\mathcal{T}\mathcal{S}_1} & ... & \mathbf{K}_{\mathcal{T}\mathcal{S}_N} & \mathbf{K}_{\mathcal{T}\mathcal{T}} \end{bmatrix}$$

if and only if $\lambda_1 = ... = \lambda_N$ *and* $|\lambda_i| \leq 1$.

Proof. Necessary condition: Let $\mathbf{K}_*$ be a *PSD* matrix. We use $\mathbf{K}_{\mathcal{S}\mathcal{S}}$ to represent the sources-to-sources block matrix, and $\mathbf{K}_{\mathcal{S}\mathcal{T}}$, $\mathbf{K}_{\mathcal{S}\mathcal{T}_*}$ to represent the sources-to-target block matrix in $\mathbf{K}$ and $\mathbf{K}_*$, respectively. Thus, we have:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathcal{S}\mathcal{S}} & \mathbf{K}_{\mathcal{S}\mathcal{T}} \\ \mathbf{K}_{\mathcal{S}\mathcal{T}}^{\mathsf{T}} & \mathbf{K}_{\mathcal{T}\mathcal{T}} \end{bmatrix}, \mathbf{K}_* = \begin{bmatrix} \mathbf{K}_{\mathcal{S}\mathcal{S}} & \mathbf{K}_{\mathcal{S}\mathcal{T}_*} \\ \mathbf{K}_{\mathcal{S}\mathcal{T}_*}^{\mathsf{T}} & \mathbf{K}_{\mathcal{T}\mathcal{T}} \end{bmatrix}.$$

Since $\mathbf{K}$ is *PSD*, according to the Schur complement theorem (Zhang, 2006), we have:

$$(\mathbf{I} - \mathbf{K}_{\mathcal{S}\mathcal{S}}\widetilde{\mathbf{K}}_{\mathcal{S}\mathcal{S}})\mathbf{K}_{\mathcal{S}\mathcal{T}} = \mathbf{0}, \tag{2}$$

$$\mathbf{K}_{\mathcal{T}\mathcal{T}} - \mathbf{K}_{\mathcal{S}\mathcal{T}}^{\mathsf{T}}\widetilde{\mathbf{K}}_{\mathcal{S}\mathcal{S}}\mathbf{K}_{\mathcal{S}\mathcal{T}} \succeq \mathbf{0}, \tag{3}$$

where $\widetilde{\mathbf{K}}_{\mathcal{S}\mathcal{S}}$ is the generalized inverse of $\mathbf{K}_{\mathcal{S}\mathcal{S}}$. By rewriting $\widetilde{\mathbf{K}}_{\mathcal{S}\mathcal{S}}$ as a block matrix using $\widetilde{\mathbf{K}}_{\mathcal{S}_i\mathcal{S}_j}$ as the element, we further derive eq. (2) and eq. (3) as:

$$\begin{bmatrix} \mathbf{K}_{\mathcal{S}_1\mathcal{T}} - \sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{K}_{\mathcal{S}_1\mathcal{S}_i}\widetilde{\mathbf{K}}_{\mathcal{S}_i\mathcal{S}_j}\mathbf{K}_{\mathcal{S}_j\mathcal{T}} \\ ... \\ \mathbf{K}_{\mathcal{S}_N\mathcal{T}} - \sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{K}_{\mathcal{S}_N\mathcal{S}_i}\widetilde{\mathbf{K}}_{\mathcal{S}_i\mathcal{S}_j}\mathbf{K}_{\mathcal{S}_j\mathcal{T}} \end{bmatrix} = \mathbf{0}, \tag{4}$$

$$\mathbf{K}_{\mathcal{T}\mathcal{T}} - \sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{K}_{\mathcal{T}\mathcal{S}_i}\widetilde{\mathbf{K}}_{\mathcal{S}_i\mathcal{S}_j}\mathbf{K}_{\mathcal{S}_j\mathcal{T}} \succeq \mathbf{0}. \tag{5}$$

Likewise, for the *PSD* matrix $\mathbf{K}_*$, we have the following two Schur complement derivations:

$$\begin{bmatrix} \lambda_1\mathbf{K}_{\mathcal{S}_1\mathcal{T}} - \sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_j\mathbf{K}_{\mathcal{S}_1\mathcal{S}_i}\widetilde{\mathbf{K}}_{\mathcal{S}_i\mathcal{S}_j}\mathbf{K}_{\mathcal{S}_j\mathcal{T}} \\ ... \\ \lambda_N\mathbf{K}_{\mathcal{S}_N\mathcal{T}} - \sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_j\mathbf{K}_{\mathcal{S}_N\mathcal{S}_i}\widetilde{\mathbf{K}}_{\mathcal{S}_i\mathcal{S}_j}\mathbf{K}_{\mathcal{S}_j\mathcal{T}} \end{bmatrix} = \mathbf{0}. \tag{6}$$

$$\mathbf{K}_{\mathcal{T}\mathcal{T}} - \sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j\mathbf{K}_{\mathcal{T}\mathcal{S}_i}\widetilde{\mathbf{K}}_{\mathcal{S}_i\mathcal{S}_j}\mathbf{K}_{\mathcal{S}_j\mathcal{T}} \succeq \mathbf{0}. \quad (7)$$

Combining eq. (4) and eq. (6), we get:

$$\begin{bmatrix} \sum_{i=1}^{N}\sum_{j=1}^{N}(\lambda_1-\lambda_j)\mathbf{K}_{\mathcal{S}_1\mathcal{S}_i}\widetilde{\mathbf{K}}_{\mathcal{S}_i\mathcal{S}_j}\mathbf{K}_{\mathcal{S}_j\mathcal{T}} \\ ... \\ \sum_{i=1}^{N}\sum_{j=1}^{N}(\lambda_N-\lambda_j)\mathbf{K}_{\mathcal{S}_N\mathcal{S}_i}\widetilde{\mathbf{K}}_{\mathcal{S}_i\mathcal{S}_j}\mathbf{K}_{\mathcal{S}_j\mathcal{T}} \end{bmatrix} = \mathbf{0}. \quad (8)$$

Since Eq. (8) must hold for all *PSD* $\mathbf{K}$, we induce $\lambda_1 = ... = \lambda_N = \lambda$. Based on such conclusion, we combine eq. (5) and eq. (7):

$$(1-\lambda^2)\mathbf{K}_{\mathcal{T}\mathcal{T}} + \lambda^2\mathbf{M} \succeq \mathbf{0}, \quad (9)$$

where $\mathbf{M} = \mathbf{K}_{\mathcal{T}\mathcal{T}} - \sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{K}_{\mathcal{T}\mathcal{S}_i}\widetilde{\mathbf{K}}_{\mathcal{S}_i\mathcal{S}_j}\mathbf{K}_{\mathcal{S}_j\mathcal{T}}$. Since eq. (9) must hold for all *PSD* $\mathbf{K}_{\mathcal{T}\mathcal{T}}$ and *PSD* $\mathbf{M}$, we resolve that $|\lambda| \leq 1$.

Sufficient condition: Let $\lambda_1 = ... = \lambda_N = \lambda$, and $|\lambda| \leq 1$. According to the Theorem 1 in (Cao et al., 2010), we obtain that $\mathbf{K}_*$ is a *PSD* matrix.

To sum up, we conclude that if $\mathbf{K}_*$ is a *PSD* matrix, λ_i should satisfy $\lambda_1 = ... = \lambda_N$ and $|\lambda_i| \leq 1$. □

From Theorem 1, we can see that $|\lambda_i| \leq 1$, which means a highly target-related source results in a full transfer of $\mathbf{K}_{\mathcal{S}_i,\mathcal{T}}$, but a completely target-unrelated source results in a zero block matrix. This indicates the adaptiveness of λ_i. However, Theorem 1 also shows that $k_*(\cdot,\cdot)$ can just give one similarity coefficient for all *S-T* domain pairs to ensure the validity of *GP-TC*$_{MS}$. Such single similarity coefficient compromises the diverse similarities between different *S-T* domain pairs. This violates the original intention of λ_i which is to distinguish the similarity diversity between different *S-T* domain pairs.

4.2. Generalization Bounds of *GP-TC*$_{MS}$

To investigate the effect of a single compromised similarity coefficient on the performance of *GP-TC*$_{MS}$, we derive its generalization error bounds. In (Chai, 2009), an earlier analysis can be found for the generalization errors and learning curves in multi-task learning (specifically, two learning tasks with the same noise variance). Our investigation is different from that work however as we are working on a *TL* setting, and more importantly, on multiple sources with different noise variances.

We denote the single compromised similarity coefficient as λ, and the noise variance for different domains as $\sigma_d^2,\ d \in \{\mathcal{S}_1, ..., \mathcal{S}_N, \mathcal{T}\}$. Thus, the transfer covariance matrix of the noisy training data is $\mathbf{C}_* = \mathbf{K}_* + \mathbf{\Sigma}$ where

$$\mathbf{K}_* = \begin{bmatrix} \mathbf{K}_{\mathcal{S}\mathcal{S}} & \lambda\mathbf{K}_{\mathcal{S}\mathcal{T}} \\ \lambda\mathbf{K}_{\mathcal{S}\mathcal{T}}^\mathsf{T} & \mathbf{K}_{\mathcal{T}\mathcal{T}} \end{bmatrix}, \mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{\mathcal{S}} & \mathbf{0} \\ \mathbf{0} & \sigma_{\mathcal{T}}^2\mathbf{I}_{\mathcal{T}\mathcal{T}} \end{bmatrix}$$

and $\mathbf{\Sigma}_{\mathcal{S}}$ is a diagonal block matrix with the diagonal block elements $\{\sigma_{\mathcal{S}_1}^2\mathbf{I}_{\mathcal{S}_1\mathcal{S}_1}, ..., \sigma_{\mathcal{S}_N}^2\mathbf{I}_{\mathcal{S}_N\mathcal{S}_N}\}$. According to (Rasmussen, 2006), if the GP prior is correctly specified, the generalization error at a point is also the posterior variance at such point. Specifically, for *GP-TC*$_{MS}$, the posterior variance at the target point $\mathbf{x}_t$ is:

$$\begin{aligned} &\delta_{\mathcal{T}}^2(\mathbf{x}_t, \lambda, \{\mathbf{X}^{(\mathcal{S}_i)}, \sigma_{\mathcal{S}_i}^2\}_{i=1}^N, \mathbf{X}_{\mathcal{T}}, \sigma_{\mathcal{T}}^2) \\ &= k_{tt} - \mathbf{k}_{*t}^\mathsf{T}\mathbf{C}_*^{-1}\mathbf{k}_{*t}, \end{aligned} \quad (10)$$

where $\mathbf{k}_{*t}^\mathsf{T} = (\lambda\mathbf{k}_{\mathcal{S}t}^\mathsf{T}, \mathbf{k}_{\mathcal{T}t}^\mathsf{T})$, $\mathbf{k}_{\mathcal{S}t}$ ($\mathbf{k}_{\mathcal{T}t}$) is the vector of covariances between $\{\mathbf{X}^{(\mathcal{S}_i)}\}_{i=1}^N$ ($\mathbf{X}^{(\mathcal{T})}$) and $\mathbf{x}_t$, and k_{tt} is the prior variance at $\mathbf{x}_t$. Wherever it is not misleading, we will simplify the posterior variance expression using $\delta_{\mathcal{T}}^2(\lambda, \{\sigma_{\mathcal{S}_i}^2\}_{i=1}^N, \sigma_{\mathcal{T}}^2)$. The generalization error for the target domain can be obtained by averaging eq. (10) over $\mathbf{x}_t$:

$$\begin{aligned} &\epsilon_{\mathcal{T}}(\lambda, \{\mathbf{X}^{(\mathcal{S}_i)}, \sigma_{\mathcal{S}_i}^2\}_{i=1}^N, \mathbf{X}_{\mathcal{T}}, \sigma_{\mathcal{T}}^2) \\ &= \int \delta_{\mathcal{T}}^2(\mathbf{x}_t, \lambda, \{\sigma_{\mathcal{S}_i}^2\}_{i=1}^N, \sigma_{\mathcal{T}}^2)p(\mathbf{x}_t)d\mathbf{x}_t. \end{aligned} \quad (11)$$

To derive the generalization error bounds for *GP-TC*$_{MS}$, we first rewrite

$$\mathbf{C}_* = \mathbf{\Lambda}\begin{bmatrix} \lambda^{-2}(\mathbf{K}_{\mathcal{S}\mathcal{S}} + \mathbf{\Sigma}_{\mathcal{S}}) & \mathbf{K}_{\mathcal{S}\mathcal{T}} \\ \mathbf{K}_{\mathcal{S}\mathcal{T}}^\mathsf{T} & \mathbf{K}_{\mathcal{T}\mathcal{T}} + \sigma_{\mathcal{T}}^2\mathbf{I}_{\mathcal{T}\mathcal{T}} \end{bmatrix}\mathbf{\Lambda},$$

where $\mathbf{\Lambda} = \begin{bmatrix} \lambda\mathbf{I}_{\mathcal{S}\mathcal{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{\mathcal{T}\mathcal{T}} \end{bmatrix}$. Thus, the posterior variance at point $\mathbf{x}_t$ becomes:

$$\delta_{\mathcal{T}}^2(\lambda, \{\sigma_{\mathcal{S}_i}^2\}_{i=1}^N, \sigma_{\mathcal{T}}^2) = k_{tt} - \mathbf{k}_t^\mathsf{T}\Phi(\lambda, \{\sigma_{\mathcal{S}_i}^2\}_{i=1}^N, \sigma_{\mathcal{T}}^2)^{-1}\mathbf{k}_t, \quad (12)$$

where $\mathbf{k}_t^\mathsf{T} = (\mathbf{k}_{\mathcal{S}t}^\mathsf{T}, \mathbf{k}_{\mathcal{T}t}^\mathsf{T})$ and

$$\begin{aligned} &\Phi(\lambda, \{\sigma_{\mathcal{S}_i}^2\}_{i=1}^N, \sigma_{\mathcal{T}}^2) \\ &= \begin{bmatrix} \lambda^{-2}(\mathbf{K}_{\mathcal{S}\mathcal{S}} + \mathbf{\Sigma}_{\mathcal{S}}) & \mathbf{K}_{\mathcal{S}\mathcal{T}} \\ \mathbf{K}_{\mathcal{S}\mathcal{T}}^\mathsf{T} & \mathbf{K}_{\mathcal{T}\mathcal{T}} + \sigma_{\mathcal{T}}^2\mathbf{I}_{\mathcal{T}\mathcal{T}} \end{bmatrix}. \end{aligned}$$

Note that the above derivation excludes the situation where $\lambda = 0$. When $\lambda = 0$, all the source domains are unrelated to the target domain, and thus no knowledge is transferred. This is easy to verify by plugging $\lambda = 0$ into eq. (12).

Further, we observe that $\delta_{\mathcal{T}}^2$ is equal for λ and $-\lambda$, so we only investigate the case $\lambda \in (0, 1]$. For eq. (12), we further decompose it as:

$$\begin{aligned} &\Phi(\lambda, \{\sigma_{\mathcal{S}_i}^2\}_{i=1}^N, \sigma_{\mathcal{T}}^2) \\ &= \begin{bmatrix} \mathbf{K}_{\mathcal{S}\mathcal{S}} & \mathbf{K}_{\mathcal{S}\mathcal{T}} \\ \mathbf{K}_{\mathcal{S}\mathcal{T}}^\mathsf{T} & \mathbf{K}_{\mathcal{T}\mathcal{T}} \end{bmatrix} + \begin{bmatrix} \sigma_{\mathcal{T}}^2\mathbf{I}_{\mathcal{S}\mathcal{S}} & \mathbf{0} \\ \mathbf{0} & \sigma_{\mathcal{T}}^2\mathbf{I}_{\mathcal{T}\mathcal{T}} \end{bmatrix} + \\ &(\lambda^{-2}-1)\begin{bmatrix} \mathbf{K}_{\mathcal{S}\mathcal{S}} + \mathbf{\Sigma}_{\mathcal{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{\Sigma}_{\mathcal{S}} - \sigma_{\mathcal{T}}^2\mathbf{I}_{\mathcal{S}\mathcal{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \Phi(1, \{\sigma_{\mathcal{T}}^2, ..., \sigma_{\mathcal{T}}^2\}_N, \sigma_{\mathcal{T}}^2) + \mathbf{E}_1 + \mathbf{E}_2, \end{aligned} \quad (13)$$

where $\mathbf{E}_1 = (\lambda^{-2}-1)\begin{bmatrix}\mathbf{K}_{\mathcal{SS}}+\mathbf{\Sigma}_{\mathcal{S}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}$ and $\mathbf{E}_2 = \begin{bmatrix}\mathbf{\Sigma}_{\mathcal{S}}-\sigma_{\mathcal{T}}^2\mathbf{I}_{\mathcal{SS}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}$. Eq. (13) unveils that the posterior variance of having instances from different source domains is equivalent to the posterior variance of having those instances from target domain with two additional correlated noise terms, $\mathbf{E}_1$ and $\mathbf{E}_2$. This shows us the two main factors that matter in the transfer performance; namely, the *S-T* similarity and the noise variances. To further analyze how the *S-T* similarity affects the transfer performance, we focus on one factor and fix the other. Assuming that all the sources are totally related to the target, i.e. $\lambda = 1$, and consequently, the noise variance for each source becomes $\xi^2_{\mathcal{S}_i}$, we define the difference:

$$\Delta = \delta^2_{\mathcal{T}}(1, \{\xi^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}}) - \delta^2_{\mathcal{T}}(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}}).$$

To obtain the upper (lower) bound of $\delta^2_{\mathcal{T}}(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})$, we are interested in those $\overline{\xi}^2_{\mathcal{S}_i}$ ($\underline{\xi}^2_{\mathcal{S}_i}$) that make $\Delta \geq 0$ ($\Delta < 0$) for all the target points.

Proposition 1. *Let $\overline{\delta}$ and $\underline{\delta}$ be the maximum and minimum eigenvalues of $\mathbf{K}_{\mathcal{SS}}$, $\overline{\xi}^2_{\mathcal{S}_i} = \lambda^{-2}\sigma^2_{\mathcal{S}_i} - (1-\lambda^{-2})\overline{\delta}$ and $\underline{\xi}^2_{\mathcal{S}_i} = \lambda^{-2}\sigma^2_{\mathcal{S}_i} - (1-\lambda^{-2})\underline{\delta}$ for every source $\mathcal{S}_i$. Then, for all the target data points, $\delta^2_{\mathcal{T}}(1, \{\underline{\xi}^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}}) \leq \delta^2_{\mathcal{T}}(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}}) \leq \delta^2_{\mathcal{T}}(1, \{\overline{\xi}^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})$.*

Proof. By applying eq. (12), we have:

$$\begin{aligned}\Delta &= \delta^2_{\mathcal{T}}(1, \{\xi^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}}) - \delta^2_{\mathcal{T}}(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})\\ &= \mathbf{k}_t^\top[\Phi(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})^{-1} - \Phi(1, \{\xi^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})^{-1}]\mathbf{k}_t\end{aligned}$$

To make $\Delta \geq 0$ for all the target data points, we need to prove $\Phi(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})^{-1} - \Phi(1, \{\xi^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})^{-1}$ is PSD, which means:

$$\begin{aligned}&\Phi(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})^{-1} - \Phi(1, \{\xi^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})^{-1} \succeq \mathbf{0}\\ \iff\quad &\Phi(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}}) \preceq \Phi(1, \{\xi^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})\\ \iff\quad &\begin{bmatrix}(1-\lambda^{-2})\mathbf{K}_{\mathcal{SS}} + (\mathbf{\Sigma}'_{\mathcal{S}} - \lambda^{-2}\mathbf{\Sigma}_{\mathcal{S}}) & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix} \succeq \mathbf{0}\end{aligned}$$

where $\mathbf{\Sigma}'_{\mathcal{S}}$ is a diagonal block matrix with the diagonal block elements $\{\xi^2_{\mathcal{S}_1}\mathbf{I}_{\mathcal{S}_1\mathcal{S}_1}, ..., \xi^2_{\mathcal{S}_N}\mathbf{I}_{\mathcal{S}_N\mathcal{S}_N}\}$

$$\begin{aligned}\iff\quad &(1-\lambda^{-2})\mathbf{K}_{\mathcal{SS}} + (\mathbf{\Sigma}'_{\mathcal{S}} - \lambda^{-2}\mathbf{\Sigma}_{\mathcal{S}}) \succeq \mathbf{0}\\ \iff\quad &\mathbf{K}_{\mathcal{SS}} \preceq \frac{(\lambda^{-2}\mathbf{\Sigma}_{\mathcal{S}} - \mathbf{\Sigma}'_{\mathcal{S}})}{(1-\lambda^{-2})}\\ \iff\quad &\overline{\delta} \leq \frac{(\lambda^{-2}\sigma^2_{\mathcal{S}_i} - \xi^2_{\mathcal{S}_i})}{(1-\lambda^{-2})} \;\; for\ every\ \mathcal{S}_i\\ \iff\quad &\xi^2_{\mathcal{S}_i} \geq \lambda^{-2}\sigma^2_{\mathcal{S}_i} - (1-\lambda^{-2})\overline{\delta} \;\; for\ every\ \mathcal{S}_i\end{aligned}$$

Note that Δ is a monotonically increasing function of $\xi^2_{\mathcal{S}_i}$, thus we take the minimum $\lambda^{-2}\sigma^2_{\mathcal{S}_i} - (1-\lambda^{-2})\overline{\delta}$ as $\overline{\xi}^2_{\mathcal{S}_i}$ to be the smallest upper bound of $\sigma^2_{\mathcal{T}}(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})$. Similarly, we have $\underline{\xi}^2_{\mathcal{S}_i} = \lambda^{-2}\sigma^2_{\mathcal{S}_i} - (1-\lambda^{-2})\underline{\delta}$ to construct the largest lower bound of $\sigma^2_{\mathcal{T}}(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})$. □

Proposition 1 gives the lower and upper bounds of the posterior variance $\delta^2_{\mathcal{T}}(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})$. By applying eq. (11), we readily obtain the generalization error bounds.

Corollary 1. *Let*

$$\begin{aligned}\overline{\epsilon}_{\mathcal{T}}(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}}) &= \epsilon_{\mathcal{T}}(1, \{\overline{\xi}^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})\\ \underline{\epsilon}_{\mathcal{T}}(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}}) &= \epsilon_{\mathcal{T}}(1, \{\underline{\xi}^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})\end{aligned}$$

Then, $\underline{\epsilon}_{\mathcal{T}}(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}}) \leq \epsilon_{\mathcal{T}}(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}}) \leq \overline{\epsilon}_{\mathcal{T}}(\lambda, \{\sigma^2_{\mathcal{S}_i}\}_{i=1}^N, \sigma^2_{\mathcal{T}})$.

Proposition 1 serves to demonstrate that λ takes effect in every source on the final transfer performance. With the assumption that different source domains have different *S-T* similarities with the target domain, a single λ that works for every sources has a great difficulty capturing such *S-T* similarity diversity. This leads us to exploit the transfer covariance function in another way.

5. Transfer Covariance Function Stacking

Considering the effectiveness showed by the stacking strategy for *MSTR* (Pardoe & Stone, 2010), we propose a framework that can integrate the capability for *S-T* similarity capture of both the transfer covariance function and stacking. We first introduce *TC$_{SS}$Stack*, a conventional way of stacking the transfer covariance function. Then, we design a more involved stacking-inspired approach that overcomes some limitations of the conventional stacking method.

5.1. Conventional Transfer Stacking *TC$_{SS}$Stack*

Motivated by the fact that both the stacking strategy and the transfer covariance function can model the *S-T* similarity and using the transfer covariance function at the base models would therefore add flexibility to the similarity capture capability of the stacking approach, we propose a *TC$_{SS}$Stack* method. In *TC$_{SS}$Stack*, we first train multiple *GP-TC$_{SS}$* models using each $\mathcal{S}_i$ and $\mathcal{T}$ (denoted as $\{f^{(\mathcal{S}_i,\mathcal{T})}(\cdot|\mathbf{\Omega}_i, \lambda_i)\}_{i=1}^N$) and then apply the conventional stacking strategy to combine their predictions. Given a target point $\mathbf{x}$, the final prediction is given by:

$$f(\mathbf{x}) = \sum_{i=1}^N \omega_i f^{(\mathcal{S}_i,\mathcal{T})}(\mathbf{x}|\mathbf{\Omega}_i, \lambda_i),\ \sum_{i=1}^N \omega_i = 1 \quad (14)$$

where ω_i are coefficients learned by minimizing the least square error on the target labeled data.

There are two major issues for the above model. (1) Since each $f^{(\mathcal{S}_i,\mathcal{T})}$ is pretrained separately, the parameters learnt for each $f^{(\mathcal{S}_i,\mathcal{T})}$ do not take the inter-domain dependencies between different source domains into account. (2) Both λ_i and ω_i reflect the *S-T* domain similarity. However, *TC*$_{SS}$*Stack* takes them as two different variables and learns them separately. Intuitively, the model importance ω_i should be positively correlated with the similarity coefficient λ_i. For example, the prediction of a *GP-TC*$_{SS}$ using an unrelated source is less trustful, and should be assigned a smaller coefficient in the stacking.

5.2. Improved Transfer Stacking *TC*$_{MS}$*Stack*

To overcome the above issues, we propose a new transfer stacking model (*TC*$_{MS}$*Stack*) as follows:

$$f^*(\mathbf{x}) = \sum\nolimits_{i=1}^{N} (g(\lambda_i)/Z) f^{(\mathcal{S}_i,\mathcal{T})}(\mathbf{x}, \mathbf{\Omega}_i, \lambda_i). \qquad (15)$$

where λ_i refers to the similarity coefficient in the *GP-TC*$_{SS}$ for the *i-th* source, $Z = \sum_{i=1}^{N} g(\lambda_i)$ is the normalization term, and $g(\lambda_i)$ is any function preserving the monotonicity of $|\lambda_i|$ so that it coordinates the model importance and the similarity coefficient. This also reduces the search efforts by lowering the number of free parameters to fit. Moreover, instead of pretraining $f^{(\mathcal{S}_i,\mathcal{T})}(\cdot|\mathbf{\Omega}_i, \lambda_i)$ separately, we jointly learn $f^{(\mathcal{S}_i,\mathcal{T})}(\cdot, \mathbf{\Omega}_i, \lambda_i)$ for all the source domains. By doing so, the multiple *GP-TC*$_{SS}$ models are learned together with the dependencies between multiple sources taken into account.

Notice that the model in eq. (15) allows for multiple options to characterize the relative importance of *GP-TC*$_{SS}$ models through $g(\cdot)$. In this paper, we use a simple function $g(\lambda_i) = |\lambda_i|$. However, the absolute value function is not smooth at the origin. Thus, we use a smooth function studied in (Yong, 2015) to approximate it as follows:

$$|\lambda_i| \approx \alpha Ln(\frac{1}{2}e^{\frac{\lambda_i}{\alpha}} + \frac{1}{2}e^{-\frac{\lambda_i}{\alpha}}).$$

We set $\alpha = 0.01$ which is the best approximation stated in (Yong, 2015). Since Theorem 1 also tells us $-1 \leq \lambda_i \leq 1$, we propose to define $\lambda_i = 2(1/(1+\mu_i))^{b_i} - 1$ ($\mu_i \geq 0$ and $b_i \geq 0$), as in (Cao et al., 2010). Then, we conduct the learning by minimizing the squared errors:

$$\min_{\{\mathbf{\Omega}_i,\mu_i,b_i\}_{i=1}^{N}} \sum\nolimits_{j=1}^{n_{\mathcal{T}_l}} \left(y_j^{(\mathcal{T}_l)} - f^*(\mathbf{x}_j^{(\mathcal{T}_l)})\right)^2. \qquad (16)$$

In the optimization, we propose to use the conjugate gradient method. Other optimization methods can also be applied to solve this objective function.

5.3. Complexity Analysis

As in usual *GP* model training, the computational time complexity of each $f^{(\mathcal{S}_i,\mathcal{T})}$ is dominated by the calculation of the inverse of its covariance matrix, i.e. $\mathcal{O}((n_{\mathcal{T}_l}+n_{\mathcal{S}_i})^3)$. Considering $n_{\mathcal{T}_l} \ll n_{\mathcal{S}_i}$ and assuming $n_{\mathcal{S}_1} = ... = n_{\mathcal{S}_N} = n_{\mathcal{S}}$, the evaluation of a *TC*$_{MS}$*Stack* model takes then $\mathcal{O}(Nn_{\mathcal{S}}^3)$. Notice that by following the stacking strategy of eq. (14) the training involves the steps of learning each $f^{(\mathcal{S}_i,\mathcal{T})}$ and subsequently learning the ω_i coefficients. The latter calls for some cross-validation approach to evaluate a meta-model, as the $f^{(\mathcal{S}_i,\mathcal{T})}$ from the previous step have been induced using the target data (Pardoe & Stone, 2010). In the extreme case of leave-one-out, this would take $\mathcal{O}(n_{\mathcal{T}_l}Nn_{\mathcal{S}}^3)$. Even if we also choose a leave-one-out validation to solve eq. (16) the cost of a *TC*$_{MS}$*Stack* model evaluation would be lower, since the first step of stacking is not required. On the other side, by following TrAdaBoost.R2 or the *GP-TC*$_{MS}$ approach, a *GP* model evaluation requires $\mathcal{O}((Nn_{\mathcal{S}})^3)$, which even exceeds the cost for *TC*$_{MS}$*Stack* using leave-one-out whenever $N > \sqrt{n_{\mathcal{T}_l}}$.

Table 1. Amazon review products dataset 15.

Top category	Source domains	Target domain
Beauty_Health_Grocery	Beauty, Grocery_Food, Health	Clothing_Shoes_Jewelry
Electronics	Electronic, Office_Product, Kindle_Store	Cellphone_Accessory
Home_Garden_Tool	Kitchen, Pet_Supplies	Tools_Home_Improvements
Movies_Music_Game	CD_Vinyl, Digital_Music, Video_Games	Movies_TV

6. Experimental Study

In the following experimental study we aim at two main goals: (i) to assess the ability of *TC*$_{MS}$*Stack* in capturing inter-domain similarity, and (ii) to evaluate its predictive effectiveness compared to other approaches.

6.1. Experiment Setting

All the GPs herein build upon a standard squared exponential covariance function. The hyperparameters of each method are optimized using the conjugate gradient implementation from the *gpml* package (Rasmussen & Nickisch, 2010). For each search, we allow a maximum of 200 evaluations. The reported results correspond to the model providing the best objective function value over 5 independent runs with random initial solutions each. We use one synthetic dataset and two real-world datasets.

Synthetic dataset. We consider a linear function $f(\mathbf{x}) = \mathbf{w}_0^{\mathrm{T}}\mathbf{x} + \epsilon$, where $\mathbf{w}_0 \in \mathbb{R}^{100}$ and ϵ is a zero-mean Gaussian noise term, as the target. We use this function to generate 100 points as target test data, and 20 points as target train data. For the source task, we use $g(\mathbf{x}) = (\mathbf{w}_0^{\mathrm{T}} + \delta\Delta\mathbf{w})\mathbf{x} + \epsilon$, where $\Delta\mathbf{w}$ is a random fluctuation vector and δ is the variable controlling the similarity between f and g (higher δ indicates lower similarity), to generate 380 points for each source with different δ.

Amazon reviews. We extract the raw data containing 15

product reviews from (McAuley et al., 2015), and categorize the products into four top categories according to the Amazon website. Products in the same category are conceptually similar. Each product is taken as a domain, and we select one as target from each category (see Table 1). Reviews in each domain are represented by the count features and are labeled using stars in the set $\{1, 2, 3, 4, 5\}$.

UJIIndoorLoc. The building location dataset covers three buildings of Universitat Jaume I with four floors each (Torres-Sospedra et al., 2014). We build 12 domains by taking the location data from each floor of each building as a domain. The first floor of each building is taken as the target. Domains from the same building are taken as similar. The received signal strength intensity from 520 wireless access points is used as the features, and the location represented by the latitude and longitude is taken as label.

6.2. Domain Similarity Capture

We first elucidate the ability of *TC$_{MS}$Stack* in capturing the diverse *S-T* similarities through the λ_i coefficients. To rationalize the assessment, we use the synthetic dataset and consider a variety of problems covering a broad spectrum of TL settings. Precisely, we build four scenarios of $N = 2, 5, 10, 15$ sources. In each scenario, we specify six problems, each given by a different combination of sources. Three problems represent settings in which all the sources are equally similar to the target, with high ($\delta = 0$), medium ($\delta = 15$) and low ($\delta = 35$) similarity strength. The other three problems reflect diverse *S-T* similarities. Each source is given by a δ randomly sampled from the set $\{0, 4, 7, 10, 15, 20, 25, 30, 35\}$ and with replacement. We enforce the three problems to be different and avoid all the sources to be equal. We show the results in Figure 1.

In the figure, the first three problems of each scenario are the cases with equal *S-T* similarities. It can be observed in the bar plots on the left hand side that the λ values learnt by *TC$_{MS}$Stack* are strictly reverse-correlated with the predefined δ values, which indicates an accurate capture of the high, medium and low strengths of *S-T* similarity. We further observe from the black dots that *GP-TC$_{MS}$* is also able to strongly coordinate δ with a single compromised λ. This is because all the sources share the same δ with the target, and thus can be regarded as a single larger source.

The remaining three problems of each scenario reflect diverse similarities across source domains. In this case, Figure 1 shows that the λ values of *TC$_{MS}$Stack* reflect the relative differences of δ across different sources fairly well in general. The learnt λ is generally reverse-correlated with the predefined δ values, but it is not strict and tight. For instance, in problem 6 of the 5-sources scenario or in the problem 5 of the 15-sources scenario, such reverse-correlated relations do not hold in all the sources. This is

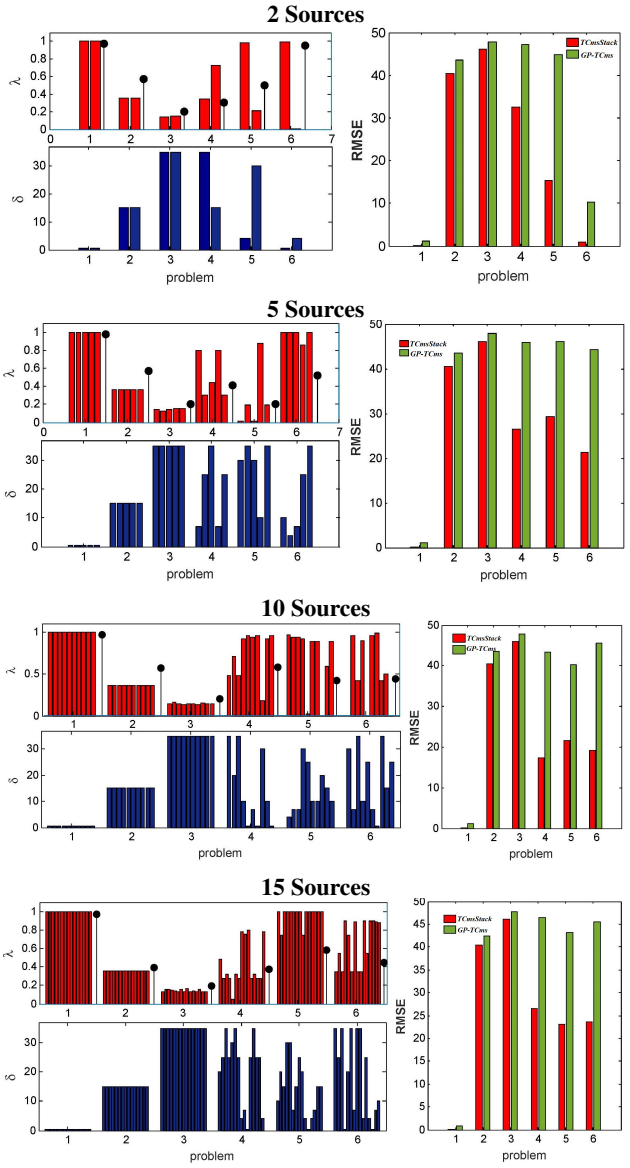

Figure 1. Results on six problem settings for each scenario of 2, 5, 10 and 15 sources. Bar plots on the left show the correspondence between each problem similarity (δ) and the similarity captured by the model (λ); bars for *TC$_{MS}$Stack* and black dots for *GP-TC$_{MS}$*. Bar plots on the right show the RMSE for *TC$_{MS}$Stack* (red) and *GP-TC$_{MS}$* (green).

because, although the learnt λs only represent the bivariate *S-T* similarities, each of them is specified during learning by considering its influence relative to the rest of similarity coefficients, i.e. the inter-domain dependencies between different sources are taken into account during the learning of the λs. Thus, in some cases, the learnt λs may not strictly approximate the real *S-T* similarities. However, λs are always learnt to guarantee the outcome of the best transfer performance. That is the reason why the above two cases

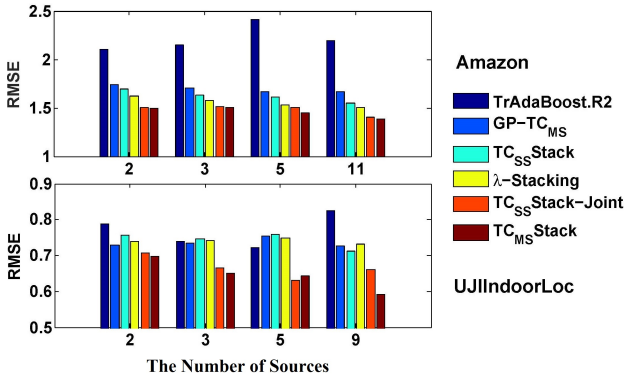

Figure 2. Comparison results on the two datasets.

still achieve satisfactory transfer performance in terms of RMSE. By contrast, we find that *GP-TC*$_{MS}$ only gives a trade-off value of λ over the diverse δ values. The right hand side of the figure shows a consistently lower RMSE for *TC*$_{MS}$*Stack* than for *GP-TC*$_{MS}$ in all the problems. In particular, a dramatic improvement is observed by utilizing *TC*$_{MS}$*Stack* for the diverse problems 4-6. These results indicate the superiority of *TC*$_{MS}$*Stack* over *GP-TC*$_{MS}$ for *MSTR*. We further verify this conclusion on the two real-world datasets in the next section.

6.3. Performance on Real-World Datasets

We compare *TC*$_{MS}$*Stack* with several *MSTR* approaches, namely: Tradaboost.R2, *GP-TC*$_{MS}$, *TC*$_{SS}$*Stack*, a variant of *TC*$_{SS}$*Stack* with joint learning of model importance coefficient and λ which we call *TC*$_{SS}$*Stack-Joint*, and a variant of *TC*$_{SS}$*Stack* using the learnt λ as the model importance which we call λ-Stacking. The evaluation comprises both the Amazon and the UJIIndoorLoc datasets. For each source domain we sample 500 points uniformly at random for training. Likewise, train and test data from each target domain are obtained by sampling 25 points and 1000 points, respectively. For the Amazon dataset, we generated a set of problems by using each target domain in Table 1, and by randomly choosing a number of source domains subsets. More precisely, for each scenario of 2, 3 and 5 sources, ten different source combinations were randomly constructed. In addition, a scenario of 11 sources with all the source domains was selected. Thus, we construct 40 transfer problems for each scenario of 2, 3 and 5 sources, and 4 problems for the scenario of 11 sources. For the UJIIndoorLoc dataset, we generate the transfer problems in a similar way to the Amazon dataset described above.

In Figure 2, we show the average RMSE results over all the problems in each scenario for the two datasets. Overall, *TC*$_{MS}$*Stack* is the winner among all the baselines on the two datasets, improving the transfer performance across the different amounts of source domains. This showcases the capability of *TC*$_{MS}$*Stack* to transfer knowledge from various sources with different *S-T* similarities.

For the other baselines, we observe that TrAdaBoost.R2 gives the poorest results due to the premature weights convergence issue. If, in addition, we consider the high computational cost for the models involved, TrAdaBoost.R2 does not seem to be a good choice for *MSTR*, especially when the number of source domains is large. As for *GP-TC*$_{MS}$, it presents a steadily inferior performance than *TC*$_{MS}$*Stack*. Overall, the outcomes are in line with those in the synthetic dataset, offering further support to the superiority of *TC*$_{MS}$*Stack* to *GP-TC*$_{MS}$. Notice that, since the current benchmark was generated randomly, it is likely a scenario to comprise diverse problem settings. Therefore, capturing the diverse similarities through a single λ coefficient may compromise the performance of *GP-TC*$_{MS}$. As opposed to *GP-TC*$_{MS}$, *TC*$_{MS}$*Stack* offers more robust performance improvements.

Finally, the comparison with the other stacking-based methods exposes the benefits of the two salient features of *TC*$_{MS}$*Stack*. Both *TC*$_{SS}$*Stack* and λ-Stacking are beaten by *TC*$_{SS}$*Stack-Joint* and *TC*$_{MS}$*Stack*. Since these two sets of methods only differ in the joint learning of the parameters, the outcomes point at the benefits of bringing in the inter-domain dependencies of the other sources during the learning. On the other side, the results for λ-Stacking and *TC*$_{MS}$*Stack* are better or comparable to those by *TC*$_{SS}$*Stack* and *TC*$_{SS}$*Stack-Joint*, repectively. This provides support to the correlation of the model importance with the similarity coefficients, which allows to specify the model by estimating fewer hyper-parameters while preserving the similarity capture capability.

7. Conclusions

We investigate a family of transfer covariance functions that represent the pairwise similarity between each source and the target domain for the *MSTR* problem. We prove that, *GP-TC*$_{MS}$, a Gaussian process with such a transfer covariance function can only capture the same similarity coefficient for all the sources. By further analyzing the generalization errors of *GP-TC*$_{MS}$, we conclude the bounds depend on the single similarity coefficient, which may penalize the transfer performance. As an alternative, we propose *TC*$_{MS}$*Stack*, an approach that integrates the transfer covariance function and the stacking strategy into one unified model. *TC*$_{MS}$*Stack* aligns the *S-T* similarity coefficients with the model importance and jointly learns the base models. Extensive experiments on one synthetic and two real-world datasets, with learning settings of up to 11 sources for the latter, show the superiority of *TC*$_{MS}$*Stack* to other *MSTR* methods.

Acknowledgments

This work was conducted within the Rolls-Royce@Nanyang Technological University Corporate Lab with support from the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme. It is also partially supported by the School of Computer Science and Engineering at Nanyang Technological University.

References

Al-Stouhi, Samir and Reddy, Chandan. Adaptive boosting for transfer learning using dynamic updates. *Machine Learning and Knowledge Discovery in Databases*, pp. 60–75, 2011.

Bhatt, Himanshu Sharad, Rajkumar, Arun, and Roy, Shourya. Multi-source iterative adaptation for cross-domain classification. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 3691–3697. AAAI Press, 2016.

Bonilla, Edwin V., Chai, Kian M., and Williams, Christopher. Multi-task gaussian process prediction. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T. (eds.), *Advances in Neural Information Processing Systems 20*, pp. 153–160. Curran Associates, Inc., 2008.

Cao, Bin, Pan, Sinno Jialin, Zhang, Yu, Yeung, Dit-Yan, and Yang, Qiang. Adaptive transfer learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pp. 407–712. AAAI Press, 2010.

Chai, Kian M. Generalization errors and learning curves for regression with multi-task gaussian processes. In *Advances in neural information processing systems*, pp. 279–287, 2009.

Dai, Wenyuan, Yang, Qiang, Xue, Gui-Rong, and Yu, Yong. Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pp. 193–200, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3.

Fang, Min, Guo, Yong, Zhang, Xiaosong, and Li, Xiao. Multi-source transfer learning based on label shared subspace. *Pattern Recognition Letters*, 51:101–106, 2015.

Ghosh, Sayan, Jacobs, Ryan, and Mavris, Dimitri N. Multi-source surrogate modeling with bayesian hierarchical regression. In *17th AIAA Non-Deterministic Approaches Conference*, pp. 1817–1829, 2015.

Lam, Kari Y, Westrick, Zachary M, Müller, Christian L, Christiaen, Lionel, and Bonneau, Richard. Fused regression for multi-source gene regulatory network inference. *PLOS Computational Biology*, 12:1–23, 2016.

Luo, Ping, Zhuang, Fuzhen, Xiong, Hui, Xiong, Yuhong, and He, Qing. Transfer learning from multiple source domains via consensus regularization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 103–112. ACM, 2008.

McAuley, Julian, Targett, Christopher, Shi, Qinfeng, and van den Hengel, Anton. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52. ACM, 2015.

Pan, Sinno Jialin and Yang, Qiang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

Pan, Sinno Jialin, Kwok, James T, and Yang, Qiang. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pp. 677–682, 2008.

Pan, Sinno Jialin, Tsang, Ivor W, Kwok, James T, and Yang, Qiang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

Pardoe, David and Stone, Peter. Boosting for regression transfer. In Frnkranz, Johannes and Joachims, Thorsten (eds.), *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 863–870. Omnipress, 2010.

Rasmussen, Carl Edward. Gaussian processes for machine learning. *Citeseer*, 2006.

Rasmussen, Carl Edward and Nickisch, Hannes. Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11:3011–3015, December 2010. ISSN 1532-4435.

Tommasi, Tatiana, Orabona, Francesco, and Caputo, Barbara. Learning categories from few examples with multi model knowledge transfer. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):928–941, 2014.

Torres-Sospedra, Joaquín, Montoliu, Raúl, Usó, Adolfo Martínez, Avariento, Joan P., Arnau, Tomas J., Benedito-Bordonau, Mauri, and Huerta, Joaquín. Ujiindoorloc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems. In *International Conference on Indoor Positioning and Indoor Navigation*, pp. 261–270. IEEE, 2014.

Wang, Qifan, Ruan, Lingyun, and Si, Luo. Adaptive knowledge transfer for multiple instance learning in image classification. In *AAAI*, pp. 1334–1340, 2014.

Wei, Pengfei, Ke, Yiping, and Goh, Chi Keong. Deep nonlinear feature coding for unsupervised domain adaptation. In *In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2189–2195. AAAI Press, 2016.

Williams, Christopher, Klanke, Stefan, Vijayakumar, Sethu, and Chai, Kian M. Multi-task gaussian process learning of robot inverse dynamics. In *Advances in Neural Information Processing Systems*, pp. 265–272, 2009.

Wolpert, David H. Stacked generalization. *Neural Networks*, 5(2):241–259, February 1992. ISSN 0893-6080.

Yao, Yi and Doretto, Gianfranco. Boosting for transfer learning with multiple sources. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pp. 1855–1862. IEEE, 2010.

Yong, Longquan. Uniform smooth approximation functions for absolute value function. *Mathematics in practice and theory*, pp. 250–255, 2015.

Zhang, Fuzhen. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

Zhou, Joey Tianyi, Pan, Sinno Jialin, Tsang, Ivor W, and Ho, Shen-Shyang. Transfer learning for cross-language text categorization through active correspondences construction. In *AAAI*, pp. 2400–2406, 2016.