Multi-Source Transfer Regression via Source-Target Pairwise Segment

Kai Yang[a,b], Jie Lu[b,*], Wanggen Wan[a], Guangquan Zhang[b]

[a] *School of Communication and Information Engineering, Institute of Smart City, Shanghai University, 200444, Shanghai, China*
[b] *School of Software Engineering, University of Technology Sydney, P.O. Box 123, Broadway NSW, Australia*

Abstract

Transfer learning addresses the problem of how to leverage acquired knowledge from a source domain to improve the learning efficiency and accuracy of the target domain that has insufficient labeled data. Instead of one source domain, multiple domains could be the source domains that are available for knowledge transfer in practice. However, there are large differences between the source and target domains, how to extract the useful knowledge from these different source domains remains a problem. To solve this problem, we propose a source-target pairwise segment method for multi-source transfer regression (STPS-MTR). The STPS-MTR method adaptively segments the different source domains and the target domain into different similar parts, and it extracts the most similar part in different source domains as the transfer knowledge. The STPS-MTR method can effectively extract the transfer knowledge from different source domains even when the source domain and the target domain have relatively low similarity, and it can avoid the negative influence between different source domains to ensure the transfer performance. Experimental results using synthetic datasets and real-world datasets demonstrate that the proposed method has better performance than existing methods, particularly when there are significant differences between different source domains and the target domain.

[*] Corresponding author
Email address: jie.lu@uts.edu.au (Jie Lu)

Keywords: Machine learning, Data mining, Transfer learning, Multi-source domain adaptation

1. Introduction

Transfer learning methods use the source domain to help train the target domain [19, 27, 15, 7, 13, 32]. The existing transfer learning methods require that the feature space or the distribution of the source domain and the target domain should have a certain similarity, and the similarity should be within a certain range. If the similarity is relatively low, the negative transfer may occur [21]. In the real-world datasets, one source domain may have limited knowledge to help target domain train a good model, and more than one source domain can be used to help train the target domain [24, 9, 29, 3, 34]. Thus, the multi-source domain transfer learning methods have been developed.

Multi-source domain transfer learning methods use different source domains to help train the target domain when the transfer performance with the single source domain is still not good. The main challenge of the multi-source domain transfer learning is how to describe the different similarities between the source domains and the target domain and how to solve the negative influence between different source domains. Many multi-source domain transfer learning methods theoretically analyze that the target domain can be expressed as the convex combination of the multiple source domains, and they demonstrate from the real-world applications that the different combination rules of the multi-source domain can improve the transfer performance [6, 28, 23, 26]. However, much of the research on transfer learning concerns classification problems, the problem of regression has been much less studied. Unlike classification problems, where the outcome variables are discrete values, the ones in regression problems are continuous.

In this paper, we focus on the multi-source domain transfer learning method for the regression problem. In previous studies, some researchers theoretically analyze that the target domain can be expressed as a convex combination of multiple source domains [17, 18, 12]. Different algorithms such as the fuzzy rule or Gaussian process are used to describe the different similarities between the source and target domains, and the combination rules are established to extract knowledge from different source domains [16, 25]. And according to the different similarities between the source domains and the target domain, the highest similarity source domain is selected to help train the target domain [35]. Although these proposed methods have good theoretical analysis and transfer performance, they did not consider that when the similarity between the source domain and the target domain is relatively low, how to extract the knowledge to do the transfer without the negative transfer, or when the distributions of the different source domains are very different, how to avoid the negative influence between different source domains.

To solve the abovementioned problem, we propose the source-target pairwise segment method for multi-source transfer regression (STPS-MTR). The STPS-MTR method adaptively segments the different source domains and the target domain into different similar parts, each similar part can satisfy the condition that the distributions of the source domain and the target domain have an approximately linear relationship, and the STPS-MTR method extracts the most similar part in different source domains as the knowledge transfer. The main contribution of this paper is that the STPS-MTR method can effectively extract the transfer knowledge from different source domains even when the source domain and the target domain have relatively low similarity, and it can avoid the negative influence between different source domains to ensure the transfer performance.

The remainder of this paper is structured as follows. Section 2 presents related work. Section 3 sets out preliminary knowledge. Section 4 describes the proposed STPS-MTR method. The experiments and results used to analyze and verify the method are presented in Section 5. The final section concludes the paper and outlines future work.

2. Related work

This section reviews the existing multi-source domain transfer learning methods, and the pairwise similarity transfer learning methods.

The main challenge of the multi-source domain transfer learning methods is to establish a combination rule for the source domains and the target domain. Yao et al. [30] use two different methods to solve the multiple source domains transfer learning problems. One is to get the useful knowledge of the target domain by summarizing all the multiple source domains, the other one is to find the mapping knowledge between each source domain and the target domain. Duan et al. [8] modify the least-squares SVM model by using the smoothness assumption, it adds the dependent regularization to enforce the target domain and source domains to share similar decision values. Chattopadhyay et al. [4] consider the conditional probability of the multi-source domain adaptation, and it proposes a two-stage weighting framework for multi-source domain adaptation. One is the marginal probability difference of the reweighted source domain, and the other is the conditional probability difference of the reweighted source domain. Zhao et al. [33] propose a new error bound for multiple source domain adaptation. The method does not need to know the target distribution and the combination rule of the multiple source domains, and it can automatically find the relationship between the source domains and the target domain using the adversarial network. Although these methods have a good performance, these methods only consider the situation that the distributions of the different source domains and the target domain have high similarity.

To solve the problem that the distributions of the different source domains and the target domain are quite different. Gress and Davidson [11] propose the pairwise similarity regularization transfer method, a flexible graph-based regularization framework that can incorporate this modeling assumption into standard supervised learning algorithms. Domain knowledge is encoded into the regularizer in the form of spatial continuity and pairwise "similarity constraints", and the method is extended to large data sets using Nystrom approximation. Long et al. [14] use the joint distribution adaptation (JDA) to describe the gap between the source domain and the target domain. When the marginal and conditional distribution of source domain and target domain are different, the maximum mean discrepancy (MMD) and generalized rayleigh quotient are used to optimize the objective function. JDA jointly adapts the marginal distribution and conditional distribution in a principled dimensionality reduction procedure, and it constructs new effective and robust feature representations to cope with large distribution differences. Courty et al. [5] propose the unsupervised domain adaptation method. The method assumes that the target domain can be represented as a nonlinear function with the source domain. It uses the separation ideal to recover the objective function when some parts of the source domain and some parts of the target domain have similar distributions. Through the joint estimation of the source domain data, the target function can be optimized. Shao and Wu [22] propose an information-based criterion for determining the number of clusters in the problem of regression clustering. The method shows that in the population of a probabilistic structure, the criterion selects a real number of regression hyperplanes with a probability of one in all class-growing classification sequences when the observed value of the population increases to infinity. However, these methods do not focus on the multi-source domain transfer learning problem.

In this work, we propose the source-target pairwise segment method which adaptively segments the source domain and the target domain into different similar parts, and we build the combination rule to extract the most useful knowledge for multiple source domains.

3. Preliminary Knowledge

In this section, we introduce the problem statement and the adaptive transfer learning method based on Gaussian process (AT-GP).

3.1. Problem Statement

In this paper, we consider the multi-source domain transfer learning for regression problem, particularly when there are significant differences between different source domains and the target domain.

We denote the source domains $S = \{S_1, ..., S_N\}$, where N is the number of source domains, $S_j = \{(x_1^{(S_j)}, y_1^{(S_j)}), ..., (x_{n^{(S_j)}}^{(S_j)}, y_{n^{(S_j)}}^{(S_j)})\}$, $x_i^{(S_j)}$ is the data instance and $y_i^{(S_j)}$ is the corresponding label, $n^{(S_j)}$ is the number of data in S_j. Similarly, the target domain $T = \{(x_1^{(T)}, y_1^{(T)}), ..., (x_{n^{(T)}}^{(T)}, y_{n^{(T)}}^{(T)})\}$, where $x_i^{(T)}$ is the input data and $y_i^{(T)}$ is the corresponding output data, $n^{(T)}$ is the number of data in target domain. The marginal probability distributions of the source and target domains are $P(x_i^{(S_j)})$ and $P(x_i^{(T)})$, the conditional probability distributions of the source and target domains are $P(y_i^{(S_j)}|x_i^{(S_j)})$ and $P(y_i^{(T)}|x_i^{(T)})$.

The source domains have a large amount of labeled data, but the target domain has very few labeled data, $n^{(S_j)} \gg n^{(T)}$. The dimensions of $x_i^{(S_j)}$ and $x_i^{(T)}$ are the same, and the dimensions of $y_i^{(S_j)}$ and $y_i^{(T)}$ are 1. Additionally, the distributions of the different source domains and the target domain can be very different, $P(x_i^{(S_j)}) \neq P(x_i^{(T)})$ and $P(y_i^{(S_j)}|x_i^{(S_j)}) \neq P(y_i^{(T)}|x_i^{(T)})$.

Our objective is to extract useful knowledge from different source domains to help train the target domain and to avoid the negative influence between different source domains, particularly when there are significant differences between different source domains and the target domain.

3.2. The AT-GP method

In a regression problem, a Gaussian process (GP) [20] is used as the prior for Bayesian inference. The AT-GP method performs well when the distributions of the source domain and the target domain have an approximately linear relationship [2].

AT-GP method uses the Gamma distribution to describe the similarity between the source domain and the target domain, instead of using a point estimation, the proposed transfer kernel is

$$K_{nm} = \begin{cases} (2(\frac{1}{1+\mu})^b - 1)K(x_n, x_m), \ \zeta(x_n, x_m) = 1, \\ K(x_n, x_m), \ otherwise. \end{cases} \tag{1}$$

Where $K(x_n, x_m)$ is the covariance function of x_n and x_m, $\zeta(x_n, x_m) = 0$, if x_n and x_m come from the same domain, otherwise, $\zeta(x_n, x_m) = 1$, and $b \geq 0$, $\mu \geq 0$.

The conditional distribution of the outputs $y^{(S)}$ and $y^{(T)}$ conditions on corresponding inputs $x^{(S)}$ and $x^{(T)}$ as the posterior distribution for the target test data, and it can be written in a Gaussian form as follows,

$$p(y^{(T)}|y^{(S)}, x^{(T)}, x^{(S)}) \sim N(\mu_T, C_T), \tag{2}$$

where

$$\mu_T = K_{21}(K_{11} + \sigma_S^2 I)^{-1} y^{(S)}, \tag{3}$$

$$C_T = (K_{22} + \sigma_T^2 I) - K_{21}(K_{11} + \sigma_S^2 I)^{-1} K_{12}, \tag{4}$$

and $K_{11} = K(x^{(S)}, x^{(S)})$, $K_{12} = \lambda K(x^{(S)}, x^{(T)})$, $K_{21} = \lambda K(x^{(T)}, x^{(S)})$, $K_{22} = K(x^{(T)}, x^{(T)})$, $\lambda = 2(\frac{1}{1+\mu})^b - 1$, $b \geq 0$, $\mu \geq 0$, σ_S and σ_T are the noise items of the source domain and the target domain.

The log likelihood function is

$$log\ p(y^{(T)}|\theta) = -\frac{1}{2}log|C_T| - \frac{1}{2}(y^{(T)} - \mu_T)^T C_T^{-1}(y^{(T)} - \mu_T) - \frac{n}{2}log(2\pi), \tag{5}$$

where θ are the covariance parameters, n is the number of the training data. The values for θ result from maximizing the log likelihood function, and it normally assumes that the model belongs to the zero-mean Gaussian.

Wei et al. [25] extend the AT-GP method to a multi-source domain transfer learning method ($TC_{MS}Stack$), the $TC_{MS}Stack$ method uses the stacking strategy to describe the different similarities between the different source domains and the target domain. However, the performance of $TC_{MS}Stack$ will be affected when there are significant differences between the different source domains and the target domain. And the AT-GP method also has the constraint that when the similarity between the source domain and the target domain is relatively low, the negative transfer may occur.

4. The source-target pairwise segment for multiple source transfer regression method

In this paper, we mainly research two problems in the multi-source domain transfer learning regression method. The first is how to extract the knowledge to do the transfer without the negative transfer when the similarity between the source domain and the target domain is relatively low, and the second is how to avoid the negative influence when the distributions of the different source domains are very different. This section presents our STPS-MTR method.

4.1. Problem Setting and Motivation

Since the AT-GP method performs well when the distribution of the source domain and the target domain is approximately linear, and the AT-GP method is based on the kernel, this linear relationship also includes the linear relationship of the function in the kernel space. Therefore, we propose a source-target pairwise segment method to solve the situation that the similarity between the source domain and the target domain is relatively low. This method can adaptively divide the source and target domains into different similar parts so that each similar part satisfies the distribution of the source domain and the target domain in an approximately linear relationship, and we use the AT-GP method to train these similar parts separately.

Although we segment the different source domains and target domain into different similar parts, the positive and negative correlations of these similar parts will occur the negative influence between different source domains. We assume the distribution functions of the source domain and the target domain are $f_S(x)$ and $f_T(x)$, and $f_T(x) \approx \lambda f_S(x)$ where λ is the correlation parameter between the source domain and the target domain, if $\lambda \geq 0$, we call it that the source domain and the target domain have the positive correlation, similarity, if $\lambda < 0$, we call it that the source domain and the target domain have the negative correlation. To avoid the negative influence, we should consider the positive correlation and the negative correlation separately. However, when the distributions of the different source domains and the target domain are very different, even if we build the independent combination rule for the positive or negative correlation of these similar parts, it still occur unpredictable influence. Thus, we propose a combination method that extracts the most similar part in different source domains as the knowledge transfer to avoid the negative influence.

4.2. The STPS-MTR method

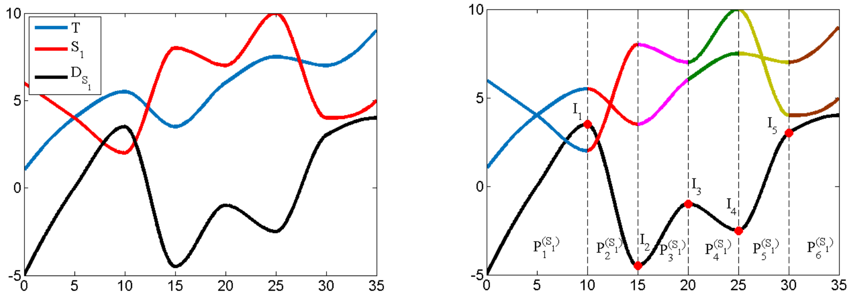
The STPS-MTR method has two purposes. The first one is to divide the different source domains and the target domain into different similar parts according to the different similarities between the source domains and the target domain. The second one is to build a combination rule to extract the jointly similar parts between different source domains, and the most similar part is selected in different source domains.

4.2.1. Explanation and Definition

To achieve the first purpose, we need to segment each source domain and the target domain into different pairwise parts. And these pairwise parts should satisfy the condition that the discrepancy between each pairwise part has an approximately linear relationship.

For the source domain S_j and the target domain T, we denote the source function is $f_{S_j}(x)$, $y_i^{(S_j)} = f_{S_j}(x_i^{(S_j)})$, the target function is $f_T(x)$, $y_i^{(T)} = f_T(x_i^{(T)})$, the discrepancy function between the target domain T and the source domain S_j is $f_{D_{S_j}}(x)$, $y_i^{(D_{S_j})} = f_{D_{S_j}}(x_i^{(D_{S_j})})$, where $x^{(D_{S_j})} = \{x_1^{(D_{S_j})}, ..., x_{n^{(D_{S_j})}}^{(D_{S_j})}\}$ are the input data of the discrepancy function, $x^{(D_{S_j})}$ include the input data of the source domain S_j and the target domain T, $x^{(D_{S_j})} = \{x^{(S_j)}, x^{(T)}\}$, $x^{(S_j)} = \{x_1^{(S_j)}, ..., x_{n^{(S_j)}}^{(S_j)}\}$, $x^{(T)} = \{x_1^{(T)}, ..., x_{n^{(T)}}^{(T)}\}$, $n^{(D_{S_j})}$ is the number of data in $x^{(D_{S_j})}$, $n^{(D_{S_j})} = n^{(S_j)} + n^{(T)}$, $y_i^{(D_{S_j})}$ is the output data of the discrepancy function, and its value is the discrepancy between pairs of instances $y_i^{(D_{S_j})} = f_T(x_i^{(D_{S_j})}) - f_{S_j}(x_i^{(D_{S_j})})$, as shown in Figure 1(a).

We get all the inflection points $I = \{I_1, ..., I_{n^{(I)}}\}$ of $f_{D_{S_j}}(x^{(D_{S_j})})$ by calculating the partial derivatives value of $f_{D_{S_j}}(x^{(D_{S_j})})$, $\frac{\partial f_{D_{S_j}}(x^{(D_{S_j})})}{\partial x^{(D_{S_j})}} = 0$, where $n^{(I)}$ is the number of inflection points. According to these inflection points, two adjacent inflection points become the segmented region. The source domain S_j and the target domain T are segmented into different pairwise parts, $P_k^{(S_j)} = \{(x_1^{(P_k)}, y_1^{(P_k)}), ..., (x_{n^{(P_k)}}^{(P_k)}, y_{n^{(P_k)}}^{(P_k)})\}$, where $x_i^{(P_k)}$ is data of $x^{(D_{S_j})}$ which belongs to the segmented region, $y_i^{(P_k)}$ is the corresponding output data of $x_i^{(P_k)}$, and $n^{(P_k)}$ is the number of data in $P_k^{(S_j)}$. In addition, we consider the noise bound that using the approximate solution to solve the partial derivative equation, we get the solution of $f_{D_{S_j}}(x^{(D_{S_j})})$, $\frac{\partial f_{D_{S_j}}(x^{(D_{S_j})})}{\partial x^{(D_{S_j})}} \approx 0$, instead of the solution of $f_{D_{S_j}}(x^{(D_{S_j})})$, $\frac{\partial f_{D_{S_j}}(x^{(D_{S_j})})}{\partial x^{(D_{S_j})}} = 0$. As shown in Figure 1(b), each pairwise part $P_k^{(S_j)}$ is described by different colors, and it can satisfy the condition that the distributions of the source domain and target domain have an approximately linear relationship.



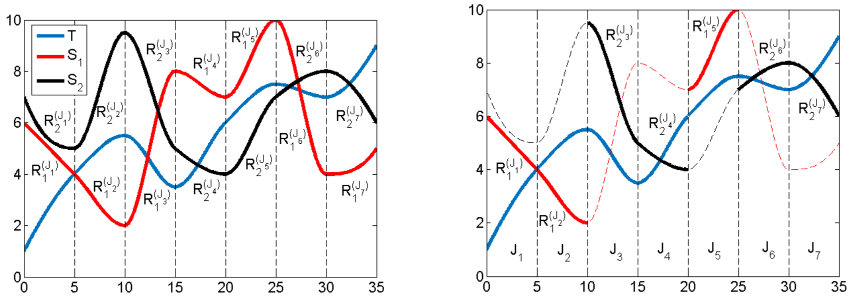
(a)

(b)

Figure 1: The pairwise part. (a) The distributions of S_1 and T, and the discrepancy between S_1 and T. (b) Five inflection points I and the segment result of six pairwise parts $P^{(S_1)}$.

To achieve the second purpose, we need to obtain the joint parts of the different pairwise parts, and according to the different source domains, we respectively train the prediction model for each joint part and select the part with the minimum error in each joint part.

Based on the different pairwise part $P_k^{(S_j)}$, we denote the joint part $J_l = \{(x_1^{(J_l)}, y_1^{(J_l)}), ..., (x_{n^{(J_l)}}^{(J_l)}, y_{n^{(J_l)}}^{(J_l)})\}$, where $x_i^{(J_l)}$ is the input data of $P_k^{(S_j)}$ which belongs to the jointly segmented region of all source domains, $y_i^{(J_l)}$ is the corresponding output data of $x_i^{(J_l)}$, and $n^{(J_l)}$ is the number of data in J_l, as shown in Figure 2(a), for each joint part J_l, it can satisfy the condition that the distributions of each source domain and target domain have an approximately linear relationship. We classify J_l according to different source domains and the target domain, $J_l = \{R_1^{(J_l)}, ..., R_N^{(J_l)}\}$, where $R_m^{(J_l)} = \{(x_1^{(R_m)}, y_1^{(R_m)}), ..., (x_{n^{(R_m)}}^{(R_m)}, y_{n^{(R_m)}}^{(R_m)})\}$, $(x_i^{(R_m)}, y_i^{(R_m)}) \in \{S_j, T\}$, $n^{(R_m)}$ is the number of data in $R_m^{(J_l)}$, and N is the number of source domains. We respectively train the prediction model for each $R_m^{(J_l)}$ by using AT-GP method, and select the minimum error part of $R_m^{(J_l)}$ in each J_l. And we combine all these minimum error part as the final prediction model, the selected parts are described by the solid line as shown in Figure 2(b).



(a)

(b)

Figure 2: The joint part. (a) Seven joint parts of the source domain S_1, the source domain S_2 and the target domain T, and the classified joint parts $R_1^{(J)}$ and $R_2^{(J)}$. (b) The minimum error part of $R_m^{J_l}$ in each J_l.

4.2.2. The STPS-MTR method description

We theoretically analyze the STPS-MTR method, however, in the real-word application, when the dimension of the input data is relatively high, it is difficult to find all the inflection points of the partial derivative equation. If we segment the source domains and the target domain into too many different pairwise parts, it will occur overfitting problem. Thus, we separately consider each dimension data of the input data and combine the segmented result of each dimension data to obtain the approximate solution of the partial derivative function with the noise bound. Table 1 summarizes frequently-used notations in STPS-MTR and the detail of the STPS-MTR method as follows.

Table 1: Notations and descriptions

Notations	Descriptions	Notations	Descriptions
S_j	The source domain	$x_i^{(S_j)}$, $y_i^{(S_j)}$	The input and output data of S_j
N	The number of source domains	$n^{(S_j)}$	The number of data in S_j
T	The target domain	$x_i^{(T)}$, $y_i^{(T)}$	The input and output data of T
$n^{(T)}$	The number of data in T	D_{S_j}	The discrepancy between S_j and T
$x_i^{(D_{S_j})}$, $y_i^{(D_{S_j})}$	The input and output data of D_{S_j}	$n^{(D_{S_j})}$	The number of data in D_{S_j}
$d_p^{(x_i)}$	The p_{th} dimension data of $x_i^{(D_{S_j})}$	$n^{(d)}$	The dimension number of $x_i^{(D_{S_j})}$
E_p	The p_{th} dimension data of D_{S_j}	$d_i^{(E_p)}$, $y_i^{(E_p)}$	The input and output data of E_p
$n^{(E_p)}$	The number of data in E_p	M	The source-target segment matrix
I	The inflection points	$n^{(I)}$	The number of inflection points
J_l	The joint part of different source domains	$x_i^{(J_l)}$, $y_i^{(J_l)}$	The input and output data of J_l
$n^{(J_l)}$	The number of data in J_l	$P_k^{(S_j)}$	The pairwise part
$x_i^{(P_k)}$, $y_i^{(P_k)}$	The input and output data of $P_k^{(S_j)}$	$n^{(P_k)}$	The number of data in $P_k^{(S_j)}$
$R_m^{(J_l)}$	The classified joint part	$x_i^{(R_m)}$, $y_i^{(R_m)}$	The input and output data of $R_m^{(J_l)}$
$n^{(R_m)}$	The number of data in $R_m^{(J_l)}$		

Step 1: Initialize the discrepancy between the source domain and the target domain.

For the source domain S_j and the target domain T, we select the Euclidean distance and cluster the inputs data $x^{(S_j)}$ of S_j into $n^{(T)}$ groups $G = \{G_1, ..., G_{n^{(T)}}\}$ by using the inputs data $x^{(T)}$ of T as the clustering center, where $n^{(T)}$ is the number of groups, G_g includes one target domain data $(x_i^{(T)}, y_i^{(T)})$ and the other clustering source domain data. In each clustering group G_g, we select the source domain data which is the nearest one to the target domain data, the selected source domain data and the target domain data as the pairwise data, $\{(x_{G_g}^{(S_j)}, y_{G_g}^{(S_j)}), (x_i^{(T)}, y_i^{(T)})\}$. For the discrepancy between S_j and T, $D_{S_j} = \{(x_1^{(D_{S_j})}, y_1^{(D_{S_j})}), ..., (x_{n^{(D_{S_j})}}^{(D_{S_j})}, y_{n^{(D_{S_j})}}^{(D_{S_j})})\}$, where the input data $x_i^{(D_{S_j})} = x_i^{(T)}$, the output data $y_i^{(D_{S_j})} = y_i^{(T)} - y_{G_g}^{(S_j)}$, and $n^{(D_{S_j})}$ is the number of data in D_{S_j}.

To avoid the overfitting problem, we separately consider each dimension data of $x_i^{(D_{S_j})} = [d_1^{(x_i)}, ..., d_{n^{(d)}}^{(x_i)}]$, where $d_p^{(x_i)}$ is the p_{th} dimension data of $x_i^{(D_{S_j})}$ and $n^{(d)}$ is the dimension number of $x_i^{(D_{S_j})}$, and we add the noise bound $B = |max\ d_p^{(x_i)} - min\ d_p^{(x_i)}|/\alpha$, where α is the positive integer, $\alpha \in [2, \frac{3}{4}n^{(D_{S_j})}]$, $max\ d_p^{(x_i)}$ is the maximum value of $d_p^{(x_i)}$ in D_{S_j}, and $min\ d_p^{(x_i)}$ is the minimum value of $d_p^{(x_i)}$ in D_{S_j}. For each dimension data of $x_i^{(D_{S_j})}$, we denote the p_{th} dimension data of D_{S_j} is $E_p = \{(d_1^{(E_p)}, y_1^{(E_p)}), ..., (d_{n^{(E_p)}}^{(E_p)}, y_{n^{(E_p)}}^{(E_p)})\}$, where $d_i^{(E_p)}$ is the p_{th} dimension data of $x_i^{(D_{S_j})}$, $y_i^{(E_p)}$ is the corresponding output data of $x_i^{(D_{S_j})}$, and $n^{(E_p)}$ is the number of data in E_p. We sort E_p by $d_i^{(E_p)}$ and get the adjacent data $(d_a^{(E_p)}, y_a^{(E_p)})$ and $(d_b^{(E_p)}, y_b^{(E_p)})$ of E_p, and if $|d_a^{(E_p)} - d_b^{(E_p)}| \leq B$, these two adjacent data of E_p combine into one new data $(\frac{d_a^{(E_p)}+d_b^{(E_p)}}{2}, \frac{y_a^{(E_p)}+y_b^{(E_p)}}{2})$.

Step 2: Calculate the inflection points.

To get the inflection points I of E_p, we denote the source-target segment matrix M. According to the sort order of E_p, we generate the upper triangular matrix M, each index (i, j) of the matrix M means from the i_{th} data of E_p to the j_{th} data of E_p, and each element of the matrix is the value of the R^2 statistic for the regression fitting using the data from the i_{th} data of E_p to the j_{th} data of E_p. We replace all the elements in M that are equal to 1 with 0, and search the inflection points I.

Start searching from i_{th} row and $(i+1)_{th}$ row respectively, and find the first time appear the value a of $M(i, j)$ less than the value b of $M(i+1, k)$ and $|a - b| < \delta$, where δ is the noise item, $\delta \in (0, 0.1)$. Compare the index j and k, if $j \neq k$, the i_{th} inflection point I_i is the i_{th} data of E_p, the $(i+1)_{th}$ inflection point I_{i+1} is the j_{th} data of E_p, and if $j = k$, compare the value of $M(i, j)$ and $M(i+1, j)$, if $M(i, j) \leq M(i+1, j)$, the result is the same, if $M(i, j) > M(i+1, j)$, the i_{th} inflection point I_i is the $(i+1)_{th}$ data of E_p, the $(i+1)_{th}$ inflection point I_{i+1} is the j_{th} data of E_p.

Step 3: Obtain the pairwise parts.

The inflection points of E_p are used to segment E_p. The adjacent inflection points become the segmented region, E_p is segmented into different parts, each part $H_q^{(E_p)} = \{(d_1^{(H_q)}, y_1^{(H_q)}), ..., (d_{n^{(H_q)}}^{(H_q)}, y_{n^{(H_q)}}^{(H_q)})\}$, where $d_i^{(H_q)}$ is the data of $d_i^{(E_p)}$ which belongs to the segmented region, $y_i^{(H_q)}$ is the corresponding output data of $d_i^{(E_p)}$, and $n^{(H_q)}$ is the number of data in $H_q^{(E_p)}$. E_p is the p_{th} dimension data of D_{S_j}, after getting the segmented results of all the dimension data of D_{S_j}, all the segmented results are combined to get the pairwise parts of D_{S_j}.

We extract the intersection part C of all $H_q^{(E_p)}$, $C = H_q^{(E_1)} \bigcap ... \bigcap H_q^{(E_{n^{(d)}})}$. If C has more than 3 intersection data, it means that D_{S_j} has the same inflection points from different dimension of $x_i^{(D_{S_j})}$ and the distributions of the source domain and target domain have an approximate linear relationship in this segmented region, we obtain the pairwise part $P_k^{(S_j)} = \{(x_1^{(P_k)}, y_1^{(P_k)}), ..., (x_{n^{(P_k)}}^{(P_k)}, y_{n^{(P_k)}}^{(P_k)})\}$, where $x_i^{(P_k)}$ is the input data of intersection data, $y_i^{(P_k)}$ is the corresponding output data of $x_i^{(P_k)}$, and $n^{(P_k)}$ is the number of data in $P_k^{(S_j)}$. And if C has less than 3 data, it means that D_{S_j} does not have the same inflection points from different dimension of $x_i^{(D_{S_j})}$ and the distributions of the source domain and target domain cannot have an approximate linear relationship in this segmented region, so we abandon this intersection part C to prevent overfitting problem.

Step 4: Get the joint parts of different source domains.

We separately obtain the the pairwise part $P_k^{(S_j)}$ of different source domains, and we get the joint part by combine all the pairwise parts, $J_l = P_k^{(S_1)} \bigcap ... \bigcap P_k^{(S_N)}$.

To avoid the overfitting problem, if $J_l = \emptyset$ or the number of the data of J_l less than 3, we use the except set to replace the intersect set, $J_l = P_k^{(S_1)} - ... - P_k^{(S_N)}$. The joint part is $J_l = \{(x_1^{(J_l)}, y_1^{(J_l)}), ..., (x_{n^{(J_l)}}^{(J_l)}, y_{n^{(J_l)}}^{(J_l)})\}$, where $x_i^{(J_l)}$ is the data of $x_i^{(P_k)}$ which belongs to the jointly segmented region for different source domains, $y_i^{(J_l)}$ is the corresponding output data of $x_i^{(J_l)}$, and $n^{(J_l)}$ is the number of data in J_l.

Step 5: Build the prediction model.

To avoid the negative influence problem, we classify J_l according to different source domains and the target domain, $J_l = \{R_1^{(J_l)}, ..., R_N^{(J_l)}\}$, where $R_m^{(J_l)} = \{(x_1^{(R_m)}, y_1^{(R_m)}), ..., (x_{n^{(R_m)}}^{(R_m)}, y_{n^{(R_m)}}^{(R_m)})\}$, $(x_i^{(R_m)}, y_i^{(R_m)}) \in \{S_j, T\}$, $n^{(R_m)}$

is the number of data in $R_m^{(J_l)}$.

We respectively train the classified joint part $R_m^{(J_l)}$ by using AT-GP method, and get prediction function $f^*(x)$. And we select the minimum error part of $R_m^{(J_l)}$ in each J_l, $min\ \Sigma_{j=1}^N(y_j^{(T)} - f^*(x_j^{(T)}))^2$. For the target unlabeled test data $x_u^{(T)}$, we calculate the distance between the target labeled data $x_j^{(T)}$ and the target unlabeled test data $x_u^{(T)}$, and get the nearest one data $x_{min}^{(T)}$, $x_u^{(T)}$ and $x_{min}^{(T)}$ belong to the same joint part. In this joint part, we use the minimum error prediction function to predict the output value $y_u^{(T)}$.

The pseudo-code of the STPS-MTR method is given in Algorithm 1, the STPS-MTR method includes a segmented process and a training process, the time complexity of the segmented process is $O(n^2)$, and we use the AT-GP method to train the model, the time complexity is $O(n^3)$. The STPS-MTR method can adaptively segment the different source domains and the target domain into different similar parts, and it can extract the most similar part in different source domains as the knowledge transfer. The STPS-MTR method effectively extracts the transfer knowledge even when the similarity between different source and target domains is relatively low, and it avoids the negative influence between different source domains. The proposed STPS-MTR method has better transfer performance than existing methods, particularly when the distributions of the different source domains and the target domain are very different. The results are shown in the experiments.

5. Experiments and results analysis

To evaluate the proposed STPS-MTR method, we develop synthetic datasets and select four real-world public datasets to test scenarios when the distributions of the source domains and the target domain are significantly different. The details are described in the following subsections.

5.1. The experiments with Synthetic datasets

The experiments with synthetic data are designed to evaluate the performance of the proposed STPS-MTR. To best illustrate the characteristics of the

Algorithm 1: The pseudo-code of STPS-MTR method

```
Input: The source domain data $S_j$ and the target domain $T$, the number
       of the source domain $N$, the dimension number of $n^{(d)}$, and the
       unlabeled target domain data $x_u^{(T)}$.
Output: The predict value $y_u^{(T)}$ for $x_u^{(T)}$.
for $j \leftarrow 1$ to $N$ do
    Initialize the discrepancy between $S_j$ and $T$
    for $p \leftarrow 1$ to $n^{(d)}$ do
        Get the $p_{th}$ dimension data $E_p$ of $D_{S_j}$
        Generate the inflection points $I$ of $E_p$
    Combine each dimension segment results to get the pairwise part $P_k^{(S_j)}$
Get the joint parts for all source domains $J_l$
Obtain the classified joint part $R_m^{(J_l)}$
Get the minimum error part of $R_m^{(J_l)}$ in each $J_l$
Determine the prediction model $min\Sigma_{j=1}^{N}(y_j^{(T)} - f^*(x_j^{(T)}))^2$
Return $y_u^{(T)}$
```

segment process using STPS-MTR, we select one-dimensional input data and two-dimensional input data as two case studies. The details of the task follow.

5.1.1. One-dimensional input data

The four source domains are $y_1 = 2x - 8$, $y_2 = -(x-5)^2 + 12$, $y_3 = -10sin^2(x) + 3sin(x) + 2$, $y_4 = 10cos^2(x+2) - 3cos(x) - 5$, the target domain is $y = 10sin(x)$. We generate 100 points, the spacing between the points is 0.11, $x \in [0, 10]$, and 11 points are selected as the target domain training data, the spacing between the points is 1, the others as the target testing data. The segment process as follow.

Step 1: for the first source domain $y_1 = 2x - 8$, according to the 11 target training points, the source domain and the target domain are clustered into 11 groups G, e.g. $G_1 = \{(x_1^{(T)}, y_1^{(T)}), (x_1^{(S_1)}, y_1^{(S_1)}), ..., (x_4^{(S_1)}, y_4^{(S_1)})\}$, there are 4 source domain points and one target domain point.

Step 2: we initialize the discrepancy between the source domain and the target domain, $D_{S_1} = \{(x_1^{(D_{S_1})}, y_1^{(D_{S_1})}), ..., (x_{11}^{(D_{S_1})}, y_{11}^{(D_{S_1})})\}$, where $x_i^{(D_{S_1})} = x_i^{(T)}$, $y_i^{(D_{S_1})} = y_i^{(T)} - y_G^{(S_1)}$, $y_G^{(S_1)}$ is the output data of the source domain data which is the nearest one to the target domain data in G, $min\{\|x_1^{(S_1)} - x_1^{(T)}\|_2, ..., \|x_4^{(S_1)} - x_1^{(T)}\|_2\}$, $x_G^{(S_1)} = x_1^{(S_1)}$, and $y_G^{(S_1)} = y_1^{(S_1)}$.

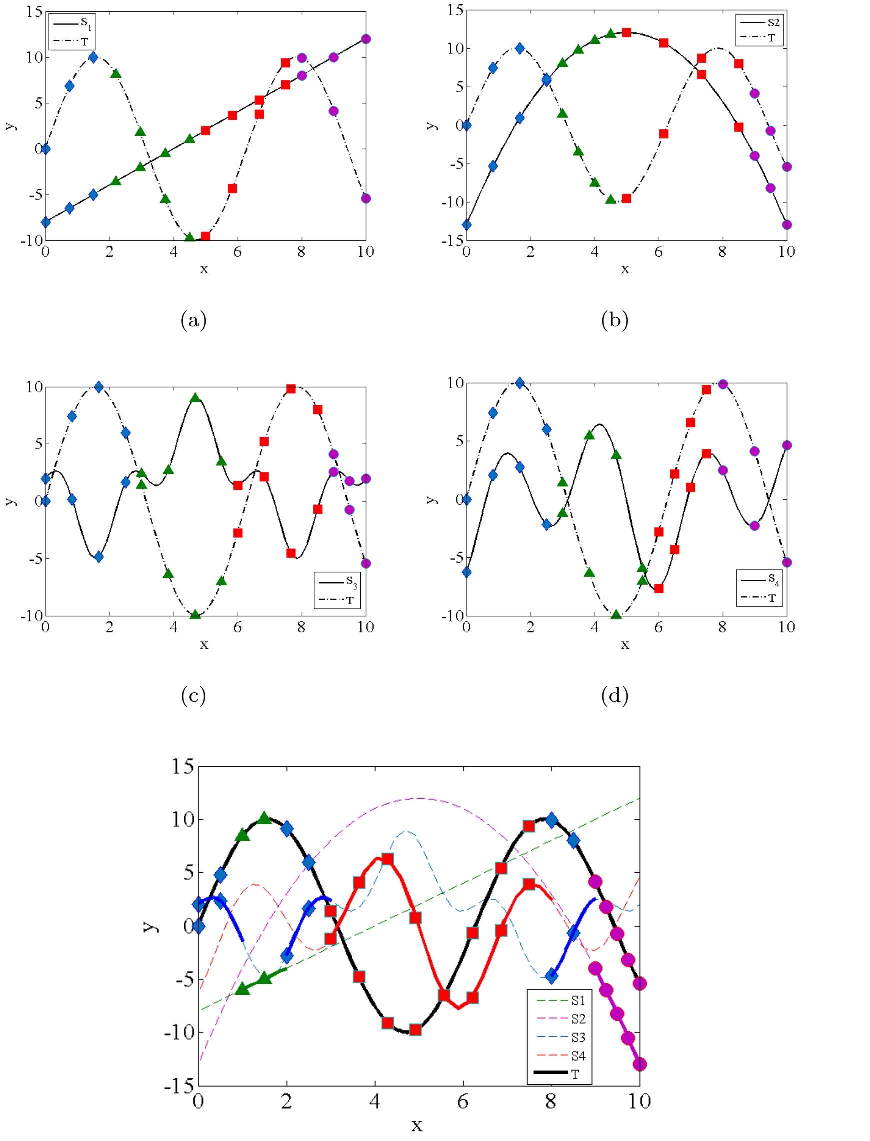
Step 3: we generate the source-target segment matrix M is

$$M = \begin{bmatrix} 0.73 & 0.07 & 0.53 & 0.74 & 0.76 & 0.55 & 0.38 & 0.42 & 0.55 \\ 0 & 0 & 0.99 & 0.98 & 0.82 & 0.40 & 0.17 & 0.21 & 0.39 \\ 0 & 0 & 0 & 0.96 & 0.64 & 0.07 & 0.01 & 0.01 & 0.18 \\ 0 & 0 & 0 & 0 & 0.10 & 0.36 & 0.61 & 0.28 & 0.02 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.86 & 0.05 & 0.26 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.57 & 0.81 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.98 \end{bmatrix}.$$

And we get 3 inflection points, $I_1 = (x_2^{(D_{S_1})}, y_2^{(D_{S_1})})$, $I_2 = (x_5^{(D_{S_1})}, y_5^{(D_{S_1})})$, $I_3 = (x_8^{(D_{S_1})}, y_8^{(D_{S_1})})$.

Step 4: we get pairwise part $P_k^{(S_j)}$ and define $\{i\}$ as $\{(x_i^{(D_{S_j})}, y_i^{(D_{S_j})})\}$. The segment result of the first source domain is {1, 2}, {3, 4, 5}, {6, 7, 8}, {9, 10, 11}; The segment result of the second source domain is {1, 2, 3}, {4, 5}, {6, 7, 8, 9}, {10, 11}; The segment result of the third source domain is {1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11}; The segment result of the fourth source domain is {1, 2, 3}, {4, 5, 6}, {7, 8}, {9, 10, 11}. The result of the multi-source domains segment is {1: 3, 2: 1, 3: 3, 4: 4, 5: 4, 6: 4, 7: 4, 8: 4, 9: 3, 10: 2, 11: 2}, where {1: 3} means the $\{(x_1^{(D_{S_3})}, y_1^{(D_{S_3})})\}$ chooses the third source domain S_3 as the final prediction model.

The segment result as shown in Figure 3, the different colors and different shapes describe the different pairwise parts, as we can see, from Figure 3(a) to Figure 3(d), the four source domains and the target domain are respectively segmented into 4 different pairwise parts. We use the AT-GP method to train these pairwise parts separately and select the most similar part, Figure 3(e) shows the target domain selects the most similar parts of the four source domains as the training model, each selected part is described by the solid line, and the same color and shape mean that the target domain selects the same source domain. The prediction error MAE (mean absolute error) and MSE (mean square error) of these four source domains are shown in Table 2.



(a)

(b)

(c)

(d)

(e)

Figure 3: The segment results of the four source domains and the target domain. (a) S_1 and T are segmented into 4 pairwise parts. (b) S_2 and T are segmented into 4 pairwise parts. (c) S_3 and T are segmented into 4 pairwise parts. (d) S_4 and T are segmented into 4 pairwise parts. (e) Target domain selects the most similar parts of the four source domains.

Table 2: The prediction result of 4 source domains

Source	1_{th}	2_{th}	3_{th}	4_{th}	All
MAE	4.87	4.70	4.41	5.46	2.48
MSE	33.60	32.11	24.53	39.52	11.83

5.1.2. Two-dimensional input data

The three source domains are $y_1 = 0.1(x_1^2+x_1^2)-30$, $y_2 = 2x_1^2e^{-(0.01x_1^2+0.01x_2^2)}-30$, $y_3 = -15(sin(\frac{x_1}{3})+cos(\frac{x_2}{3}))$, the target domain is $y = 15(sin(\frac{x_1}{4})+cos(\frac{x_2}{4}))$. We generate 121 points, the spacing between the points is π, $x_1, x_2 \in [-5\pi, 5\pi]$ and 25 points are selected as the target domain training data, the spacing between the points is 2.5π, the others as the target testing data.

For the first source domain $y_1 = 0.1(x_1^2 + x_1^2) - 30$, the input data have two dimensions x_1 and x_2, the dimension x_1 is segmented into two pairwise parts, we also define {1} as the $\{(x_1^{(D_{S_1})}, y_1^{(D_{S_1})})\}$, the first pairwise part is {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}, the second pairwise part is {16, 17, 18, 19, 20, 21, 22, 23, 24, 25}. Similarity, the dimension x_2 is segmented into two pairwise parts, {1} as the $\{(x_1^{(D_{S_1})}, y_1^{(D_{S_1})})\}$, the first pairwise part is {1, 2, 3, 6, 7, 8, 11, 12, 13, 16, 17, 18, 21, 22, 23}, the second pairwise part is {3, 4, 5, 8, 9, 10, 13, 14, 15, 18, 19, 20, 23, 24, 25}. After combining these two dimension segment results, we get four joint parts, J_1 is {1, 2, 3, 6, 7, 8, 11, 12, 13, 16, 17, 18}, J_2 is {3, 4, 5, 8, 9, 10, 13, 14, 15, 18, 19, 20}, J_3 is {16, 17, 18, 21, 22, 23}, J_4 is {18, 19, 20, 23, 24, 25}.

The other segment result as shown in Figure 4. Figure 4(a) to Figure 4(d) describes the function distributions of the source domains and the target domain. Figure 4(e) to Figure 4(g) describes the segment results of the source domains and the target domain, the different colors mean the different segment area of the input data, as we can see, 3 source domains and the target domain are respectively segmented into 4, 6, 9 pairwise parts. Figure 4(h) describes that the target domain selects the most similar parts of three source domains, the same color means these groups belong to the same source domain. Figure

4(i) to Figure 4(l) describes the function distributions of the segment results, the color of the different groups corresponding to the color from Figure 4(e) to Figure 4(h). And the prediction error MAE and MSE of these four source domains are shown in Table 3.

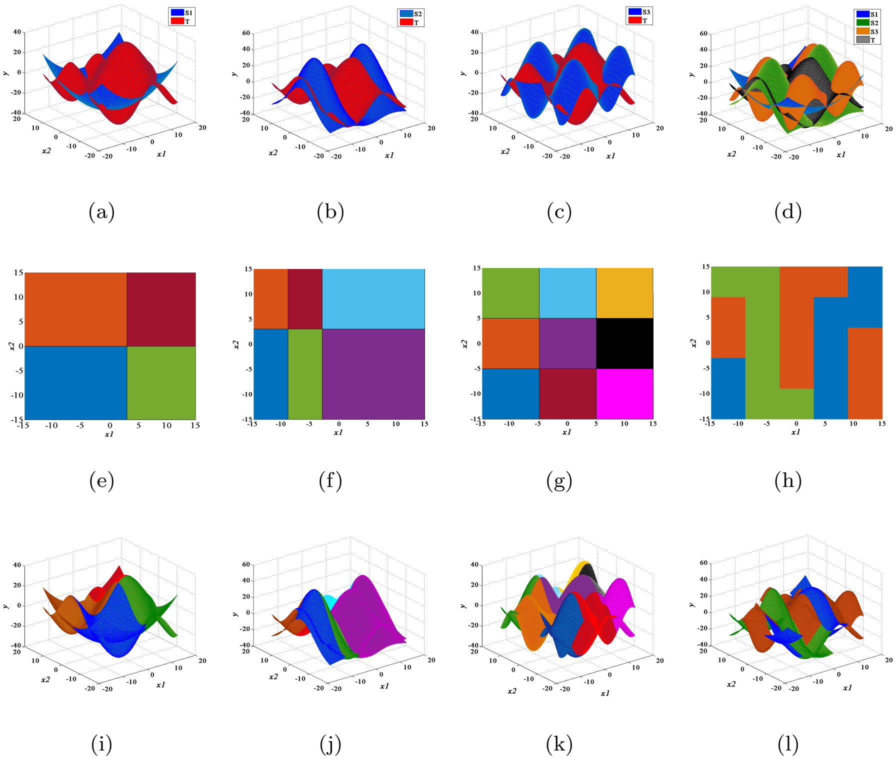


Figure 4: The segment results of the three source domains and the target domain. (a) The distributions of S_1 and T. (b) The distributions of S_2 and T. (c) The distributions of S_3 and T. (d) The distributions of all source domains and target domain. (e) S_1 and T are segmented into 4 pairwise parts. (f) S_2 and T are segmented into 6 pairwise parts. (g) S_3 and T are segmented into 9 pairwise parts. (h) Target domain selects the most similar parts of the three source domains. (i) The segment result of S_1 and T. (j) The segment result of S_2 and T. (k) The segment result of S_3 and T. (l) The segment result of all source domains and target domain.

Table 3: The prediction of 3 source domain

Source	1_{th}	2_{th}	3_{th}	All
MAE	10.65	7.68	6.04	3.09
MSE	200.47	91.92	64.83	17.97

5.1.3. Discussion

The experiment results show that the proposed STPS-MTR method can adaptively segment the source domains and the target domain into different pairwise parts, and each pairwise part can satisfy the condition that the distributions of the source domain and the target domain have an approximately linear relationship. For the multi-source domain transfer learning problem, from Figure 3(e) and Figure 4(l), as we can see, the target domain selects the most similar part of different source domains as the training model to avoid the negative influence between different source domains. The STPS-MTR method has better transfer performance when the number of the different source domains getting larger, the results are shown in Tables 2 and 3.

5.2. The experiments with Real-world datasets

We use the public GP dataset SARCOS, the public UCI dataset UJIIndoorLoc, the public Amazon reviews dataset, and the public SemEval-2014 dataset to evaluate the performance of the proposed STPS-MTR method.

5.2.1. SARCOS Dataset

This dataset relates to an inverse dynamics problem for a seven degrees-of-freedom SARCOS anthropomorphic robot arm. The task requires to map from a 21-dimensional input space (7 joint positions, 7 joint velocities, 7 joint accelerations) to the corresponding 7 joint torques. This dataset is used to test the multi-source domain transfer learning problem that there are large differences in the similarities between the different source domains and the target domain. We consider the seven degrees as seven domains, the first degree as the target domain, and the others as six source domains.

5.2.2. UJIIndoorLoc Dataset

This dataset describes the WiFi Fingerprinting indoor location, it covers three buildings of Universitat Jaume I with 4 floors. The 520 WiFi fingerprint attributes are used as the input data, the location represented by the latitude and longitude is taken as the output data. The latitude is almost the same in the same building, and the longitude is almost the same on the same floor. To test the multi-source domain transfer learning problem that there are large differences in the similarities between the different source domains and the target domain, we choose the same building and the different floor as the domain, the building 1 and the 1 floor as the target domain, the building 1 and the 2, 3, 4 floors as the source domains, the longitude as the output data.

5.2.3. Amazon reviews Dataset

This dataset contains product review text and rating labels taken from Amazon. We select four different categories of products as four domains, clothing (clothing, shoes, and jewelry), grocery (grocery and gourmet food), office (office products), and movies (movies and TV). A vocabulary of 2000 words is defined that occur at least five times at the intersection of the four domains. These words are used to define input data, where every feature is encoded by the number of occurrences of each word. The clothing domain as the target domain, the others as the source domains.

5.2.4. SemEval-2014 Dataset

This dataset contains customer reviews with human-authored annotations identifying the mentioned aspects of the target entities and provides the sentiment polarity of each aspect. We select four different categories of products as four domains, laptops, restaurants, food, and price. A vocabulary of 1000 words is defined that occur at least five times at the intersection of the four domains. These words are used to define input data, where every feature is encoded by the number of occurrences of each word. The laptops domain as the target domain, the others as the source domains.

5.2.5. Experiment Settings

Like other GP models, we assume the mean value of the model to be 0. Therefore, in our experiments, we subtract the mean value of the output feature for both the source domain and the target domain. For each source domain has 500 instances uniformly at random as the training data, likewise, the target domain has 25 instances as the training data and 1000 instances as the testing data. To evaluate the results, we use the standard indicators of MAE and MSE. For the two domains transfer problem, STPS-MTR is compared to GP for regression which is not a transfer learning method [20], TwoStageTrAdaBoostR2 (TS-TR) [1], WDC [31], and AT-GP [2]; for the multi-source domain transfer problem, STPS-MTR is compared to WDC [31], MultiSourceTrAdaBoost (MST) [30] and $TC_{MS}Stack$ [25]. The proposed STPS-MTR method is based on the AT-GP method, thus, in the experiments, we mainly compare the STPS-MTR with $TC_{MS}Stack$ which is also based on the AT-GP method.

For TS-TR and MST, we use the decision tree regression method as the base learner and run all the datasets more than 30 times, and the minimum prediction error values are used as the result. For WDC, we use the source domain to pre-train the model and use the target domain to fine-tune the model. For STPS-MTR, AT-GP, and $TC_{MS}Stack$, we use the Gaussian kernel as the learner, again running all datasets more than 30 times. Here, the minimum value of the optimization function is used as the result. The objective of this overall set of experiments is only to compare the MAE and MSE for different methods. Therefore, the variance of the GP method is not used as a comparison as the above transfer learning methods do not consider a variance when solving prediction tasks.

When the similarity between the source domain and the target domain is relatively low, the source domain and the target domain may be segmented too many pairwise parts. To avoid the overfitting problem, we set the noise bound parameter $\alpha \in [2, 20]$ and the noise item $\delta = 0.03$, the number of data in each pairwise part should be more than 5, otherwise, we will abandon this pairwise part.

5.2.6. Results

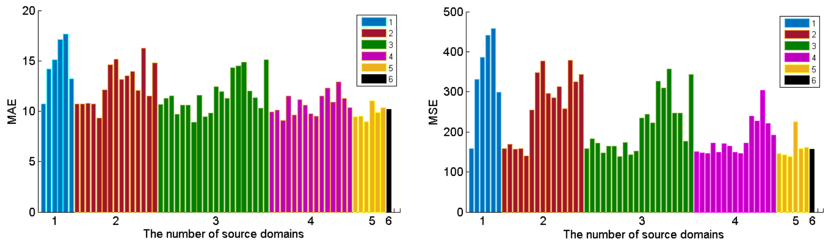
Table 4 shows the results of these experiments. For the SARCOS dataset, it has six source domains and one target domain. According to the number of source domains, we build 6 experimental groups, the number of source domains ranges from 1 to 6, each experimental group consists of the permutation and combination of these 6 source domains, the number of each experimental group is 6, 15, 20, 15, 6, and 1 respectively. Figure 5(a) and Figure 5(b) show the MAE and MSE of STPS-MTR for these experimental groups. And Table 4 shows the average MAE and MSE for 6 experimental groups. In terms of MAE and MSE, STPS-MTR has the best result. When the number of the source is 1, TS-TR and WDC show the negative transfer problem that the prediction error is worse than GP, where GP is not a transfer learning method and it only uses the target domain to train the model. According to the order of MST, WDC, $TC_{MS}Stack$, and STPS-MTR, the best result of these 6 experimental groups is 4, 5, 6, and 5 respectively. But the general trend is that the results get better as the number of different source domains increasing.

For the UJIIndoorLoc dataset, the STPS-MTR also shows the best result of MAE and MSE. And except for STPS-MTR, all the methods suffer the negative transfer problem that the prediction error is worse than GP. According to the order of MST, WDC, $TC_{MS}Stack$, and STPS-MTR, the best result of these 7 experimental groups respectively is F2, F2,3,4, F4, and F2,3,4. WDC and STPS-MTR have the trend that the results get better as the number of different source domains increasing. The STPS-MTR has better transfer performance when the distributions of the source domains and the target domain are significantly different.

For the Amazon reviews dataset, the STPS-MTR still shows the best result of MAE and MSE. And all the methods do not suffer the negative transfer problem. According to the order of MST, WDC, $TC_{MS}Stack$, and STPS-MTR, the best result of these 7 experimental groups respectively is O, G,O,M, G,O, and G,O,M. WDC and STPS-MTR still have the trend that the results get better as the number of different source domains increasing. The other methods do not have this kind of characteristic.

For the SemEval-2014 dataset, the STPS-MTS also shows the best result of MAE and MSE, WDC is the second best result of MAE and MSE. The other methods still suffer the negative transfer problem that the prediction error is worse than GP.

For these four datasets, the Friedman Test [10] is used to prove the statistical significance of these results. The null hypothesis for the test is, H0: there is no significant difference in the prediction results of these comparison methods, H1: there is a significant difference in the prediction results of these comparison methods. The test statistic values of four datasets are 40.47, 47.79, 51.43, and 42.17, respectively. The p-value of the test, returned as a scalar value in the range [0,1], which is the probability of observing a test statistic as extreme as, or more extreme than, the observed value under the null hypothesis. The p-values of these test are 3.46e-08, 9.70e-10, 4.53e-11, and 1.53e-08, respectively, and their values are smaller than 0.01, so the null hypothesis of the test is rejected. The results show that the prediction results of these comparison methods are significantly different.



(a)

(b)

Figure 5: The results of STPS-MTR for the SARCOS. (a) The MAE of STPS-MTR for the SARCOS, (b)The MSE of STPS-MTR for the SARCOS.

Table 4: The comparative of four datasets

SARCOS		GP		TS-TR / MST		WDC		AT-GP / $TC_{MS}Stack$		STPS-MTR	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Number of Source	1	17.96	487.32	24.37	913.59	22.86	821.92	16.81	427.70	**14.68**	**345.44**
	2	17.96	487.32	18.40	552.09	16.35	603.24	14.78	422.71	**12.63**	**263.85**
	3	17.96	487.32	15.40	353.60	14.82	434.65	13.18	311.59	**11.62**	**215.29**
	4	17.96	487.32	15.36	346.89	13.34	320.91	12.57	279.63	**10.71**	**183.34**
	5	17.96	487.32	15.48	345.06	12.96	283.43	11.95	272.02	**9.87**	**162.38**
	6	17.96	487.32	16.42	378.60	14.07	415.78	11.55	286.46	**10.17**	**155.97**
UJIIndoorLoc		GP		TS-TR / MST		WDC		AT-GP / $TC_{MS}Stack$		STPS-MTR	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Source	F2	18.21	599.37	20.52	658.69	20.35	634.28	18.25	600.29	**12.79**	**379.92**
	F3	18.21	599.37	23.57	1059.51	24.22	930.57	18.75	641.70	**13.77**	**427.53**
	F4	18.21	599.37	20.73	899.87	19.55	602.06	16.34	499.37	**12.80**	**390.42**
	F2,3	18.21	599.37	22.46	735.84	21.25	716.04	20.46	644.00	**11.25**	**343.55**
	F2,4	18.21	599.37	22.32	737.55	18.75	557.47	19.35	601.41	**11.81**	**360.41**
	F3,4	18.21	599.37	27.15	1220.67	21.05	702.63	18.27	601.22	**12.45**	**386.53**
	F2,3,4	18.21	599.37	24.36	840.65	18.35	533.94	17.74	559.00	**11.13**	**332.25**
Amazon reviews		GP		TS-TR / MST		WDC		AT-GP / $TC_{MS}Stack$		STPS-MTR	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Source	Grocery(G)	0.94	1.47	0.88	1.23	0.65	0.84	0.93	1.40	**0.65**	**0.82**
	Office(O)	0.94	1.47	0.86	1.19	0.74	0.98	0.90	1.35	**0.71**	**0.95**
	Movies(M)	0.94	1.47	0.87	1.23	0.78	1.12	0.90	1.34	**0.76**	**1.04**
	G,O	0.94	1.47	0.86	1.20	0.58	0.76	0.85	1.22	**0.55**	**0.70**
	G,M	0.94	1.47	0.88	1.24	0.61	0.81	0.88	1.27	**0.57**	**0.71**
	O,M	0.94	1.47	0.87	1.21	0.68	0.88	0.89	1.21	**0.65**	**0.86**
	G,O,M	0.94	1.47	0.88	1.22	0.56	0.70	0.89	1.21	**0.53**	**0.66**
SemEval-2014		GP		TS-TR / MST		WDC		AT-GP / $TC_{MS}Stack$		STPS-MTR	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Source	Restaurants(R)	6.08	61.45	6.36	68.64	6.05	61.71	6.15	62.78	**6.02**	**60.83**
	Food(F)	6.08	61.45	6.30	67.64	5.73	58.04	6.10	61.67	**5.71**	**57.50**
	Price(p)	6.08	61.45	6.24	66.39	5.98	61.25	6.27	63.61	**5.92**	**60.00**
	R,F	6.08	61.45	6.16	63.22	5.82	59.76	5.91	60.56	**4.95**	**44.44**
	R,P	6.08	61.45	5.73	56.89	5.35	51.92	5.46	51.67	**4.99**	**45.56**
	F,P	6.08	61.45	5.86	60.25	5.48	55.83	5.66	53.33	**5.47**	**55.00**
	R,F,P	6.08	61.45	5.63	52.44	5.13	47.37	5.29	48.06	**4.75**	**42.50**

5.2.7. Discussion

From the above results, for the two domain transfer learning problems, the STPS-MTR method has smaller prediction error than other methods, it can be adapted to the situation that the distributions of the source domains and the target domain are significantly different, even if the source domain may not be suitable for the transfer that the AT-GP method and the other methods occur the negative transfer problem when the similarity between the source domain and the target domain is relatively low. For the multiple source domains transfer learning problems, the STPS-MTR method also has the best transfer performance, the $TC_{MS}Stack$ and other methods suffer the negative influence problem that the transfer performance becomes worse as the number of source domains increases, but the transfer performance of STPS-MTR still has the improvement tend as the number of different source domains increases. The WDC and STPS-MTR methods have the trend that the results get better as the number of different source domains increasing, but the WDC method is mainly concerned with text datasets, the transfer performance on Amazon Review and SemEval-2014 datasets is better than other datasets, and the STPS-MTR method can work well on these four datasets. We conclude that the STPS-MTR method can effectively extract the transfer knowledge even when the similarity between the source domain and the target domain is relatively low and it can avoid the negative influence when the distributions of different source domains and the target domain are significantly different.

Table 5 shows the best result for each number of the source domains of Figure 5. For the multi-source domain transfer problem, although the STPS-MTR method has the trend that the results get better as the number of source domains increases and the transfer performance with the multiple source domains is better than the transfer performance with the single source domain, from the Table 5 we can see, the best result is the number of 3 source domains when we select the 2,4,7 degree as the source domains. We can conclude that the STPS-MTR method can improve the transfer performance when the number of source domains increases and it can make sure the transfer performance with the multiple source domains is better than the transfer performance with the single source domain, but it still depends on the source domain selection, if we select the source domain is not suit for transfer, it also will affect the transfer performance.

Table 5: The best result of the STPS-MTR for SARCOS

Source	2	2,7	2,4,7	2,3,4,7	2,4,5,6,7	2,3,4,5,6,7
MAE	10.71	9.31	**8.93**	9.12	8.94	10.17
MSE	158.31	139.42	**138.30**	145.26	138.62	155.97

6. Conclusions and further study

In this paper, we focus on the multi-source domain transfer regression problem when the transfer performance with the single source domain is still not good and there are significant differences in the similarities between the different source domains and the target domain. As demonstrated through a series of experiments, the proposed STPS-MTR method has two advantages. The first is the STPS-MTR method effectively extracts the transfer knowledge when the similarity between the source domain and the target domain is relatively low. The second is the STPS-MTR method overcomes the negative influence between different source domains. And a comparison between the STPS-MTR method and the other existing methods such as TS-TR, AT-GP, MST, WDC, and $TC_{MS}Stack$ also shows that the proposed STPS-MTR method can better estimate the prediction values in the target domain and has better transfer learning performance, particularly when there are significant differences in the similarities between the different source domains and the target domain and all the similarities between the different source domains and the target domain are not relatively high. Although the STPS-MTR method also has better performance than other existing methods when the number of the different source domains getting larger, the transfer performance of STPS-MTR still depends on similarities between the different source domains and the target domain rather than the number of the different source domains getting larger.

This study concerns on transfer learning on homogenous domains, that is, the source domains and the target domain share the same feature space, and the proposed STPS-MTR method is based on the AT-GP method. In future studies, we will focus on more general source-target segment modeling which can apply to any transfer learning method, and we will consider the transfer learning on heterogeneous domains, where the source domains and the target domain have different feature space. We will further develop new feature mapping methods to capture the similarity between different source domains and the target domain. In addition, real-world applications of the proposed multi-source transfer learning methods will be developed.

Acknowledgements

This work was supported by the Australian Research Council (ARC) under Discovery Grant DP170101632, and the International Cooperation Project of Shanghai Scientific Committee Grant 18510760300.

References

[1] Al-Stouhi, S., Reddy, C. K., 2011. Adaptive boosting for transfer learning using dynamic updates. In: Proc. Conf. Machine Learning and Knowledge Discovery in Databases. Athens, Greece, pp. 60–75.

[2] Cao, B., Pan, S. J., Zhang, Y., Yeung, D.-Y., Yang, Q., 2010. Adaptive transfer learning. In: Proc. 24th Conf. Artificial Intelligence. Vol. 2. Atlanta, Georgia, USA, pp. 407–412.

[3] Cawley, G. C., Talbot, N. L., 2010. On over-fitting in model selection and subsequent selection bias in performance evaluation. Journal of Machine Learning Research 11, 2079–2107.

[4] Chattopadhyay, R., Sun, Q., Fan, W., Davidson, I., Panchanathan, S., Ye, J., 2012. Multisource domain adaptation and its application to early detection of fatigue. ACM Transactions on Knowledge Discovery from Data 6 (4), 18–28.

[5] Courty, N., Flamary, R., Habrard, A., Rakotomamonjy, A., 2017. Joint distribution optimal transportation for domain adaptation. In: Advances in Neural Information Processing Systems. pp. 3730–3739.

[6] Ding, Z., Shao, M., Fu, Y., 2016. Incomplete multisource transfer learning. IEEE Transactions on Neural Networks and Learning Systems 29 (2), 310–323.

[7] Dinh, V. Q., Munir, F., Azam, S., Yow, K.-C., Jeon, M., 2020. Transfer learning for vehicle detection using two cameras with different focal lengths. Information Sciences 514, 71–87.

[8] Duan, L., Tsang, I. W., Xu, D., Chua, T.-S., 2009. Domain adaptation from multiple sources via auxiliary classifiers. In: Proc. 26th Int. Conf. Machine Learning. pp. 289–296.

[9] Duan, L., Tsang, I. W., Xu, D., Maybank, S. J., 2009. Domain transfer svm for video concept detection. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. Miami, FL, USA, pp. 1375–1381.

[10] Friedman, M., 1940. A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics 11 (1), 86–92.

[11] Gress, A., Davidson, I., 2017. Transfer regression via pairwise similarity regularization. In: Proc. Int. Conf. Machine Learning, arXiv:1712.08855.

[12] Hoffman, J., Mohri, M., Zhang, N., 2017. Multiple-source adaptation for regression problems. In: Proc. Int. Conf. Machine Learning, arXiv:1711.05037.

[13] Lei, C., Dai, H., Yu, Z., Li, R., 2020. A service recommendation algorithm with the transfer learning based matrix factorization to improve cloud security. Information Sciences 513, 98–111.

[14] Long, M., Wang, J., Ding, G., Sun, J., Yu, P. S., 2013. Transfer feature learning with joint distribution adaptation. In: Proc. IEEE Int. Conf. Computer Vision. pp. 2200–2207.

[15] Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., Zhang, G., 2015. Transfer learning using computational intelligence: A survey. Knowledge Based System 80, 14–23.

[16] Lu, J., Zuo, H., Zhang, G., 2019. Fuzzy multiple-source transfer learning. IEEE Transactions on Fuzzy Systems.

[17] Mansour, Y., Mohri, M., Rostamizadeh, A., 2009. Domain adaptation with multiple sources. In: Advances in neural information processing systems. pp. 1041–1048.

[18] Mansour, Y., Mohri, M., Rostamizadeh, A., 2009. Multiple source adaptation and the r'enyi divergence. In: Proc. 25th Conf. Uncertainty in Artificial Intelligence. pp. 367–374.

[19] Pan, S. J., Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22 (10), 1345–1359.

[20] Rasmussen, C. E., 2003. Gaussian processes in machine learning. In: Proc. Advanced Lectures on Machine Learning. Springer, Berlin, Heidelberg, pp. 63–71.

[21] Rosenstein, M. T., Marx, Z., Kaelbling, L. P., Dietterich, T. G., 2005. To transfer or not to transfer. In: NIPS 2005 Workshop on Transfer Learning. Vol. 898. pp. 1–4.

[22] Shao, Q., Wu, Y., 2005. A consistent procedure for determining the number of clusters in regression clustering. Journal of Statistical Planning and Inference 135 (2), 461–476.

[23] Sun, Q., Chattopadhyay, R., Panchanathan, S., Ye, J., 2011. A two-stage weighting framework for multi-source domain adaptation. In: Advances in neural information processing systems. pp. 505–513.

[24] Torralba, A., Efros, A. A., 2011. Unbiased look at dataset bias. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. Colorado Springs, CO, USA, pp. 1521–1528.

[25] Wei, P., Sagarna, R., Ke, Y., Ong, Y.-S., Goh, C.-K., 2017. Source-target similarity modelings for multi-source transfer gaussian process regression. In: Proc. 34th Int. Conf. Machine Learning. pp. 3722–3731.

[26] Wu, F., Huang, Y., 2016. Sentiment domain adaptation with multiple sources. In: Proc. 54th Annual Meeting of the Association for Computational Linguistics. pp. 301–310.

[27] Xu, Q., Yang, Q., 2011. A survey of transfer and multitask learning in bioinformatics. Journal of Computing Science and Engineering 5 (3), 257–268.

[28] Xu, Z., Sun, S., 2012. Multi-source transfer learning with multi-view adaboost. In: Proc. Int. Conf. Neural Information Processing. pp. 332–339.

[29] Yang, J., Yan, R., Hauptmann, A. G., 2007. Cross-domain video concept detection using adaptive svms. In: Proc. 15th Int. Conf. Multimedia. Augsburg, Germany, pp. 188–197.

[30] Yao, Y., Doretto, G., 2010. Boosting for transfer learning with multiple sources. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. pp. 1855–1862.

[31] Zeng, J., Su, J., Wen, H., Liu, Y., Xie, J., Yin, Y., Zhao, J., 2018. Multi-domain neural machine translation with word-level domain context discrimination. In: Proc. 2018 Conf. Empirical Methods in Natural Language Processing. pp. 447–457.

[32] Zhang, L., Yang, J., Zhang, D., 2017. Domain class consistency based transfer learning for image classification across domains. Information Sciences 418, 242–257.

[33] Zhao, H., Zhang, S., Wu, G., Gordon, G. J., et al., 2018. Multiple source domain adaptation with adversarial learning. In: Proc. 6th Int. Conf. Learning Representations.

[34] Zuo, H., Zhang, G., Pedrycz, W., Behbood, V., Lu, J., 2017. Fuzzy regression transfer learning in takagi–sugeno fuzzy models. IEEE Transactions on Fuzzy Systems 25 (6), 1795–1807.

[35] Zuo, H., Zhang, G., Pedrycz, W., Lu, J., 2019. Domain selection of transfer learning in fuzzy prediction models. In: in Proc. 2019 IEEE Int. Conf. Fuzzy Systems. pp. 1–6.