SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2023, Vol. 66 132104:1–132104:10
https://doi.org/10.1007/s11432-020-3526-4

A convergence algorithm for graph co-regularized transfer learning

Zuyuan YANG[1], Naiyao LIANG[1*], Zhenni LI[1,2] & Shengli XIE[1,3]

[1]*Guangdong Key Laboratory of IoT Information Technology, School of Automation, Guangdong University of Technology, Guangzhou* 510006*, China;*
[2]*Key Laboratory of iDetection and Manufacturing-IoT, Ministry of Education, Guangzhou* 510006*, China;*
[3]*Guangdong-HongKong-Macao Joint Laboratory for Smart Discrete Manufacturing, Guangzhou* 510006*, China*

Received 2 November 2020/Revised 11 November 2021/Accepted 4 March 2022/Published online 3 February 2023

Abstract Transfer learning is an important technology in addressing the problem that labeled data in a target domain are difficult to collect using extensive labeled data from the source domain. Recently, an algorithm named graph co-regularized transfer learning (GTL) has shown a competitive performance in transfer learning. However, its convergence is affected by the used approximate scheme, degenerating learned results. In this paper, after analyzing convergence conditions, we propose a novel update rule using the multiplicative update rule and develop a new algorithm named improved GTL (IGTL) with a strict convergence guarantee. Moreover, to prove the convergence of our method, we design a special auxiliary function whose value is intimately related to that of the objective function. Finally, the experimental results on the synthetic dataset and two real-world datasets confirm that the proposed IGTL is convergent and performs better than the compared methods.

Keywords transfer learning, convergence analysis, non-negative matrix factorization, multiplicative update algorithm, optimization

Citation Yang Z Y, Liang N Y, Li Z N, et al. A convergence algorithm for graph co-regularized transfer learning. Sci China Inf Sci, 2023, 66(3): 132104, https://doi.org/10.1007/s11432-020-3526-4

1 Introduction

In some emerging real-world applications, labeled data cannot be easily collected from a target task, affecting the performance of standard machine learning methods. One promising way is to leverage available labeled source data from some related but different domains. For such a cross-domain knowledge transfer purpose, transfer learning has attracted increasing research interest in recent years [1, 2]. Moreover, it has been applied to a wide range of research fields, such as computer vision [3], emotion recognition [4], 5G technology [5], and aerodynamic missile design [6].

Generally, the distribution of data in different domains is quite complex. These complexities pose a great challenge in leveraging knowledge from the source domain. To solve this problem, many efforts have been made to develop transfer learning methods. For instance, a kernel learning-based framework was introduced in [7] to minimize the distribution mean discrepancy and distribution scatter discrepancy between the source and target domains. An attractive deep neural network framework is proposed to extract deep features for the domain adaptation task [8]. Moreover, a useful method introduced in [9] focuses on addressing the heterogeneous domain adaptation problem. With the growth of data, transfer learning approaches employing dimension reduction have become popular [10, 11]. Among them, the methods based on non-negative matrix factorization (NMF) play important parts [12, 13]. For example, the work in [14] proposed a general transfer learning framework to simultaneously explore shared and distinct concepts among all the domains. In [15], an attractive method called graph co-regularized transfer learning (GTL) was introduced to preserve the geometric structure in each domain and simultaneously alleviate the effect of the negative transfer. Although GTL has achieved competitive performance in

[*] Corresponding author (email: naiyaogdut@aliyun.com)

transfer learning, its convergence is affected by the approximate scheme used, restricting the application of this method.

To be specific, the convergence of GTL is largely affected by one variable that needs to be always updated in a certain circumstance. However, this condition may not be satisfied due to the used approximation scheme in the optimization. In reality, the convergence of machine learning algorithms determines not only their good performance but also their successful applications [16–19]. In this paper, we systemically analyze convergence conditions and propose a new algorithm, named improved GTL (IGTL), by employing the multiplicative update rule. Lastly, experimental results on the synthetic dataset and real-world datasets verify that IGTL is convergent and obtains much better performance than GTL. This paper has the following main contributions.

- A novel method with a convergence guarantee is proposed for transfer learning, which addresses the convergence problem of the popular method GTL.
- To prove the convergence of the proposed method with multiple variables, we employ an upper bound-based scheme. To implement the upper bound, we design an auxiliary function whose value is no less than the original objective function and equals the latter on the optimal point.

2 GTL

NMF is a popular technology in data analysis due to its potential to obtain meaningful features of non-negative data [20–22]. For a non-negative data matrix $\boldsymbol{X} \in \mathbb{R}_+^{m\times n}$, NMF factorizes the data matrix into two non-negative matrices $\boldsymbol{U} \in \mathbb{R}_+^{m\times c}$ and $\boldsymbol{V} \in \mathbb{R}_+^{n\times c}$, where the rank of $\boldsymbol{U}$ and $\boldsymbol{V}$ is generally less than the rank of $\boldsymbol{X}$ (i.e., $c < \min\{m, n\}$). Mathematically, the standard NMF with Frobenius norm is represented by

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}\|_F^2 \;\; \text{s.t.}\, \boldsymbol{U} \succeq 0,\; \boldsymbol{V} \succeq 0, \tag{1}$$

where $\succeq$ denotes the component-wise $\geqslant$.

Due to the effectiveness of the model (1), it is further extended to the domain adaptation. In domain adaptation scenarios, a typical assumption is that different domains share common latent factors [10, 12, 14]. Let the data matrix $\boldsymbol{X}_s \in \mathbb{R}_+^{m\times n_s}$ denote the source domain and the data matrix $\boldsymbol{X}_t \in \mathbb{R}_+^{m\times n_t}$ denote the target domain, which can also be represented by $\{\boldsymbol{X}_\pi\}_{\pi\in\Pi}$ and $\Pi = \{s, t\}$. Following the idea of sharing the common latent factors, one of the representation formulations in the NMF setting is to share the basis matrix $\boldsymbol{U} \in \mathbb{R}_+^{m\times c}$ across domains [15]:

$$\min_{\boldsymbol{U},\boldsymbol{V}_\pi} \sum_{\pi\in\Pi} \|\boldsymbol{X}_\pi - \mathbf{U}\boldsymbol{V}_\pi^{\mathrm{T}}\|_F^2 \;\; \text{s.t.}\, \boldsymbol{U} \succeq 0,\; \boldsymbol{V}_\pi \succeq 0,\; \forall\pi \in \Pi, \tag{2}$$

where $\boldsymbol{V}_\pi \in \mathbb{R}^{n_\pi\times c}$ is the coefficient matrix of the π domain.

Considering the effect of the negative transfer, recent studies showed that GTL [15] alleviates the negative transfer by preserving the geometric structure of the data space and feature space simultaneously, i.e., $\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}_\pi^u\boldsymbol{U}^{\mathrm{T}})$ and $\mathrm{tr}(\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{L}_\pi^v\boldsymbol{V}_\pi)$, where $\boldsymbol{L}_\pi^v \in \mathbb{R}^{n_\pi\times n_\pi}$ and $\boldsymbol{L}_\pi^u \in \mathbb{R}^{m\times m}$ are the example and feature graph Laplacian matrices of the π domain, respectively. Moreover, it uses an orthogonal constraint to handle the trivial solution problem, i.e., $\|\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi - \boldsymbol{I}\|_F^2$, where $\boldsymbol{I} \in \mathbb{R}^{c\times c}$ is an identity matrix. The overall objective function of GTL is as follows [15]:

$$\begin{aligned} &\min_{\boldsymbol{U},\boldsymbol{V}_\pi} \sum_{\pi\in\Pi} \|\boldsymbol{X}_\pi - \boldsymbol{U}\boldsymbol{V}_\pi^{\mathrm{T}}\|_F^2 + \frac{\sigma}{2}\sum_{\pi\in\Pi} \|\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi - \boldsymbol{I}\|_F^2 + \lambda\sum_{\pi\in\Pi} \mathrm{tr}(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{L}_\pi^u\boldsymbol{U}) + \gamma\sum_{\pi\in\Pi} \mathrm{tr}(\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{L}_\pi^v\boldsymbol{V}_\pi) \\ &\text{s.t.}\, \boldsymbol{U} \succeq 0,\; \boldsymbol{V}_\pi \succeq 0,\; \forall\pi \in \Pi. \end{aligned} \tag{3}$$

The corresponding update rules for $\boldsymbol{U}$ and $\boldsymbol{V}_\pi$ are shown as

$$\boldsymbol{U} \leftarrow \boldsymbol{U} \odot \frac{\sum_{\pi\in\Pi}(\boldsymbol{X}_\pi\boldsymbol{V}_\pi^{\mathrm{T}} + \lambda\boldsymbol{W}_\pi^u\boldsymbol{U})}{\sum_{\pi\in\Pi}(\boldsymbol{U}\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi + \lambda\boldsymbol{D}_\pi^u\boldsymbol{U})}, \tag{4}$$

$$\boldsymbol{V}_\pi \leftarrow \boldsymbol{V}_\pi \odot \frac{\boldsymbol{X}_\pi^{\mathrm{T}}\boldsymbol{U} + \gamma\boldsymbol{W}_\pi^v\boldsymbol{V}_\pi + \sigma\boldsymbol{V}_\pi}{\boldsymbol{V}_\pi\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} + \gamma\boldsymbol{D}_\pi^v\boldsymbol{V}_\pi + \sigma\boldsymbol{V}_\pi\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi}, \tag{5}$$

Algorithm 1 Proposed IGTL algorithm

```
Input: data matrices $\{\boldsymbol{X}_\pi\}_{\pi\in\Pi}$, label matrix of the source domain $\boldsymbol{Y}_s$, number of the nearest neighbors $\alpha$, and trade-off parameters $\lambda$, $\gamma$, and $\sigma$;
Calculate graph Laplacian matrices $\boldsymbol{L}_\pi^v$ and $\boldsymbol{L}_\pi^u$, $\forall\pi\in\Pi$;
Initialize $\boldsymbol{U}$ and $\boldsymbol{V}_\pi$, $\forall\pi\in\Pi$;
Set the iteration number $T$;
for $i=1,2,\ldots,T$ do
    Update $\boldsymbol{U}$ by (4);
    for each $\pi\in\Pi$ do
        update $\boldsymbol{V}_\pi$ by (7);
    end for
end for
Output: label matrix of the target domain $\boldsymbol{Y}_t$.
```

where $\odot$ denotes the component-wise product; $\boldsymbol{W}_\pi^v \in \mathbb{R}^{n_\pi\times n_\pi}$ and $\boldsymbol{W}_\pi^u \in \mathbb{R}^{m\times m}$ are the example and feature similarity matrices in the π domain, respectively; $\boldsymbol{D}_\pi^v \in \mathbb{R}^{n_\pi\times n_\pi}$ and $\boldsymbol{D}_\pi^u \in \mathbb{R}^{m\times m}$ are the diagonal matrices calculated by $\boldsymbol{W}_\pi^v$ and $\boldsymbol{W}_\pi^u$, respectively; $\boldsymbol{L}_\pi^v = \boldsymbol{W}_\pi^v - \boldsymbol{D}_\pi^v$ and $\boldsymbol{L}_\pi^u = \boldsymbol{W}_\pi^u - \boldsymbol{D}_\pi^u$.

The convergence of the method above is largely affected by the update rule for $\boldsymbol{V}_\pi$ in (5), and it requires $(\boldsymbol{V}_\pi\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{ij} \geqslant v_{ij}((\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{jj} + v_{jj}^2 - 1)$, where v_{ij} represents the (i,j)-entry of $\boldsymbol{V}_\pi$ for simplicity. Based on the used approximate condition, it holds that $(\boldsymbol{V}_\pi\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{ij} \geqslant v_{ij}(\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{jj} \geqslant v_{ij}((\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{jj} + v_{jj}^2 - 1)$. Furthermore, the convergence requirement meets under the condition $v_{jj}^2 \leqslant 1$. However, this condition cannot be easily satisfied using the update rule (5) in each iteration, as the strict orthogonality constraint is replaced by an approximate method for the convenience of optimization. Actually, the approximation scheme only makes $\sum_i v_{ij}^2 = (\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{jj}$ equal 1 approximatively, which means that v_{jj}^2 is less than 1 approximatively. In some cases, it may hold that $v_{jj}^2 > 1$, especially when v_{jj}^2 is dominant in $\sum_i v_{ij}^2$. This condition brings the convergence risk, restricting the performance of the method introduced above [16–19].

3 IGTL

In this section, we propose an IGTL algorithm with a strict convergence guarantee. First, we define a matrix $\boldsymbol{K}_\pi$ and its element is given by

$$k_{ij} = \sigma v_{ij}^2 + \frac{v_{ij}}{1-v_{ij}}(\boldsymbol{X}_\pi^{\mathrm{T}}\boldsymbol{U})_{ij}. \tag{6}$$

In NMF-based methods, the multiplicative update rules can balance the speed and ease of implementation for solving the optimization problem [23]. To conveniently solve the model, we solve $\boldsymbol{V}_\pi$ using the multiplicative update rule. The new update rule of $\boldsymbol{V}_\pi$ can be constructed by adding $\boldsymbol{K}_\pi$ to the original update rule (5):

$$\boldsymbol{V}_\pi \leftarrow \boldsymbol{V}_\pi \odot \frac{\boldsymbol{X}_\pi^{\mathrm{T}}\boldsymbol{U} + \gamma\boldsymbol{W}_\pi^v\boldsymbol{V}_\pi + \sigma\boldsymbol{V}_\pi + \boldsymbol{K}_\pi}{\boldsymbol{V}_\pi\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} + \gamma\boldsymbol{D}_\pi^v\boldsymbol{V}_\pi + \sigma\boldsymbol{V}_\pi\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi + \boldsymbol{K}_\pi}, \tag{7}$$

where the initialization condition of $\boldsymbol{V}_\pi$ is $0 < v_{ij} < 1$. The update rule (7) and the above initialization condition ensure that each element in $\boldsymbol{V}_\pi$ satisfies $v_{ij} < 1$ in each iteration (see (17) and the corresponding proofs).

Then, the new algorithm is developed by substituting (5) with the new update rule (7). The complete algorithm of our IGTL is presented in Algorithm 1.

For the proposed algorithm, the convergence cannot be easily analyzed directly, as each updated matrix has multiple elements, which are viewed as multiple variables. Here, we employ an upper bound-based scheme. Regarding the upper bound, we design an auxiliary function whose value is no less than the original objective function and equals the latter on the optimal point. We have the following theorem about convergence.

Theorem 1. The objective function (3) is non-increasing under the updating rules of $\boldsymbol{U}$ and $\boldsymbol{V}_\pi$ in (4) and (7).

Proof. To prove Theorem 1, we first analyze the non-increasing property of the objective function on $\boldsymbol{V}_\pi$. According to the model (3), the optimization problem with respect to $\boldsymbol{V}_\pi$ is

$$J_{\boldsymbol{V}} = \|\boldsymbol{X}_\pi - \mathbf{U}\boldsymbol{V}_\pi^{\mathrm{T}}\|_F^2 + \frac{\sigma}{2}\|\boldsymbol{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi - \boldsymbol{I}\|_F^2 + \gamma\mathrm{tr}(\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{L}_\pi^v\boldsymbol{V}_\pi). \tag{8}$$

For any element v_{ij} in $\boldsymbol{V}_\pi$, we use G_{ij} as the part of $J_{\boldsymbol{V}}$ related to v_{ij}. As analyzed in [23], optimizing G_{ij} can be changed to the minimization of an auxiliary function of G_{ij}. Then, we define the following function for G_{ij}:

$$A(v,v_{ij}) = G_{ij}(v_{ij}) + G'_{ij}(v_{ij})(v-v_{ij}) + \frac{(\boldsymbol{V}_\pi \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} + \gamma \boldsymbol{D}^v_\pi \boldsymbol{V}_\pi + \sigma \boldsymbol{V}_\pi \boldsymbol{V}^{\mathrm{T}}_\pi \boldsymbol{V}_\pi + \boldsymbol{K}_\pi)_{ij}}{v_{ij}}(v-v_{ij})^2. \tag{9}$$

To prove that Eq. (9) is an auxiliary function of G_{ij}, two conditions, i.e., $A(v,v) = G_{ij}(v)$ and $A(v,v_{ij}) \geqslant G_{ij}(v)$ [23], need to be proven. Clearly, $A(v,v) = G_{ij}(v)$ holds. As for proving $A(v,v_{ij}) \geqslant G_{ij}(v)$, we first give the following first-order and second-order derivatives of G_{ij} with respect to v_{ij}:

$$\begin{aligned} G'_{ij} &= 2(\boldsymbol{V}_\pi \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} + \gamma \boldsymbol{D}^v_\pi \boldsymbol{V}_\pi + \sigma \boldsymbol{V}_\pi \boldsymbol{V}^{\mathrm{T}}_\pi \boldsymbol{V}_\pi)_{ij} - 2(\boldsymbol{X}^{\mathrm{T}}_\pi \boldsymbol{U} + \gamma \boldsymbol{W}^v_\pi \boldsymbol{V}_\pi + \sigma \boldsymbol{V}_\pi)_{ij} \\ &= 2(\boldsymbol{V}_\pi \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} + \gamma \boldsymbol{D}^v_\pi \boldsymbol{V}_\pi + \sigma \boldsymbol{V}_\pi \boldsymbol{V}^{\mathrm{T}}_\pi \boldsymbol{V}_\pi)_{ij} - 2(\boldsymbol{X}^{\mathrm{T}}_\pi \boldsymbol{U} + \gamma \boldsymbol{W}^v_\pi \boldsymbol{V}_\pi + \sigma \boldsymbol{V}_\pi)_{ij} + k_{ij} - k_{ij}, \end{aligned} \tag{10}$$

$$G''_{ij} = 2((\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})_{jj} + \gamma(\boldsymbol{D}^v_\pi - \boldsymbol{W}^v_\pi)_{ii}) + 2\sigma((\boldsymbol{V}_\pi \boldsymbol{V}^{\mathrm{T}}_\pi)_{ii} + (\boldsymbol{V}^{\mathrm{T}}_\pi \boldsymbol{V}_\pi)_{jj} + v^2_{ij} - 1). \tag{11}$$

We substitute (10) and (11) into the Taylor series expansion of G_{ij}, i.e., $G_{ij}(v) = G_{ij}(v_{ij}) + G'_{ij}(v_{ij})(v - v_{ij}) + \frac{1}{2}G''_{ij}(v_{ij})(v-v_{ij})^2$, and compare it to (9). The results show that proving $A(v,v_{ij}) \geqslant G_{ij}(v)$ amounts to proving the following inequality:

$$\begin{aligned} &\frac{(\boldsymbol{V}_\pi \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} + \gamma \boldsymbol{D}^v_\pi \boldsymbol{V}_\pi + \sigma \boldsymbol{V}_\pi \boldsymbol{V}^{\mathrm{T}}_\pi \boldsymbol{V}_\pi + \boldsymbol{K}_\pi)_{ij}}{v_{ij}} \\ &\quad \geqslant (\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})_{jj} + \gamma(\boldsymbol{D}^v_\pi - \boldsymbol{W}^v_\pi)_{ii} + \sigma((\boldsymbol{V}_\pi \boldsymbol{V}^{\mathrm{T}}_\pi)_{ii} + (\boldsymbol{V}^{\mathrm{T}}_\pi \boldsymbol{V}_\pi)_{jj} + v^2_{ij} - 1). \end{aligned} \tag{12}$$

For ease of comparison, the inequality (12) is rewritten as

$$L_1 + \gamma L_2 + \sigma L_3 + k_{ij} \geqslant R_1 + \gamma R_2 + \sigma R_3, \tag{13}$$

where $L_1 = (\boldsymbol{V}_\pi \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})_{ij}$, $L_2 = (\boldsymbol{D}^v_\pi \boldsymbol{V}_\pi)_{ij}$, $L_3 = (\boldsymbol{V}_\pi \boldsymbol{V}^{\mathrm{T}}_\pi \boldsymbol{V}_\pi)_{ij}$, $R_1 = v_{ij}(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})_{jj}$, $R_2 = v_{ij}(\boldsymbol{D}^v_\pi - \boldsymbol{W}^v_\pi)_{ii}$, and $R_3 = v_{ij}((\boldsymbol{V}_\pi \boldsymbol{V}^{\mathrm{T}}_\pi)_{ii} + (\boldsymbol{V}^{\mathrm{T}}_\pi \boldsymbol{V}_\pi)_{jj} + v^2_{ij} - 1)$. To prove the inequality (13), we will prove $L_1 \geqslant R_1$, $L_2 \geqslant R_2$, and $\sigma L_3 + k_{ij} \geqslant \sigma R_3$. Using the algebra manipulations, it holds

$$L_1 = (\boldsymbol{V}_\pi \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})_{ij} = \sum_l v_{il}(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})_{lj} \geqslant v_{ij}(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})_{jj} = R_1, \tag{14}$$

$$L_2 = (\boldsymbol{D}^v_\pi \boldsymbol{V}_\pi)_{ij} = \sum_l (\boldsymbol{D}^v_\pi)_{il} v_{lj} \geqslant (\boldsymbol{D}^v_\pi)_{ii} v_{ij} \geqslant v_{ij}(\boldsymbol{D}^v_\pi - \boldsymbol{W}^v_\pi)_{ii} = R_2. \tag{15}$$

Regarding L_3 and R_3, we have the following inequality under the condition $0 \leqslant v_{ij} \leqslant 1$:

$$\begin{aligned} &\sigma L_3 + k_{ij} - \sigma R_3 \\ &= \sigma \sum_{l\neq i, h\neq j} v_{ih}v_{lh}v_{lj} + \sigma v_{ij}(\boldsymbol{V}^{\mathrm{T}}_\pi \boldsymbol{V}_\pi)_{jj} + \sigma(\boldsymbol{V}_\pi \boldsymbol{V}^{\mathrm{T}}_\pi)_{ii} v_{ij} - \sigma v^3_{ij} + k_{ij} - \sigma R_3 \\ &= \sigma \sum_{l\neq i, h\neq j} v_{ih}v_{lh}v_{lj} + \sigma v_{ij}(\boldsymbol{V}^{\mathrm{T}}_\pi \boldsymbol{V}_\pi)_{jj} + \sigma(\boldsymbol{V}_\pi \boldsymbol{V}^{\mathrm{T}}_\pi)_{ii} v_{ij} - \sigma v^3_{ij} + \sigma v^2_{ij} + \frac{v_{ij}}{1-v_{ij}}(\boldsymbol{X}^{\mathrm{T}}_\pi \boldsymbol{U})_{ij} - \sigma R_3 \\ &\geqslant \sigma \sum_{l\neq i, h\neq j} v_{ih}v_{lh}v_{lj} + \sigma v_{ij}(\boldsymbol{V}^{\mathrm{T}}_\pi \boldsymbol{V}_\pi)_{jj} + \sigma(\boldsymbol{V}_\pi \boldsymbol{V}^{\mathrm{T}}_\pi)_{ii} v_{ij} + \frac{v_{ij}}{1-v_{ij}}(\boldsymbol{X}^{\mathrm{T}}_\pi \boldsymbol{U})_{ij} - \sigma R_3 \\ &\geqslant \sigma \sum_{l\neq i, h\neq j} v_{ih}v_{lh}v_{lj} + \sigma v_{ij}(\boldsymbol{V}^{\mathrm{T}}_\pi \boldsymbol{V}_\pi)_{jj} + \sigma(\boldsymbol{V}_\pi \boldsymbol{V}^{\mathrm{T}}_\pi)_{ii} v_{ij} - \sigma R_3 \\ &= \sigma \sum_{l\neq i, h\neq j} v_{ih}v_{lh}v_{lj} + \sigma v_{ij}(1 - v^2_{ij}) \geqslant 0. \end{aligned} \tag{16}$$

From the above inequalities, we find that the inequality (12) holds under the condition $0 \leqslant v_{ij} \leqslant 1$. For ensuring $0 \leqslant v_{ij} \leqslant 1$ in each iteration, given that the initialization of $\boldsymbol{V}_\pi$ is $0 < v_{ij} < 1$, we will guarantee that the numerator is less than the denominator in the update rule for $\boldsymbol{V}_\pi$:

$$\boldsymbol{V}_\pi \odot (\boldsymbol{X}^{\mathrm{T}}_\pi \boldsymbol{U} + \sigma \boldsymbol{V}_\pi + \boldsymbol{K}_\pi + \gamma \boldsymbol{W}^v_\pi \boldsymbol{V}_\pi) \leqslant \boldsymbol{V}_\pi \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} + \sigma \boldsymbol{V}_\pi \boldsymbol{V}^{\mathrm{T}}_\pi \boldsymbol{V}_\pi + \boldsymbol{K}_\pi + \gamma \boldsymbol{D}^v_\pi \boldsymbol{V}_\pi. \tag{17}$$

It amounts to

$$\begin{aligned} &v_{ij}(\boldsymbol{X}_\pi^{\mathrm{T}}\boldsymbol{U})_{ij} + v_{ij}\frac{v_{ij}}{1-v_{ij}}(\boldsymbol{X}_\pi\boldsymbol{U})_{ij} + \sigma v_{ij}^3 + \gamma v_{ij}(\boldsymbol{W}_\pi^v\boldsymbol{V}_\pi)_{ij} \\ &\quad \leqslant \frac{v_{ij}}{1-v_{ij}}(\boldsymbol{X}_\pi\boldsymbol{U})_{ij} + \sigma(\boldsymbol{V}_\pi\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{ij} + (\boldsymbol{V}_\pi\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})_{ij} + \gamma(\boldsymbol{D}_\pi^v\boldsymbol{V}_\pi)_{ij}. \end{aligned} \tag{18}$$

For the convenience of comparison, the inequality (18) is rewritten as

$$B_1 + B_2 + \gamma B_3 \leqslant C_1 + C_2 + \gamma C_3, \tag{19}$$

where $B_1 = v_{ij}(\boldsymbol{X}_\pi^{\mathrm{T}}\boldsymbol{U})_{ij} + v_{ij}\frac{v_{ij}}{1-v_{ij}}(\boldsymbol{X}_\pi\boldsymbol{U})_{ij}$, $B_2 = \sigma v_{ij}^3$, $B_3 = v_{ij}(\boldsymbol{W}_\pi^v\boldsymbol{V}_\pi)_{ij}$, $C_1 = \frac{v_{ij}}{1-v_{ij}}(\boldsymbol{X}_\pi\boldsymbol{U})_{ij}$, $C_2 = \sigma(\boldsymbol{V}_\pi\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{ij} + (\boldsymbol{V}_\pi\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})_{ij}$, and $C_3 = (\boldsymbol{D}_\pi^v\boldsymbol{V}_\pi)_{ij}$. Under the initialization $0 < v_{ij} < 1$ and based on algebra manipulations, we have

$$B_1 = v_{ij}(\boldsymbol{X}_\pi^{\mathrm{T}}\boldsymbol{U})_{ij} + v_{ij}\frac{v_{ij}}{1-v_{ij}}(\boldsymbol{X}_\pi^{\mathrm{T}}\boldsymbol{U})_{ij} = \frac{v_{ij}}{1-v_{ij}}(\boldsymbol{X}_\pi^{\mathrm{T}}\boldsymbol{U})_{ij} = C_1, \tag{20}$$

$$\begin{aligned} C_2 &= \sigma(\boldsymbol{V}_\pi\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{ij} + (\boldsymbol{V}_\pi\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})_{ij} \geqslant \sigma(\boldsymbol{V}_\pi\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{ij} = \sigma\sum_l v_{il}(\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{lj} \\ &\geqslant \sigma v_{ij}(\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{jj} = \sigma v_{ij}\sum_l v_{lj}^2 \geqslant \sigma v_{ij}v_{ij}^2 = \sigma v_{ij}^3 = B_2, \end{aligned} \tag{21}$$

$$\begin{aligned} C_3 &= (\boldsymbol{D}_\pi^v\boldsymbol{V}_\pi)_{ij} = \sum_l d_{il}^v v_{lj} \geqslant d_{ii}^v v_{ij} = \sum_l w_{il}^v v_{ij} \\ &= v_{ij}\sum_l w_{il}^v \geqslant v_{ij}\sum_l w_{il}^v v_{lj} = v_{ij}(\boldsymbol{W}_\pi^v\boldsymbol{V}_\pi)_{ij} = B_3. \end{aligned} \tag{22}$$

Therefore, the inequality (17) and the condition $0 \leqslant v_{ij} \leqslant 1$ hold. Based on the condition $0 \leqslant v_{ij} \leqslant 1$ and (16), the inequality $\sigma L_3 + k_{ij} \geqslant \sigma R_3$ holds. Because $L_1 \geqslant R_1$, $L_2 \geqslant R_2$, and $\sigma L_3 + k_{ij} \geqslant \sigma R_3$ hold, the inequality (12) holds. That is, $A(v, v_{ij}) \geqslant G_{ij}(v)$ holds. As $A(v, v) = G_{ij}(v)$ and $A(v, v_{ij}) \geqslant G_{ij}(v)$ hold, $A(v, v_{ij})$ is an auxiliary function of G_{ij}.

Then, by setting $\frac{\partial A(v,v_{ij})}{\partial v} = 0$, we have

$$v = v_{ij} - \frac{v_{ij}G_{ij}^{'}}{2(\boldsymbol{V}_\pi\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} + \gamma\boldsymbol{D}_\pi^v\boldsymbol{V}_\pi + \sigma\boldsymbol{V}_\pi\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi + \boldsymbol{K}_\pi)_{ij}}. \tag{23}$$

Let v be the updated v_{ij}; then substituting (10) into (23) results in the following update rule for v_{ij}:

$$v_{ij} \leftarrow v_{ij}\frac{(\boldsymbol{X}_\pi^{\mathrm{T}}\boldsymbol{U} + \gamma\boldsymbol{W}_\pi^v\boldsymbol{V}_\pi + \sigma\boldsymbol{V}_\pi + \boldsymbol{K}_\pi)_{ij}}{(\boldsymbol{V}_\pi\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} + \gamma\boldsymbol{D}_\pi^v\boldsymbol{V}_\pi + \sigma\boldsymbol{V}_\pi\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi + \boldsymbol{K}_\pi)_{ij}}. \tag{24}$$

The above update rule is the same as (7). That is, as Eq. (9) is an auxiliary function, G_{ij} (i.e., the objective function (3)) is non-increasing under the update rule (7).

For $\boldsymbol{U}$, for any entry u_{ij} in $\boldsymbol{U}$, let F_{ij} denote the part of the objective function related to u_{ij}. Similarly, we define the following function for F_{ij}:

$$\hat{A}(u, u_{ij}) = F_{ij}(u_{ij}) + F_{ij}'(u_{ij})(u - u_{ij}) + \frac{\sum_{\pi\in\Pi}(\boldsymbol{U}\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi + \lambda\boldsymbol{D}_\pi^u\boldsymbol{U})_{ij}}{u_{ij}}(u - u_{ij})^2. \tag{25}$$

The Taylor series expansion of F_{ij} is $F_{ij}(u) = F_{ij}(u_{ij}) + F_{ij}'(u_{ij})(u - u_{ij}) + \frac{1}{2}F_{ij}''(u_{ij})(u - u_{ij})^2$, and $F_{ij}'' = 2\sum_{\pi\in\Pi}((\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{jj} + \lambda(\boldsymbol{D}_\pi^u - \boldsymbol{W}_\pi^u)_{ii})$. We find that proving $\hat{A}(u, u_{ij}) \geqslant F_{ij}(u)$ amounts to proving the following inequalities:

$$\sum_{\pi\in\Pi}(\boldsymbol{U}\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{ij} = \sum_{\pi\in\Pi}\sum_l u_{il}(\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{lj} \geqslant u_{ij}\sum_{\pi\in\Pi}(\boldsymbol{V}_\pi^{\mathrm{T}}\boldsymbol{V}_\pi)_{jj}, \tag{26}$$

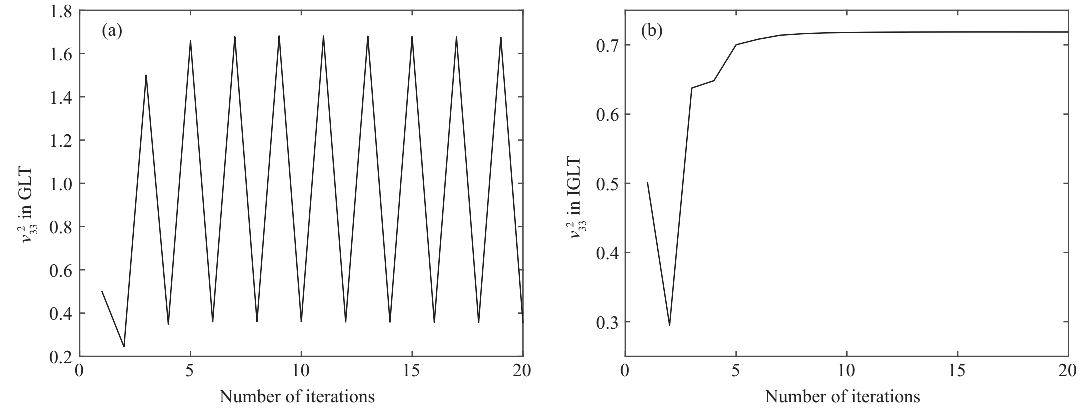

Figure 1 Iterative update curves of v_{33}^2 on the synthetic dataset. (a) GTL; (b) IGTL.

$$\sum_{\pi\in\Pi}(\boldsymbol{D}_\pi^u\boldsymbol{U})_{ij} = \sum_{\pi\in\Pi}\sum_{l}(\boldsymbol{D}_\pi^u)_{il}u_{lj} \geqslant u_{ij}\sum_{\pi\in\Pi}(\boldsymbol{D}_\pi^u)_{ii} \geqslant u_{ij}\sum_{\pi\in\Pi}(\boldsymbol{D}_\pi^u - \boldsymbol{W}_\pi^u)_{ii}. \tag{27}$$

Evidently, $\hat{A}(u,u) = F_{ij}(u)$ holds. Because $\hat{A}(u,u) = F_{ij}(u)$ and $\hat{A}(u,u_{ij}) \geqslant F_{ij}(u)$ hold, $\hat{A}(u,u_{ij})$ is an auxiliary function of $F_{ij}(u)$. By setting $\frac{\partial\hat{A}(u,u_{ij})}{\partial u} = 0$, one can obtain the update rule of u_{ij}, which is identical with (4). Therefore, the objective function (3) is also non-increasing under the update rule (4). This finding completes the proof of Theorem 1.

The computational complexity of IGTL is related to GTL. GTL requires the computational complexity $O(\sum_{\pi\in\Pi}Tc(mn_\pi+m^2+n_\pi^2)+\sum_{\pi\in\Pi}(m^2n_\pi+mn_\pi^2))$, where m is the sample dimension, n_π is the sample number of the π domain, c is the subspace dimension, and T is the iteration number. Moreover, the IGTL requires the extra computational complexity $O(\sum_{\pi\in\Pi}Tcmn_\pi)$ to compute $\boldsymbol{K}_\pi$. Therefore, the overall computational complexity of the proposed approach is $O(\sum_{\pi\in\Pi}Tc(2mn_\pi+m^2+n_\pi^2)+\sum_{\pi\in\Pi}(m^2n_\pi+mn_\pi^2))$.

4 Experiment

4.1 Experiments on the synthetic dataset

We test GTL and IGTL empirically on the synthetic dataset, where the generated data must be the same in the sample dimension to satisfy the domain adaptation setting. For generating the synthetic data, we randomly generate $\boldsymbol{X}_s \in \mathbb{R}_+^{4\times5}$ and $\boldsymbol{X}_t \in \mathbb{R}_+^{4\times5}$ and assume the number of classes $c = 3$ for performing the numerical verification experiments. Regarding the parameter setting, the settings for GTL and IGTL are the same, i.e., $\alpha = 1$, $\gamma = 0.01$, $\sigma = 100$, and $\lambda = 0.1$.

In the beginning, we empirically test the probability of $v_{jj}^2 > 1$. To be specific, we run GTL on 1000 randomly generated synthetic datasets, and the case of $v_{jj}^2 > 1$ happens 505 times; i.e., the probability of $v_{jj}^2 > 1$ is 50.5%. Then, we test the iterative process of GTL and IGTL and randomly generate all the matrices required for the corresponding algorithm, where the data matrices $\boldsymbol{X}_s$ and $\boldsymbol{X}_t$ are displayed as

$$\boldsymbol{X}_s = \begin{pmatrix} 0.3133\ 0.3213\ 0.2629\ 0.1301\ 0.1002 \\ 0.0105\ 0.6818\ 0.1186\ 0.2236\ 0.0249 \\ 0.0749\ 0.3060\ 0.1055\ 0.6880\ 0.2695 \\ 0.9879\ 0.9193\ 0.6957\ 0.2078\ 0.0913 \end{pmatrix}, \tag{28}$$

$$\boldsymbol{X}_t = \begin{pmatrix} 0.7318\ 0.7802\ 0.8326\ 0.5217\ 0.8747 \\ 0.1981\ 0.6758\ 0.2882\ 0.9653\ 0.4083 \\ 0.9158\ 0.6981\ 0.5147\ 0.3305\ 0.4442 \\ 0.4255\ 0.6861\ 0.8759\ 0.9694\ 0.6129 \end{pmatrix}. \tag{29}$$

To verify that $\boldsymbol{V}_t$ given by GTL meets $v_{jj}^2 > 1$, Figure 1 shows the change curves of v_{jj}^2 with the iteration increasing, where the values of v_{33}^2 in $\boldsymbol{V}_t$ are shown. As shown in Figure 1(a), the values of v_{33}^2

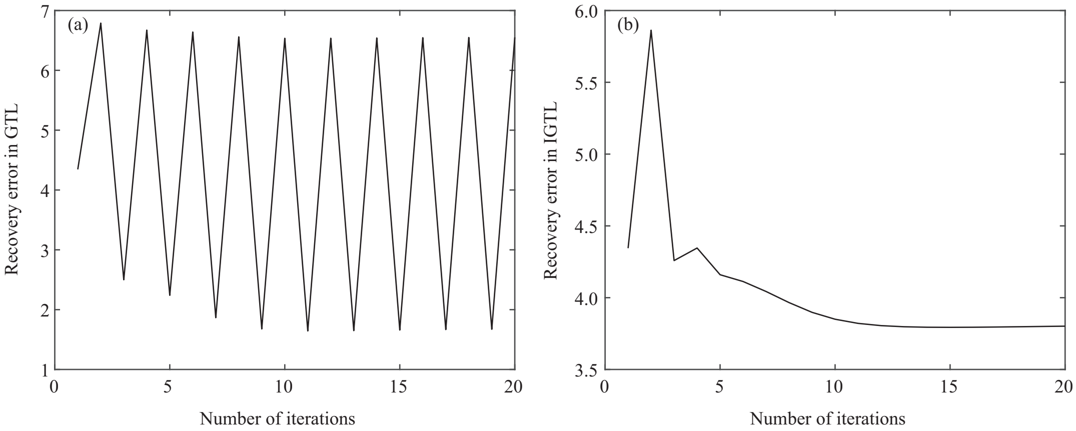

Figure 2 Recovery error curves of $\boldsymbol{X}_t$ on the synthetic dataset. (a) GTL; (b) IGTL.

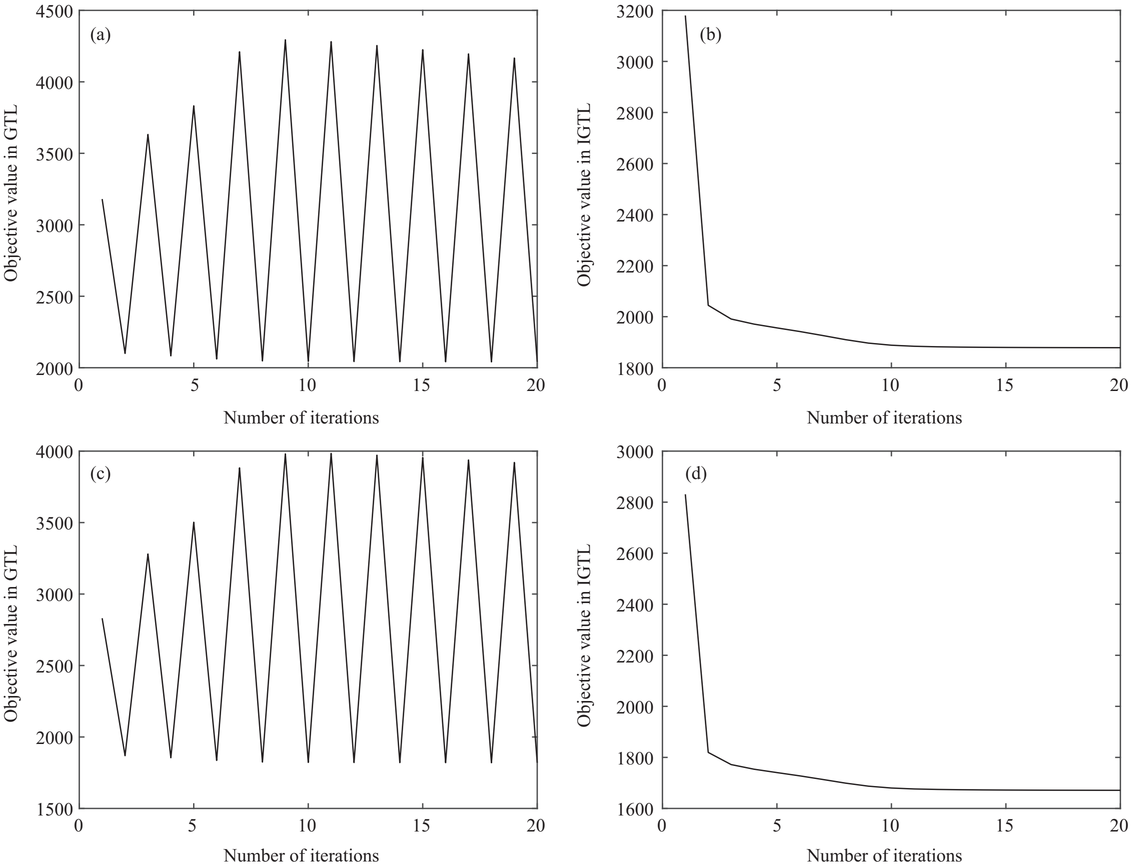

Figure 3 Objective values on the synthetic dataset. (a) GTL while only updating $\boldsymbol{V}_t$; (b) IGTL while only updating $\boldsymbol{V}_t$; (c) GTL while updating all the matrices; (d) IGTL while updating all the matrices.

in GTL are sometimes greater than 1. This result verifies that GTL does not satisfy the required work condition of the utilized balance method. Meanwhile, in Figure 1(b), the values of v_{33}^2 in IGTL are always less than 1, satisfying the required work condition of the utilized balance method. In Figure 2, we show the recovery error curves of $\boldsymbol{X}_t$ under GTL and IGTL. In Figure 2(a), the curve of GTL oscillates with the iteration increasing, while in Figure 2(b), the curve of IGTL is stable with the iteration increasing.

To investigate the change in objective values caused by $\boldsymbol{V}_t$ under different algorithms, we discuss two different cases. The first case is only updating $\boldsymbol{V}_t$, and the second case is updating all the matrices. Regarding the first case, Figures 3(a) and (b) show the objective values with respect to GTL and IGTL, respectively. For the second case, Figures 3(c) and (d) indicate the objective values of GTL and IGTL,

Table 1 Statistics of datasets

Dataset	Domain	Instance	Class	Dimension
COIL20	COIL1	720	20	1024
	COIL2	720	20	1024
Office+Caltech	Webcam	295	10	800
	DSLR	157	10	800
	Caltech	1123	10	800

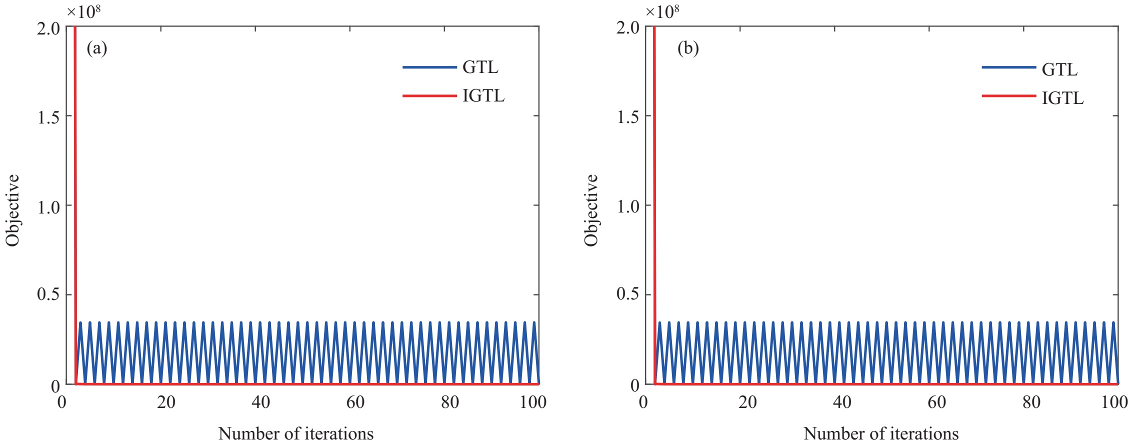

Figure 4 (Color online) Objective values of GTL and IGTL. (a) COIL1→COIL2; (b) COIL2→COIL1.

respectively. The curves of GTL are oscillating, i.e., Figures 3(a) and (c). This is because the approximate convergence of GTL cannot monotonically decrease the objective values. By contrast, the objective values of IGTL vary properly (see Figures 3(b) and (d)). This is reasonable since IGTL can achieve strict convergence.

4.2 Experiments on real-world datasets

4.2.1 *Description of datasets*

We use two real-world domain adaptation datasets to evaluate GTL and IGTL, i.e., the COIL20 dataset and the Office+Caltech dataset. For convenience, the datasets are summarized in Table 1.

COIL20. The dataset contains 20 object categories, and each category has 72 images, totaling 1440 images. Each sample is a grayscale image with the size of 32×32 pixels. Following [24], the dataset is further divided into two domains, COIL1 and COIL2, where COIL1 and COIL2 use different shooting angles. By randomly selecting one dataset as the source domain and the other as the target domain, we construct two cross-domain datasets (source domain→target domain), i.e., COIL1→COIL2 and COIL2→COIL1.

Office+Caltech. This is a popular dataset for domain adaptation [24]. We choose two real-world object domains from the Office database, i.e., Webcam and DSLR. Moreover, adding the Caltech dataset, there are three domains. By randomly selecting two different domains as the source domain and target domain, respectively, six cross-domain datasets are constructed, including Caltech→DSLR, Caltech→Webcam, DSLR→Caltech, DSLR→Webcam, Webcam→Caltech, and Webcam→DSLR.

4.2.2 *Experimental results*

First, we check the convergence of the iterative algorithms on COIL20 while randomly initializing $\boldsymbol{V}_t$. Figures 4(a) and (b) show the results on COIL1→COIL2 and COIL2→COIL1, respectively. As shown in the results, the curves of IGTL monotonically decline while the curves of GTL are oscillating.

Then, we provide the classification performance comparison on two real-world datasets to validate the effectiveness of IGTL. We compare our IGTL with GTL [15] and TCA [25]. In the experiments, the parameter settings of the compared methods are set following the original papers. For IGTL, the parameter setting is $\alpha = 10$, $\gamma = 10$, $\sigma = 100$, and $\lambda = 0.1$, which is the same as those of GTL. The number of the iterations is set to 200. For all the results, we run each algorithm 20 times and report the average classification accuracy and standard deviation.

Table 2 Accuracy (standard deviation) on COIL20 (%) while randomly initializing $\boldsymbol{V}_t$. The values in bold mean the best performance

Method	COIL1→COIL2	COIL2→COIL1	Average
GTL	40.76 (8.09)	40.77 (7.60)	40.77
IGTL	**57.26 (5.07)**	**58.13 (3.85)**	**57.70**

Table 3 Accuracy (standard deviation) on COIL20 (%). $\boldsymbol{V}_t$ in GTL and IGTL are initialized by logistic regression. The values in bold mean the best performance

Method	COIL1→COIL2	COIL2→COIL1	Average
TCA	74.58 (0.00)	77.78 (0.00)	76.18
GTL	77.78 (0.00)	79.72 (0.00)	78.75
IGTL	**78.52 (0.00)**	**80.42 (0.00)**	**79.47**

Table 4 Accuracy (standard deviation) on Office+Caltech (%) while randomly initializing $\boldsymbol{V}_t$. The values in bold mean the best performance

Method	Caltech→DSLR	Caltech→Webcam	DSLR→Caltech	DSLR→Webcam	Webcam→Caltech	Webcam→DSLR	Average
GTL	41.56 (3.85)	42.25 (7.70)	29.96 (1.07)	73.39 (3.24)	30.80 (2.08)	65.57 (4.19)	47.26
IGTL	**42.68 (3.68)**	**46.78 (5.83)**	**31.24 (1.00)**	**78.17 (3.04)**	**31.06 (1.75)**	**70.16 (2.19)**	**50.02**

Table 5 Accuracy (standard deviation) on Office+Caltech (%). $\boldsymbol{V}_t$ in GTL and IGTL are initialized by logistic regression. The values in bold mean the best performance

Method	Caltech→DSLR	Caltech→Webcam	DSLR→Caltech	DSLR→Webcam	Webcam→Caltech	Webcam→DSLR	Average
TCA	44.59 (0.00)	51.19 (0.00)	**32.50 (0.00)**	73.56 (0.00)	**35.71 (0.00)**	70.06(0.00)	51.27
GTL	45.00 (0.37)	52.25 (0.17)	30.93 (0.00)	74.58 (0.00)	34.53 (0.06)	68.79 (0.00)	51.01
IGTL	**48.41 (0.00)**	**59.00 (0.08)**	32.41 (0.00)	**79.68 (0.00)**	34.93 (0.05)	**75.76 (0.00)**	**55.03**

Tables 2 and 3 list the classification results on COIL1→COIL2 and COIL2→COIL1. Accordingly, when $\boldsymbol{V}_t$ is randomly initialized, the average accuracy of IGTL is much better than GTL, while the standard deviation of IGTL is much smaller than GTL. Specifically, IGTL obtains at least 16% improvements over GTL. The major reason is that GTL only achieves the approximate convergence, which restricts its performance. By contrast, IGTL can achieve strict convergence and obtain significant improvements over GTL. Consistently, for the case of initializing $\boldsymbol{V}_t$ with logistic regression (LR), the average accuracy of IGTL is superior to that of GTL, where the LR classifier is trained on the labeled source data to initialize $\boldsymbol{V}_t$ following [15].

Tables 4 and 5 show the classification results on Office+Caltech. In the tables, IGTL outperforms GTL in all six cross-domain datasets. Moreover, from the value of the standard deviation, the standard deviation of IGTL is smaller than that of GTL. These results demonstrate that IGTL can achieve more stable and better performance than GTL, which implies the effectiveness of IGTL.

5 Conclusion

In this paper, we focus on an optimization problem of GTL, which obtains promising results in transfer learning but can only achieve approximate convergence. Based on the analysis of the convergence conditions, we propose a new update rule and develop an algorithm called IGTL with strict convergence. The comparison experiments on the synthetic dataset and two real-world datasets confirm the validity of our IGTL algorithm, which can obtain better classification performance than the compared algorithms. In addition, GTL and IGTL use a fixed graph in the objective function, and the recent work [26] introduced the dynamic graph method. Learning a dynamic graph has the potential to improve performance. However, it will bring extra constraints to the objective function, affecting the subsequent optimization.

Acknowledgements This work was supported in part by Key-Area Research and Development Program of Guangdong Province (Grant No. 2019B010121001), National Natural Science Foundation of China (Grant Nos. 61722304, 61803096), and Guangdong Natural Science Foundation (Grant No. 2022A1515010688).

References

1 Pan S J, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng, 2010, 22: 1345–1359

2 Zhuang F Z, Qi Z Y, Duan K Y, et al. A comprehensive survey on transfer learning. Proc IEEE, 2021, 109: 43–76

3 Zhang Y, Gao X B, He L H, et al. Objective video quality assessment combining transfer learning with CNN. IEEE Trans Neural Netw Learn Syst, 2020, 31: 2716–2730

4 Li J, Qiu S, Shen Y Y, et al. Multisource transfer learning for cross-subject EEG emotion recognition. IEEE Trans Cybern, 2020, 50: 3281–3293

5 Wang E K, Liu X, Chen C M, et al. Voice-transfer attacking on industrial voice control systems in 5G-aided IIoT domain. IEEE Trans Ind Inf, 2021, 17: 7085–7092

6 Yan X H, Zhu J H, Kuang M C, et al. Missile aerodynamic design using reinforcement learning and transfer learning. Sci China Inf Sci, 2018, 61: 119204

7 Tao J W, Chung F L, Wang S T. A kernel learning framework for domain adaptation learning. Sci China Inf Sci, 2012, 55: 1983–2007

8 Long M S, Cao Y, Cao Z J, et al. Transferable representation learning with deep adaptation networks. IEEE Trans Pattern Anal Mach Intell, 2019, 41: 3071–3085

9 Wu Q Y, Wu H R, Zhou X M, et al. Online transfer learning with multiple homogeneous or heterogeneous sources. IEEE Trans Knowl Data Eng, 2017, 29: 1494–1507

10 Li J J, Lu K, Huang Z, et al. Heterogeneous domain adaptation through progressive alignment. IEEE Trans Neural Netw Learn Syst, 2019, 30: 1381–1391

11 Chen S T, Han L, Liu X L, et al. Subspace distribution adaptation frameworks for domain adaptation. IEEE Trans Neural Netw Learn Syst, 2020, 31: 5204–5218

12 Long M S, Wang J M, Ding G G, et al. Dual transfer learning. In: Proceedings of SIAM International Conference on Data Mining, Anaheim, 2012. 540–551

13 Wang D Q, Lu C W, Wu J J, et al. Softly associative transfer learning for cross-domain classification. IEEE Trans Cybern, 2020, 50: 4709–4721

14 Zhuang F Z, Luo P, Du C Y, et al. Triplex transfer learning: exploiting both shared and distinct concepts for text classification. IEEE Trans Cybern, 2014, 44: 1191–1203

15 Long M S, Wang J M, Ding G G, et al. Transfer learning with graph co-regularization. IEEE Trans Knowl Data Eng, 2014, 26: 1805–1818

16 Yang S M, Yi Z, Ye M, et al. Convergence analysis of graph regularized non-negative matrix factorization. IEEE Trans Knowl Data Eng, 2014, 26: 2151–2165

17 Li L X, Wu L, Zhang H S, et al. A fast algorithm for nonnegative matrix factorization and its convergence. IEEE Trans Neural Netw Learn Syst, 2014, 25: 1855–1863

18 Lu S T, Hong M Y, Wang Z D. A nonconvex splitting method for symmetric nonnegative matrix factorization: convergence analysis and optimality. IEEE Trans Signal Process, 2017, 65: 3120–3135

19 Zhao R B, Tan V Y F. A unified convergence analysis of the multiplicative update algorithm for regularized nonnegative matrix factorization. IEEE Trans Signal Process, 2018, 66: 129–138

20 Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization. Nature, 1999, 401: 788–791

21 Kong W W, Lei Y J, Lei Y, et al. Technique for image fusion based on non-subsampled contourlet transform domain improved NMF. Sci China Inf Sci, 2010, 53: 2429–2440

22 Yang Z Y, Zhang Y, Xiang Y, et al. Non-negative matrix factorization with dual constraints for image clustering. IEEE Trans Syst Man Cybern Syst, 2020, 50: 2524–2533

23 Lee D D, Seung H S. Algorithms for non-negative matrix factorization. In: Proceedings of Neural Information Processing Systems, 2001. 556–562

24 Long M S, Wang J M, Ding G G, et al. Transfer feature learning with joint distribution adaptation. In: Proceedings of IEEE International Conference on Computer Vision, Sydney, 2013. 2200–2207

25 Pan S J, Tsang I W, Kwok J T, et al. Domain adaptation via transfer component analysis. IEEE Trans Neural Netw, 2010, 22: 199–210

26 Li Z H, Nie F P, Chang X J, et al. Dynamic affinity graph construction for spectral clustering using multiple features. IEEE Trans Neural Netw Learn Syst, 2018, 29: 6323–6332