



# Multi-source Transfer Learning Based on the Power Set Framework

Bingbing Song<sup>1</sup> · Jianhan Pan<sup>1</sup> · Qiaoli Qu<sup>1</sup> · Zexin Li<sup>1</sup>

Received: 6 August 2022 / Accepted: 27 May 2023  
© The Author(s) 2023

## Abstract

Transfer learning is a great technology that can leverage knowledge from label-rich domains to address problems in similar domains that lack labeled data. Most previous works focus on single-source transfer, assuming the source domain contains sufficient labeled data and is close to the target domain. However, in practical applications, this assumption is hardly met, and labeled data exist in different domains. To improve the adaptability of transfer learning models for multi-source scenarios, many existing methods utilize the commonality and specificity across source domains. They either map all source domains with the target domain into a common feature space for knowledge transfer or combine multiple classifiers trained on pairs of each source and target to form a target classifier. However, the correlations across multiple source domains that can bring significant impacts on learning performance are ignored. In light of this, we propose a novel multi-source transfer learning method based on the power set framework (PSF-MSTL). First, PSF-MSTL constructs a power set framework that enables different source domains to be interrelated. Second, PSF-MSTL makes the source-domain framework integral and able to provide complementary knowledge using a dual-promotion strategy. Additionally, PSF-MSTL is formulated as an optimization problem, and an iterative algorithm is presented to address it. Finally, we conduct extensive experiments to show that PSF-MSTL can outperform many advanced multi-source transfer learning methods.

**Keywords** Multi-source transfer learning · Power set framework · Non-negative matrix tri-factorization · Knowledge fusion

## Abbreviations

PSF-MSTL	Multi-source transfer learning based on the power set framework
NMTF	Non-negative matrix tri-factorization
CE	Concept extension
CI	Concept intension
PLSA	Probabilistic latent semantic analysis
LR	Logistic regression
RCD-PLSA	Refined collaborative dual-PLSA
MST3L	Multi-source tri-training transfer learning
MCPC	Multi-classifier parameter combination
SDA	Selective domain adaptation

## 1 Introduction

Most traditional classification methods are based on an ideal assumption that the labeled training data are ample and subject to the distribution same as the unlabeled test data for achieving satisfactory performance. Since collecting labeled data is expensive and time consuming, these methods are difficult to quickly establish reliable classification models when the labeled data are scarce or the distribution of existing data changes. Transfer learning [1, 2], which leverages knowledge from label-rich source domains to train better target learners, can help alleviate these problems. Many transfer learning studies [3–19] focus on single-source transfer learning, and they assume that all the labeled data come from a single-source domain close to the target domain. However, in practical applications, the labeled data contained in one source domain are often inadequate, and available labeled data exist in other different domains. Thus, single-source transfer learning models are hardly applied in realistic multi-source scenarios.

To develop the adaptability of transfer learning models, many multi-source transfer learning methods are proposed [20–40]. A direct strategy adopted in some methods [20, 21, 38] is to mine the commonality of all sources, such as

✉ Jianhan Pan  
peter.jhpan@gmail.com  
Bingbing Song  
bbsong126@foxmail.com  
Qiaoli Qu  
1484738355@qq.com  
Zexin Li  
2963235203@qq.com

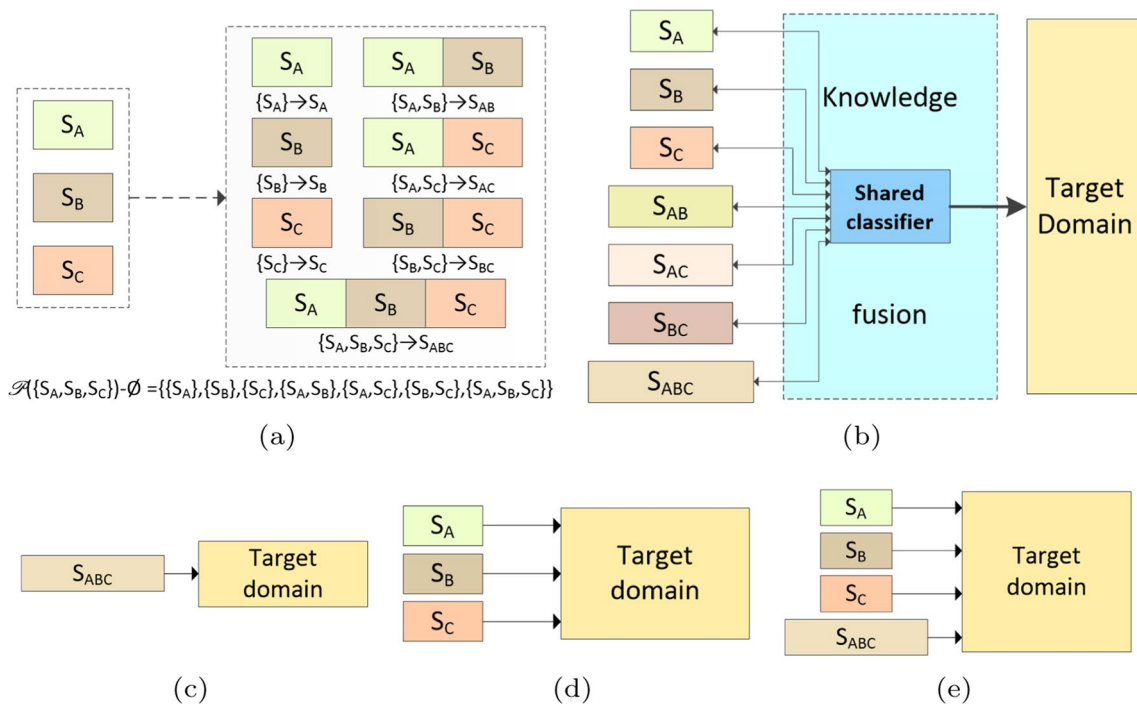
<sup>1</sup> Jiangsu Normal University, Xuzhou 221116, China

combining multiple source domains into a single-source domain by aligning all the raw features or mapping all source domains with the target domain into a low-dimensional space for knowledge transfer. With the help of the data expansion, they can improve the learning performance. However, the mutual interferences across source domains are ignored: one is that the common features that can build shared structures will reduce as source domains increase, and the other is that mining commonalities cannot help derive domain-specific features that have some discriminative power for training classifiers. For instance, if the recommender system of Tiktok is trained on the viewing habits of all customers, although it may benefit from the huge amount of data and deliver funny videos winning most users' favor, it will not ensure that people with different features can be accurately fed the videos more appealing to them. To capture the specificity of each source domain, some methods [22, 23] construct the target classifier by combining multiple classifiers trained on pairs of each source domain and target domain. These methods mitigate the negative effects of aligning or mapping strategies and extract domain-specific information effectively. However, the data distribution discrepancy among multiple sources makes it difficult to fuse all the information. For example, Tiktok can only recommend major-related videos to students relying on this strategy, and it hardly develops their versatility and inspires their potential talents. Other methods [24–26] incorporate the above two strategies, and thus they can utilize both commonality and specificity

across domains. However, they still fail to actually correlate multiple sources with each other and transfer knowledge from an integral source-domain framework. For example, using this strategy, Tiktok can only attract users by delivering funny and major-related videos, but not any other types of videos.

In this paper, we propose a novel multi-source transfer learning method based on the power set framework (PSF-MSTL), which can multiply combine each source domain with other source domains and fuse the knowledge learned from different source combinations to train a shared classifier. First, we introduce the power set concept to construct a source-domain framework that enables different source domains to be interrelated. The power set in mathematics is defined as a set composed of all subsets of the original set. For instance, the power set of  $\{A, B, C\}$  can be denoted as  $\mathcal{P}(\{A, B, C\}) = \{\emptyset, \{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}\}$ , where  $\emptyset$  represents an empty set. With this concept, we can view the collection of source domains as an original set and obtain its power set to identify the correlations among multiple source domains. Then different types of training datasets are generated according to the power set, structuring a power set framework that allows multiple source domains to be associated with each other. Figure 1 presents the three-source case, and Fig. 1a shows the source collection can be extended to a power set framework.

Second, to make the power set framework integral and able to provide comprehensive knowledge, we utilize a



**Fig. 1** In the three-source case, **a** shows the extension of the three-source collection, **b** presents the framework of our method, **c, d,** and **e** briefly visualizes three existing multi-source transfer learning strategies

dual-promotion strategy to integrate all the training datasets. Specifically, PSF-MSTL mines the latent information from corresponding latent feature spaces for pairs of each training dataset and target domain, and fuses all the information to bring the knowledge in the framework to the state of mutual complementarity. Then our method uses the fused knowledge and label information to train a shared classifier, which is exploited to further extract knowledge from every training dataset. In other words, refining the knowledge from the framework and training the shared classifier can reinforce each other directly. Thus, learning from different training datasets can promote each other indirectly by the shared classifier drawn on the complementary knowledge. Figure 1b shows that PSF-MSTL extracts and fuses latent information from seven training datasets to obtain complementary knowledge, which is used to train a shared classifier to help further learn from each training dataset. Figure 1c–e visualize three existing multi-source transfer learning strategies, respectively. It indicates that PSF-MSTL can build sounder relationships of multiple source domains than other strategies.

The main contributions of our work are listed below:

- (1) We systematically analyze the correlations among multiple source domains and propose PSF-MSTL, which can construct a power set framework to enable different source domains to be interrelated.
- (2) To obtain complementary knowledge from an integral framework, we utilize a dual-promotion strategy to integrate all the training datasets. Besides, PSF-MSTL is formulated as an optimization problem, and an iterative algorithm is presented to address it.
- (3) Additionally, we conducted extensive experiments on 20Newsgroups and Sentiment to verify the superiority and effectiveness of PSF-MSTL.

We organize the rest of this paper as follows: Sect. 2 summarizes some related works. Section 3 gives the notations and preliminary knowledge. Section 4 presents our method PSF-MSTL. Section 5 shows the experiment results. At last, Sect. 6 is the conclusion of our work.

## 2 Related Work

In this section, we summarize several existing single-source and multi-source transfer learning approaches related to our research.

We refer to the authoritative review literature [1, 2] and introduce some single-source transfer learning methods as well as some theoretical studies related to domain adaptation. Dai et al. [3] extended co-clustering to obtain the common word clusters and utilized them to propagate in-domain knowledge to the out-of-domain. Chen et al. [9] presented topical correspondence learning where the common and

domain-specific features can jointly make up their corresponding topics, and the knowledge is transferred through the common features. Jiang et al. [4] proposed a two-step approach where the common features shared across domains are mined and assigned appropriate weights first, and then the features specific to the target are identified by semi-supervised learning. Uguroglu et al. [6] distinguished variant and invariant features to fit different data distributions and defined the distribution issue as a convex optimization problem. Fang et al. [17] proved a generalization error theory that the labeled data in source domain and unlabeled data in target domain can reduce the risk of target in heterogeneous scenarios, and proposed two novel algorithms based on the theory. Blitzer et al. [7] adopted structural correspondence learning and exploited pivot features to model the correlations of source and target. Zhuang et al. [8] argued that feature clusters have more stable associations with document classes, and they presented a novel classification method to cluster raw features into concepts. Pan et al. [10] analyzed the word semantics in different scenarios, and proposed quadruple transfer learning to reduce marginal and conditional distribution discrepancies between domains. Fang et al. [18, 19] pioneered the learning bound for open set domain adaptation and proposed relevant algorithms to regularize the open set difference bound.

However, in real-world applications, labeled data exist in multiple source domains, which have different data distributions from each other. To adapt to more difficult scenarios, three common strategies are adopted in many multi-source classification methods. A straightforward strategy is to find the common information of all domains for knowledge transfer. Xu et al. [20] developed a novel domain adaptation method, which maps all source and target domains into a common feature space to reduce data distribution discrepancies, and then generates prediction results under the distribution weighted combining rule. Zhao et al. [21] presented an optimization model and developed it by smoothed approximation to obtain the generalization bound of all source domains. Another strategy is to capture the specificities of multiple sources by transferring knowledge from each source domain. Zhu et al. [22] developed a two-alignment-step framework where the first step is to align the distributions of each source domain with the target domain, and the second step is to align the classification results learned from each pair. Cheng et al. [23] trained multiple weak classifiers on pairs of each source and target, and constructed the target classifier by co-training the results produced by each classifier. Other methods incorporate the first two strategies to refine both common and specific knowledge from multiple source domains. Zhang et al. [25] presented a multi-source knowledge transfer method which can learn different feature extraction networks to align two-stage features and fuse information from two parts. Xu et al. [27] combined joint and separate alignment strategies and make two branches learn from each other through their complementarity. In addition,

we introduce some innovative multi-source learning strategies. Dai et al. [24] presented a selective domain adaptation method, which can extract the private features from the source domain nearest to the target domain, and then merge these features with the common features to train a target classifier. Li et al. [39] exploited an automatic sampling strategy to align classifiers and developed the cross-domain classification ability with the help of pseudo target labels. Wu et al. [40] proposed a novel method, which uses multiple graphs to model partial information and learn low-rank embedding through domain discrepancy and relevance. Kim et al. [41] mined the shared semantic space in the source and target domains and proposed a new concept-driven text classification algorithm based on deep neural networks. However, these methods ignore that the intrinsic relationships among multiple source domains may bring a large impact on the classification result. To address these problems, we propose the multi-source transfer learning method based on the power set framework(PSF-MSTL).

### 3 Preliminary Knowledge

In this section, we provide some mathematical notations with their denotations and give brief introductions of the high-level concepts and non-negative matrix tri-factorization(NMTF) technique that both support our method.

#### 3.1 Notations

We use the calligraphic letter  $\mathcal{D}$  to denote domains and the script letter  $\mathcal{P}$  to abbreviate the power set framework. The data matrices are expressed by the uppercase(such as  $X$ ), and  $X_{[i,j]}$  represents the  $i$ -th row and  $j$ -th element of matrix  $X$ . In addition, the set of non-negative real numbers and real numbers are denoted as  $R_+$  and  $R$ , respectively. Table 1 lists some notations with their denotations frequently used in this paper.

#### 3.2 High-Level Concepts

Due to the data distribution discrepancies among different domains in multi-source transfer learning problems, it is hard to obtain satisfactory performance by learning from the raw features directly. Thus, to build a more stable transfer learning bridge, we incorporate the high-level concepts to transform raw features into concepts. To be specific, the high-level concepts include two fundamental definitions: the concept extension (CE) and the concept intension (CI) [12, 13]. CE refers to the distribution of the raw word features that imply the same concept. CI refers to the associations between the document classes and the concepts. In this paper, we exploit identical concepts and alike concepts to reduce the data distribution discrepancies and train a shared classifier. Both of them are summarized in Table 2.

**Table 1** Notations and denotations

Notations	Denotations
$\mathcal{D}$	The data domain
$\mathcal{P}$	The power set framework
$X$	The feature-document co-occurrence matrix of $\mathcal{D}$
$i_s$	The index of source domain
$s$	The number of source domains
$t$	The index of the target domain
$m$	The number of raw features
$n$	The number of documents
$k$	The number of all high-level concepts
$c$	The number of document classes
$k_1$	The number of identical concepts
$k_2$	The number of alike concepts
$F$	The matrix of concept extension
$H$	The matrix of concept intension
$T$	Transposition of matrix

**Table 2** Identical concepts and alike concepts

High-level concepts	Extension	Intension
Identical concepts	Same	Same
Alike concepts	Different	Same

#### 3.3 Non-negative Matrix Tri-factorization

To implement the high-level concepts mentioned above, we introduce the non-negative matrix tri-factorization(NMTF) technique, which has been widely used in text and image classification researches [10–13, 16, 42]. NMTF can decompose the data matrix into the multiplication of three non-negative factor matrices, and the basic formula of NMTF is:

$$X_{m \times n} = F_{m \times k} H_{k \times c} G_{n \times c}^T \quad (1)$$

where  $X \in R^{m \times n}$ ,  $F \in R^{m \times k}$ ,  $H \in R^{k \times c}$ ,  $G \in R^{n \times c}$ . Here,  $m$ ,  $n$ ,  $k$ , and  $c$  are the numbers of features, documents, high-level concepts, and document classes, respectively. Besides,  $X_{m \times n}$  is the feature-document matrix which has  $m$  rows and  $n$  columns,  $F_{m \times k}$  is the feature-concept matrix where each column indicates the distribution of raw features which belong to one concept,  $H_{k \times c}$  is the factor matrix to associate concepts with document classes, and  $G_{n \times c}^T$  is the transposition of  $G_{n \times c}$  that can be considered as a target classifier. In addition,  $F$  and  $H$  represent CE and CI, respectively.

Furthermore, NMTF is an optimization problem as follows:

$$\min_{F, H, G \geq 0} \|X - FHG^T\|^2 \text{ s.t. } \sum_{i=1}^m F_{[i,j]} = 1, \sum_{j=1}^c G_{[i,j]} = 1 \quad (2)$$

Since there are multiple source domains and one target domain in multi-source transfer learning tasks, NMTF is developed to obtain better adaptability. The optimization problem can be developed as follows:

$$\begin{aligned}
 & \min_{F_{i_s}, F_t, H, G_{i_s}, G_t \geq 0} \sum_{i_s=1}^s \|X_{i_s} - F_{i_s} H G_{i_s}^T\|^2 + \|X_t - F_t H G_t^T\|^2 \\
 & s.t. \sum_{i=1}^m F_{i_s[i,j]} = 1, \sum_{i=1}^m F_{t[i,j]} = 1, \sum_{j=1}^c G_{i_s[i,j]} = 1, \sum_{j=1}^c G_{t[i,j]} = 1
 \end{aligned} \tag{3}$$

### 4 Multi-source Transfer Learning Based on the Power Set Framework

In this section, PSF-MSTL is proposed to address multi-source transfer learning problems. We first discuss how many training datasets are generated in a classification task. Then PSF-MSTL is defined as an optimization problem, and an iterative algorithm is presented to solve it.

#### 4.1 The Number of Training Datasets

Since PSF-MSTL multiply combines each source domain, it is necessary to analyze all possible combinations of source domains to ascertain how many training datasets are generated. According to the theory of the power set in mathematics, the number of the subsets in  $A$ 's power set is  $2^{|A|}$ , where  $|A|$  is the cardinality of set  $A$ , which means  $A$ 's power set contains  $2^{|A|} - 1$  subsets that are not empty. With this theory, PSF-MSTL produces  $2^s - 1$  training datasets in a classification task, where  $s$  represents the total number of source domains. For clarity, we use  $l$ -source domain to denote each training dataset, where  $l$  is the number of source domains contained in a training dataset. According to  $l$ , these training datasets can be divided into  $s$  types, denoted as  $l$ -source type. By analyzing the number of training datasets in each type, we can infer that  $l$ -source type contains  $C_s^l$   $l$ -source domains. For example, in the three-source task, a total of  $7(2^3 - 1)$  training datasets are generated, and they can be divided into three types: the one-source type contains  $3(C_3^1)$  one-source domains, the two-source type contains  $3(C_3^2)$  two-source domains, and the three-source type contains  $1(C_3^3)$  three-source domain.

#### 4.2 Problem Definition

Suppose, in a multi-source transfer learning problem, the data domain contains  $s$  source and one target domains, denoted as  $\mathcal{D} = (\mathcal{D}_1, \dots, \mathcal{D}_s, \mathcal{D}_t)$ . Since PSF-MSTL extends the source collection to the power set framework which has

$2^s - 1$  training datasets, we define a general formula for the framework, denoted as  $\mathcal{P} = (\mathcal{D}'_1, \mathcal{D}'_2, \dots, \mathcal{D}'_{2^s-1})$ , where  $\mathcal{D}'$  represents the training dataset. As a result, the data domain can be developed to  $\mathcal{D} = (\mathcal{D}'_1, \dots, \mathcal{D}'_{2^s-1}, \mathcal{D}_t)$ , where the first  $2^s - 1$  domains are labeled training datasets, i.e.,  $\mathcal{D}'_{i_s} = \{X_{i_s}^i, y_{i_s}^i\}_{i=1}^{n_{i_s}}$  ( $1 \leq i_s \leq 2^s - 1$ ), and the left one is the unlabeled target domain, i.e.,  $\mathcal{D}_t = \{x_t^i\}_{i=1}^{n_t}$ .  $n_{i_s}$  is the number of documents in the  $i_s$ -th source domain, and  $n_t$  is the number of documents in the target domain. In addition, the feature-document matrices of the data domain can be given as  $X = \{X_1, \dots, X_{2^s-1}, X_t\}$ . Then the objective function is formulated as:

$$\mathcal{L} = \sum_{i_s=1}^{2^s-1} \left( \|X_{i_s} - F_{i_s} H_{i_s} G_{i_s}^T\|^2 + \|X_t - F_{t|i_s} H_{t|i_s} G_t^T\|^2 \right) \tag{4}$$

where  $X_{i_s} \in R_+^{m \times n_{i_s}}$ ,  $X_t \in R_+^{m \times n_t}$ ,  $F_{i_s} \in R_+^{m \times k}$ ,  $F_{t|i_s} \in R_+^{m \times k}$ ,  $H_{i_s} \in R_+^{k \times c}$ ,  $H_{t|i_s} \in R_+^{k \times c}$ ,  $G_{i_s} \in R_+^{n_{i_s} \times c}$ ,  $G_t \in R_+^{n_t \times c}$ .

Since the identical concepts and the alike concepts are utilized in this paper,  $F_{i_s}$ ,  $F_{t|i_s}$ ,  $H_{i_s}$  and  $H_{t|i_s}$  are separately divided into two parts. Specifically, in a pair of training dataset and target domain, the identical concept shares the same CE across domains while the alike concept has different CE in different domains. Thus, in the  $i_s$ -th pair, the feature-concept matrix of identical concept is denoted as  $F_{i_s}^1$  for both the training dataset and target domain, and the feature-concept matrix of alike concept is denoted as  $F_{i_s}^2$  and  $F_{t|i_s}^2$  for the training dataset and target domain, respectively. As a result,  $F_{i_s} = [F_{i_s}^1, F_{i_s}^2]$  and  $F_{t|i_s} = [F_{i_s}^1, F_{t|i_s}^2]$ , where  $F_{i_s}^1 \in R_+^{m \times k_1}$ ,  $F_{i_s}^2 \in R_+^{m \times k_2}$ ,  $F_{t|i_s}^2 \in R_+^{m \times k_2}$ , and  $k_1 + k_2 = k$ . Similarly, both the identical and alike concepts keep the same CI in the  $i_s$ -th pair, and we use  $H_{i_s}^1$  and  $H_{i_s}^2$  to express these two types of associations between concepts and document classes, i.e.,  $H_{i_s} = H_{t|i_s} = \begin{bmatrix} H_{i_s}^1 \\ H_{i_s}^2 \end{bmatrix}$ , where  $H_{i_s}^1 \in R_+^{k_1 \times c}$ ,  $H_{i_s}^2 \in R_+^{k_2 \times c}$ . Thus, we can rewrite the objective function in (4):

$$\begin{aligned}
 \mathcal{L} &= \sum_{i_s=1}^{2^s-1} \left( \|X_{i_s} - F_{i_s} H_{i_s} G_{i_s}^T\|^2 + \|X_t - F_{t|i_s} H_{t|i_s} G_t^T\|^2 \right) \\
 &= \sum_{i_s=1}^{2^s-1} \left( \|X_{i_s} - [F_{i_s}^1, F_{i_s}^2] \begin{bmatrix} H_{i_s}^1 \\ H_{i_s}^2 \end{bmatrix} G_{i_s}^T\|^2 + \|X_t - [F_{i_s}^1, F_{t|i_s}^2] \begin{bmatrix} H_{i_s}^1 \\ H_{i_s}^2 \end{bmatrix} G_t^T\|^2 \right) \tag{5}
 \end{aligned}$$

There are five constraints to  $F_{i_s}^1$ ,  $F_{i_s}^2$ ,  $F_{t|i_s}^2$ ,  $G_{i_s}$ , and  $G_t$ . Therefore, the optimization problem is as follows:

$$\begin{aligned}
 & F_{i_s}, F_{t|i_s}, H_{i_s}, G_{i_s}, G_t \geq 0 \\
 & \min \mathcal{L} \\
 & s.t. \sum_{i=1}^m F_{i_s[i,j]}^1 = 1, \\
 & \sum_{i=1}^m F_{i_s[i,j]}^2 = 1, \sum_{i=1}^m F_{t|i_s[i,j]}^2 = 1, \\
 & \sum_{j=1}^c G_{i_s[i,j]} = 1, \sum_{j=1}^c G_{t[i,j]} = 1
 \end{aligned} \tag{6}$$

### 4.3 Solution to MSTL

Observing that  $\mathcal{L}$  is not concave, we further develop the objective function and propose an iterative algorithm instead of the latest nonlinear optimization techniques. According to the properties of the trace and Frobenius norm, we can expand the objective function as follows:

$$\begin{aligned}
 \mathcal{L} &= \sum_{i_s=1}^{2^s-1} (\|X_{i_s} - [F_{i_s}^1, F_{i_s}^2] \begin{bmatrix} H_{i_s}^1 \\ H_{i_s}^2 \end{bmatrix} G_{i_s}^\top\|^2 \\
 & \quad + \|X_t - [F_{i_s}^1, F_{i_s}^2] \begin{bmatrix} H_{i_s}^1 \\ H_{i_s}^2 \end{bmatrix} G_t^\top\|^2) \\
 &= \sum_{i_s=1}^{2^s-1} (tr(X_{i_s}^\top X_{i_s} - 2 \cdot X_{i_s}^\top [F_{i_s}^1, F_{i_s}^2] \begin{bmatrix} H_{i_s}^1 \\ H_{i_s}^2 \end{bmatrix} G_{i_s}^\top \\
 & \quad + G_{i_s} \begin{bmatrix} H_{i_s}^1 \\ H_{i_s}^2 \end{bmatrix}^\top [F_{i_s}^1, F_{i_s}^2]^\top \\
 & \quad [F_{i_s}^1, F_{i_s}^2] \begin{bmatrix} H_{i_s}^1 \\ H_{i_s}^2 \end{bmatrix} G_{i_s}^\top) + tr(X_t^\top X_t \\
 & \quad - 2 \cdot X_t^\top [F_{i_s}^1, F_{i_s}^2] \begin{bmatrix} H_{i_s}^1 \\ H_{i_s}^2 \end{bmatrix} G_t^\top \\
 & \quad + G_t \begin{bmatrix} H_{i_s}^1 \\ H_{i_s}^2 \end{bmatrix}^\top [F_{i_s}^1, F_{i_s}^2]^\top [F_{i_s}^1, F_{i_s}^2] \begin{bmatrix} H_{i_s}^1 \\ H_{i_s}^2 \end{bmatrix} G_t^\top) \\
 &= \sum_{i_s=1}^{2^s-1} (tr(X_{i_s}^\top X_{i_s} - 2 \cdot X_{i_s}^\top A_{i_s} - 2 \cdot X_{i_s}^\top B_{i_s} + G_{i_s} H_{i_s}^\top F_{i_s}^\top A_{i_s} \\
 & \quad + G_{i_s} H_{i_s}^\top F_{i_s}^\top B_{i_s} + 2 \cdot G_{i_s} H_{i_s}^\top F_{i_s}^\top B_{i_s}) \\
 & \quad + tr(X_t^\top X_t - 2 \cdot X_t^\top A_t \\
 & \quad - 2 \cdot X_t^\top B_t + G_t H_t^\top F_t^\top A_t \\
 & \quad + G_t H_t^\top F_t^\top B_t + 2 \cdot G_t H_t^\top F_t^\top B_t)) \\
 & s.t. \sum_{i=1}^m F_{i_s[i,j]}^1 = 1, \sum_{i=1}^m F_{i_s[i,j]}^2 = 1, \sum_{i=1}^m F_{t|i_s[i,j]}^2 = 1, \sum_{j=1}^c G_{i_s[i,j]} = 1, \\
 & \sum_{j=1}^c G_{t[i,j]} = 1
 \end{aligned} \tag{7}$$

where  $A_{i_s} = F_{i_s}^1 H_{i_s}^1 G_{i_s}^\top$ ,  $B_{i_s} = F_{i_s}^2 H_{i_s}^2 G_{i_s}^\top$ ,  $A_t = F_{i_s}^1 H_{i_s}^1 G_t^\top$ ,  $B_t = F_{i_s}^2 H_{i_s}^2 G_t^\top$ . Since the real label information of source domains already exists, we only need to solve  $G_t$ . Then we come to the partial differentials of  $\mathcal{L}$  as follows:

$$\begin{aligned}
 \frac{\partial \mathcal{L}}{\partial F_{i_s}^1} &= -2 \cdot X_{i_s} G_{i_s} H_{i_s}^{1\top} + 2 \cdot A_{i_s} G_{i_s} H_{i_s}^{1\top} \\
 & \quad + 2 \cdot B_{i_s} G_{i_s} H_{i_s}^{1\top} \\
 & \quad - 2 \cdot X_t G_t H_{i_s}^{1\top} + 2 \cdot A_t G_t H_{i_s}^{1\top} \\
 & \quad + 2 \cdot B_t G_t H_{i_s}^{1\top}
 \end{aligned} \tag{8}$$

$$\begin{aligned}
 \frac{\partial \mathcal{L}}{\partial F_{i_s}^2} &= -2 \cdot X_{i_s} G_{i_s} H_{i_s}^{2\top} \\
 & \quad + 2 \cdot B_{i_s} G_{i_s} H_{i_s}^{2\top} + 2 \cdot A_{i_s} G_{i_s} H_{i_s}^{2\top}
 \end{aligned} \tag{9}$$

$$\begin{aligned}
 \frac{\partial \mathcal{L}}{\partial F_{t|i_s}^2} &= -2 \cdot X_t G_t H_{i_s}^{2\top} \\
 & \quad + 2 \cdot B_t G_t H_{i_s}^{2\top} + 2 \cdot A_t G_t H_{i_s}^{2\top}
 \end{aligned} \tag{10}$$

$$\begin{aligned}
 \frac{\partial \mathcal{L}}{\partial H_{i_s}^1} &= -2 \cdot F_{i_s}^{1\top} X_{i_s} G_{i_s} \\
 & \quad + 2 \cdot F_{i_s}^{1\top} A_{i_s} G_{i_s} + 2 \cdot F_{i_s}^{1\top} B_{i_s} G_{i_s} \\
 & \quad - 2 \cdot F_{i_s}^{1\top} X_t G_t + 2 \cdot F_{i_s}^{1\top} A_t G_t \\
 & \quad + 2 \cdot F_{i_s}^{1\top} B_t G_t
 \end{aligned} \tag{11}$$

$$\begin{aligned}
 \frac{\partial \mathcal{L}}{\partial H_{i_s}^2} &= -2 \cdot F_{i_s}^{2\top} X_{i_s} G_{i_s} \\
 & \quad + 2 \cdot F_{i_s}^{2\top} B_{i_s} G_{i_s} + 2 \cdot F_{i_s}^{2\top} A_{i_s} G_{i_s} \\
 & \quad - 2 \cdot F_{i_s}^{2\top} X_t G_t + 2 \cdot F_{i_s}^{2\top} B_t G_t \\
 & \quad + 2 \cdot F_{i_s}^{2\top} A_t G_t
 \end{aligned} \tag{12}$$

$$\frac{\partial \mathcal{L}}{\partial G_t} = -2 \cdot X_t^\top \sum_{i_s=1}^{2^s-1} F_{t|i_s} H_{i_s} + 2 \cdot G_t \sum_{i_s=1}^{2^s-1} H_{i_s}^\top F_{t|i_s}^\top F_{t|i_s} H_{i_s} \tag{13}$$

The iterative algorithm updates these factor matrices as follows:



$$F^1_{i_s[ij]} \leftarrow F^1_{i_s[ij]} \cdot \sqrt{\frac{[X_{i_s} G_{i_s} H^1_{i_s}{}^\top + X_t G_t H^1_{i_s}{}^\top]_{[ij]}}{[A_{i_s} G_{i_s} H^1_{i_s}{}^\top + B_{i_s} G_{i_s} H^1_{i_s}{}^\top + A_t G_t H^1_{i_s}{}^\top + B_t G_t H^1_{i_s}{}^\top]_{[ij]}}} \tag{14}$$

$$F^2_{i_s[ij]} \leftarrow F^2_{i_s[ij]} \cdot \sqrt{\frac{[X_{i_s} G_{i_s} H^2_{i_s}{}^\top]_{[ij]}}{[B_{i_s} G_{i_s} H^2_{i_s}{}^\top + A_{i_s} G_{i_s} H^2_{i_s}{}^\top]_{[ij]}}} \tag{15}$$

$$F^2_{t|i_s[ij]} \leftarrow F^2_{t|i_s[ij]} \cdot \sqrt{\frac{[X_t G_t H^2_{i_s}{}^\top]_{[ij]}}{[B_t G_t H^2_{i_s}{}^\top + A_t G_t H^2_{i_s}{}^\top]_{[ij]}}} \tag{16}$$

$$H^1_{i_s[ij]} \leftarrow H^1_{i_s[ij]} \cdot \sqrt{\frac{[F^1_{i_s}{}^\top X_{i_s} G_{i_s} + F^1_{i_s}{}^\top X_t G_t]_{[ij]}}{[F^1_{i_s}{}^\top A_{i_s} G_{i_s} + F^1_{i_s}{}^\top B_{i_s} G_{i_s} + F^1_{i_s}{}^\top A_t G_t + F^1_{i_s}{}^\top B_t G_t]_{[ij]}}} \tag{17}$$

$$H^2_{i_s[ij]} \leftarrow H^2_{i_s[ij]} \cdot \sqrt{\frac{[F^2_{i_s}{}^\top X_{i_s} G_{i_s} + F^2_{t|i_s}{}^\top X_t G_t]_{[ij]}}{[F^2_{i_s}{}^\top B_{i_s} G_{i_s} + F^2_{i_s}{}^\top A_{i_s} G_{i_s} + F^2_{t|i_s}{}^\top B_t G_t + F^2_{t|i_s}{}^\top A_t G_t]_{[ij]}}} \tag{18}$$

$$G_{t|i_j} \leftarrow G_{t|i_j} \cdot \sqrt{\frac{[X_t{}^\top \sum_{i_s=1}^{2^s-1} F_{t|i_s} H_{i_s}]_{[ij]}}{[G_t \sum_{i_s=1}^{2^s-1} H_{i_s}{}^\top F_{t|i_s}{}^\top F_{t|i_s} H_{i_s}]_{[ij]}}} \tag{19}$$

We calculate all the variable matrices in each round of iteration and utilize Eq.(20) to normalize  $F^1_{i_s}$ ,  $F^2_{i_s}$ ,  $F^2_{t|i_s}$ ,  $H^1_{i_s}$ ,  $H^2_{i_s}$ , and  $G_t$  as follows:

$$F^1_{i_s[ij]} \leftarrow \frac{F^1_{i_s[ij]}}{\sum_{j=1}^{k_1} F^1_{i_s[ij]}}, F^2_{i_s[ij]} \leftarrow \frac{F^2_{i_s[ij]}}{\sum_{j=1}^{k_2} F^2_{i_s[ij]}}, \tag{20}$$

$$F^2_{t|i_s[ij]} \leftarrow \frac{F^2_{t|i_s[ij]}}{\sum_{j=1}^{k_2} F^2_{t|i_s[ij]}},$$

$$H^1_{i_s[ij]} \leftarrow \frac{H^1_{i_s[ij]}}{\sum_{j=1}^c H^1_{i_s[ij]}}, H^2_{i_s[ij]} \leftarrow \frac{H^2_{i_s[ij]}}{\sum_{j=1}^c H^2_{i_s[ij]}},$$

$$G_{t|i_j} \leftarrow \frac{G_{t|i_j}}{\sum_{j=1}^c G_{t|i_j}}$$

According to Eqs.(14–20), an iterative algorithm is proposed in Algorithm 1. The data matrices are normalized such that  $X^T_{i_s} 1_m = 1_n$ ,  $X^T_t 1_m = 1_n$ . To initialize CE in each pair of the target domain and training dataset, all the data in each pair are combined to implement PLSA [44]. For example, we set the number of concepts in the  $i_s$ -th pair as  $k_1 + k_2$ , and obtain feature-concept matrix  $W_{i_s}$ .  $W_{i_s}$  is divided into two parts  $W_{i_s} = [W^1_{i_s}, W^2_{i_s}]$ , where  $W^1_{i_s} \in R^{m \times k_1}$ ,  $W^2_{i_s} \in R^{m \times k_2}$ . As a result,  $F^1_{i_s}$  is initialized as  $W^1_{i_s}$  as well as  $F^2_{i_s}$  and  $F^2_{t|i_s}$  are initialized as  $W^2_{i_s}$ . In addition, we initialize the classifier of each pair by implementing logistic regression [43], denoted as  $G_{t|i_s}$ , and then integrate them by Eqs.(14–20) to get the initial  $G_t$ .  $G_{i_s}$  is assigned as the true label information of documents.

### 4.4 Computational Complexity of the Iterative Algorithm

We verify the efficiency of our method by analyzing the computational complexities of Eqs.(14–19) in each round of iteration. For instance, the computational complexity of Eq.(14) is  $O((2^s - 1)(5mnc + 2mkc + 6mk_1c + mk_1))$ , where  $n = n_s + n_t$ . In general,  $c < k_1 < k \ll n$ ,  $s \ll n < m$ , so the computational complexity can be written as  $O(mnc)$ . Similarly, the computational complexities of Eqs.(15), (16), (17), (18), and (19) are, respectively,  $O(mn_s c)$ ,  $O(mnc + mk_1 n)$ ,  $O(mnc + mk_2 n)$ , and  $O(mn_t k)$ . Thus, in each round of iteration, the maximal computational complexity is  $O(mnc + mnk)$ . As a result, the computational complexity of Algorithm 1 is  $O(maxIter \cdot (mnc + mnk))$ .

---

**Algorithm 1:** Multi-source Transfer Learning based on the Power Set Framework
 

---

**Input:**  $\{X_{i_s}\}_{i_s=1}^{2^s-1}$ ,  $X_t$ ,  $\{G_{i_s}\}_{i_s=1}^{2^s-1}$ , parameters  $k_1$ ,  $k_2$ , and the number of iterations  $maxIter$ .

**Output:**  $F_{i_s}^1$ ,  $F_{i_s}^2$ ,  $F_{t|i_s}^2$ ,  $H_{i_s}^1$ ,  $H_{i_s}^2$  ( $1 \leq i_s \leq 2^s - 1$ ), and  $G_t$ .

1 Normalize the data matrices by

$$X_{i_s[i,j]} \leftarrow X_{i_s[i,j]} / \sum_{i=1}^m X_{i_s[i,j]} (1 \leq i_s \leq 2^s - 1),$$

$$X_{t[i,j]} \leftarrow X_{t[i,j]} / \sum_{i=1}^m X_{t[i,j]};$$

2  $F_{i_s}^1$ ,  $F_{i_s}^2$ ,  $F_{t|i_s}^2$ ,  $G_t$  are initialized according to Section 4.3;  $H_{i_s}^1$ ,  $H_{i_s}^2$  are randomly assigned;

3 **for**  $k \leftarrow 1$  **to**  $maxIter$  **do**

4     **for**  $i_s \leftarrow 1$  **to**  $2^s - 1$  **do**

5         Update  $F_{i_s}^{1(k)}$ ,  $F_{i_s}^{2(k)}$ ,  $F_{t|i_s}^{2(k)}$ ,  $H_{i_s}^{1(k)}$ ,  $H_{i_s}^{2(k)}$  by Eq.(14), (15), (16), (17), (18) respectively;

6     **end**

7     Update  $G_t^{(k)}$  by Eq.(19);

8     Normalize these CE, CI and  $G_t^{(k)}$  by Eq.(20);

9 **end**

10 Return  $F_{i_s}^1$ ,  $F_{i_s}^2$ ,  $F_{t|i_s}^2$ ,  $H_{i_s}^1$ ,  $H_{i_s}^2$  ( $1 \leq i_s \leq 2^s - 1$ ), and  $G_t$ .

---

## 5 Experimental Evaluation

In this section, we compare our method with other advanced multi-source transfer learning methods on two benchmark datasets. Furthermore, we systematically verify the effectiveness of PSF-MSTL.

### 5.1 Data Preparation

20-Newsgroups<sup>1</sup> is one of the international standard datasets used in text classification research. It collects approximately 20000 news documents and divides them into 20 different newsgroups. Similar newsgroups are integrated into one topic, resulting in four topics. For instance, comp.graphics, comp.sys.mac.hardware, comp.sys.ibm.pc.hardware, comp.os.-ms-windows.misc are categorized into the topic comp. The topics and their corresponding newsgroups in 20news-groups are listed in Table 3. We design the three-source classification tasks as follows:

Since documents are classified into two classes in the binary classification tasks, we first choose rec as the positive class and sci as the negative class. Then we randomly select

one newsgroup from rec and sci, respectively, to construct the first source domain, and each newsgroup has 200 news documents. The other two source domains and the target domain are generated in a similar way. Thus,  $576(P_4^4 \times P_4^4)$  three-source text classification tasks are produced. Since there exist  $6(P_3^3)$  permutations of three sources, and different permutations can hardly affect the experimental results, the number of tasks can be reduced to  $96(P_4^4 \times P_4^4 \div P_3^3)$ . In summary, there are 96 three-source transfer learning tasks on 20NewsGroups.

Sentiment<sup>2</sup> is another dataset widely used in binary text classification. It is composed of four fields, i.e., books, electronics, kitchen, and DVD. We randomly choose them as three source domains and one target domain, and each domain has 200 positive and 200 negative documents. Thus, we produce  $4(P_4^3 \div P_3^3)$  three-source classification tasks on Sentiment.

Office-Home<sup>3</sup> has four major categories, i.e., Art, Clipart, Product, and Real World. Each major category contains the same subcategories. We randomly choose two subcategories as positive and negative classes. In our experiment, we choose Alarm\_Clock vs. Backpack. For three-source transfer learning tasks, since there are four major categories in

<sup>1</sup> <http://people.csail.mit.edu/jrennie/20NewsGroups/>.

<sup>2</sup> <http://www.cs.jhu.edu/mdredze/datasets/sentiment/>.

<sup>3</sup> <https://paperswithcode.com/dataset/office-home>.



**Table 3** Topics and newsgroups

Topics	Newsgroups
<i>Rec</i>	<i>Rec.</i> { <i>sport.hockey,sport.baseball,motorcycles,autos</i> }
<i>Sci</i>	<i>Sci.</i> { <i>space,med,electronics,crypt</i> }
<i>Talk</i>	<i>Talk.</i> { <i>religion.misc,politics.mideast,politics.guns,politics.misc</i> }
<i>Comp</i>	<i>Comp.</i> { <i>graphics,os.ms-windows.misc,sys.ibm.pc.hardware,sys.mac.hardware</i> }

Office-Home, we randomly choose three of them as source domains and choose the rest domain as the target domain. Thus, four classification tasks are constructed in a set of experiments. All images are first converted to grayscale images and compressed to 32\*32 pixels, followed by the normalization of grayscale values. Then the grayscale features of a picture are arranged into rows, forming a matrix with features as rows and samples as columns.

## 5.2 Experimental Setting

### 5.2.1 Algorithms in Comparison

- (1) Logistic regression(LR) [43]: LR is a traditional supervised classification method. We combine all source domains into one source domain to train a classifier and use the target domain to test.
- (2) Non-negative matrix tri-factorization(NMTF) [9]: We use NMTF that can train on the source and target domains simultaneously as the baseline method.
- (3) Multi-source text classification methods: RCD-PLSA [38], MST3L [23], and MCPC [26] use three strategies described in Sect. 1, respectively. In addition, PSF-MSTL is compared to the unsupervised text classification method SDA [24] and deep-neural-network-based method TSM-DNN [41].

### 5.2.2 Parameter Settings

Since it is exceedingly difficult to quantify the high-level concepts and find an optimal solution, the numbers of identical and alike concepts in each pair of the target domain and training dataset are empirically set as 20( $k_1 = k_2 = 20$ ), and the number of max iterations is set as 100( $maxIter = 100$ ). In addition, LR is conducted on *Matlab*.<sup>4</sup> NMTF is obtained from [9]. Their parameters are set as default ones. To be fair, RCD-PLSA, MST3L, MCPC, and SDA are incorporated NMTF technique, and they are trained with the parameters

same as PSF-MSTL. The initial learning rate of TSM-DNN is set as 0.01.

### 5.2.3 Evaluation Metrics

We use two general evaluation matrices and count the number of negative transfer to evaluate the experimental results.

(1) *Accuracy*

$$Accuracy = \frac{|\{d : d \in D \wedge f(d) = y(d)\}|}{n}$$
 where  $y(d)$  is the true label of document  $d$ ,  $f(d)$  is the predictive label and  $n$  is the number of documents.

(2) *F1 – measure*

$$F1 - measure = \frac{F_{1N} + F_{1P}}{2}$$
 where  $F_{1N}$  ( $F_1$  on negative predictions) =  $(2 \cdot P_N \cdot R_N) / (P_N + R_N)$ ,  $F_{1P}$  ( $F_1$  on positive predictions) =  $(2 \cdot P_P \cdot R_P) / (P_P + R_P)$ ,  $R_N$  (recall on negative predictions) =  $a / (a + b)$ ,  $P_N$  (precision on negative predictions) =  $a / (a + c)$ ,  $R_P$  (recall on positive predictions) =  $d / (d + c)$ ,  $P_P$  (precision on positive predictions) =  $d / (d + b)$ .

(3) *Count of negative transfer*

We count the occurrences of negative transfer to measure transfer learning methods from another perspective [10]. The number of negative transfers is denoted as *NumNT*.

## 5.3 Experimental Results

Here, PSF-MSTL is compared with LR, NMTF, RCD-PLSA, MST3L, MCPC, and SDA on 20Newsgroups and Sentiment datasets, respectively. The average performances of PSF-MSTL and other transfer learning methods are shown in Table 4.

(1) Comparison on 20Newsgroups

- (a) The experimental results on 20Newsgroups are shown in Table 4. We can observe that PSF-MSTL outperforms all the compared methods. There are two reasons why PSF-MSTL can achieve the best performance. First, it constructs a power set framework, which allows each single-source domain to join different training datasets, and thus multiple source domains correlate from various perspectives preliminarily. Second, PSF-MSTL integrates all the training datasets, enabling the target classifier to be trained on complementary knowledge from an integral framework. Consequently, PSF-MSTL not only mines the commonality and specificity across domains but also builds

<sup>4</sup> <http://www.kyb.tuebingen.mpg.de/bs/people/pgehler/code/index.html>.

**Table 4** Performances (%) on 20Newsgroups and Sentiment (10 repeated experiments)

	LR [43]	NMTF [9]	TSM-DNN [41]	RCD-PLSA [38]	MST3L [23]	MCPC [26]	SDA [24]	PSF-MSTL
Performance of total 96 three-source tasks on 20Newsgroups								
<i>Accuracy</i>	72.67	82.26	82.88	91.50	94.46	94.79	90.12	95.23
<i>F1 – Measure</i>	72.09	81.92	82.59	91.47	94.45	94.78	90.05	95.22
<i>NumNT</i>	–	–	–	0	0	0	1	0
Performance of total 4 three-source tasks on Sentiment								
<i>Accuracy</i>	74.69	69.36	70.11	75.38	72.65	75.00	76.38	78.3
<i>F1 – Measure</i>	74.59	68.70	69.78	75.09	72.44	74.86	76.29	78.28
<i>NumNT</i>	–	–	–	2	3	2	1	0

**Table 5** Performances (%) on Office-Home (10 repeated experiments)

	LR	NMTF	TSM-DNN	RCD-PLSA	MST3L	MCPC	SDA	PSF-MSTL
Mean accuracy	57.09	57.50	47.91	56.63	55.46	60.86	59.48	61.32

the intrinsic relevance of multiple source domains. Other methods hardly make full use of the latent information because they do not construct correlations across multiple source domains, which causes them to be inferior to PSF-MSTL on multi-source classification tasks.

- (b) All the multi-source transfer learning methods outperform LR, which means traditional machine learning methods hardly address multi-source transfer learning problems.
- (c) NMTF, RCD-PLSA obtain poorer performances in multi-source tasks, which verifies our analysis in Sect. 1 that the combining strategy may ignore the mutual interferences across source domains.
- (d) MST3L and MCPC achieve better performance than RCD-PLSA. This indicates that fusing all single-source knowledge can improve learning performance.

## (2) Comparison on Sentiment

To verify that our method is able to perform more complex work, we construct classification tasks where the discrepancies among source domains are large. The experimental results are shown in Table 4, and we can find that PSF-MSTL has more obvious advantages over all the compared methods. Other multi-source transfer learning methods fail in challenging tasks because the common and domain-specific information is extracted from a few raw features, weakening their discriminative power for document classification. Only PSF-MSTL constructs a complete relation structure of multiple source domains, which can provide complementary latent information to reduce the learning bias in a harder scenario. Thus, the tasks on

Sentiment further demonstrate that PSF-MSTL is more stable than other transfer learning methods.

## (3) Comparison on Office-Home

To further validate the adaptation capability of PSF-MSTL, we performed more validation experiments on the image dataset Office-Home. The experimental results are shown in Table 5, and we can find that PSF-MSTL can still obtain the optimal performance. The reason is that even in more challenging image datasets, PSF-MSTL can construct the power set framework that contains correlations across multiple source domains and mine complementary knowledge from it to further improve classification performance. Therefore, our algorithm can still obtain outstanding performances.

In summary, the comparison results verify the effectiveness and stability of PSF-MSTL on both traditional and challenging multi-source classification tasks.

## 5.4 Effectiveness of PSF-MSTL

To further evaluate PSF-MSTL, we design two-source and three-source experiments and perform several groups of single-source transfer learning tasks where the source domain is set as one of the generated training datasets. According to the number of training datasets in the power set framework, 864(288 × 3) and 672(96 × 7) single-source tasks are produced in two-source and three-source scenarios, respectively. The parameter settings in these tasks are the same as those in PSF-MSTL. Table 6 shows the learning performances on the complete framework and its components under the two-source scenario, and Table 7 shows the three-source one. Since we remove the duplicate tasks in Sect. 5.1,

**Table 6** Average performances (%) comparison between the complete framework and its components for two-source (10 repeated experiments)

	One-source		Two-source	PSF-MSTL
	$\mathcal{D}'_1$	$\mathcal{D}'_2$	$\mathcal{D}'_3$	
Performance of total 288 two-source tasks on 20Newsgroups				
<i>Accuracy</i>	83.47	88.76	90.07	93.45
<i>TypeAverage</i>	86.12		90.07	–
<i>F1 – Measure</i>	83.42	88.72	90.02	93.44
<i>TypeAverage</i>	86.07		90.02	–
<i>NumNT</i>	25	9	5	1
Performance of total 12 two-source tasks on Sentiment				
<i>Accuracy</i>	68.52	67.27	72.14	75.55
<i>TypeAverage</i>	67.9		72.14	–
<i>F1 – Measure</i>	68.40	67.21	71.99	75.50
<i>TypeAverage</i>	67.81		71.99	–
<i>NumNT</i>	9	9	10	4

better or worse results may gather in a certain group of tasks, and thus we give the mean accuracy of each type to make the results more objective.

First, both Tables 6 and 7 show that the learning accuracy on any training dataset cannot exceed the one on the power set framework, indicating that it is necessary to construct a complete framework that contains richer knowledge. In addition, the reason why PSF-MSTL can reduce negative transfers in two-source tasks and avoid negative transfers in

three-source tasks is that our method can build more sophisticated intrinsic relevance when there exist more source domains.

Second, by comparing the experimental results of two-source and three-source types in Table 7, we can infer that the increase of source domains may lead to different consequences in different scenarios: in traditional multi-source tasks, data expansion results in a limited development in learning performance, and in contrast, it brings a large improvement in challenging tasks. However, PSF-MSTL significantly improves the learning capabilities on both datasets, indicating that it can solve the problem of mutual interference among multiple sources and effectively utilize the valuable information in each source domain wherever it performs.

In summary, these results not only verify our ideas presented in Sect. 1 but also prove the effectiveness of PSF-MSTL.

## 5.5 Running Time

We randomly select six three-source tasks on 20Newsgroups and Sentiment, respectively, to check the running time of PSF-MSTL. The experimental results are shown in Table 8. The reason why PSF-MSTL has the longest running time is that how fast the high-level concepts-based algorithm runs has a positive correlation with the number of high-level concepts, and PSF-MSTL has to extract the latent information from more latent feature spaces. Additionally, the running time of our method is within acceptable limits.<sup>5</sup>

**Table 7** Average performances (%) comparison between the complete framework and its components for three-source (10 repeated experiments)

	One-source			Two-source			Three-source	PSF-MSTL
	$\mathcal{D}'_1$	$\mathcal{D}'_2$	$\mathcal{D}'_3$	$\mathcal{D}'_4$	$\mathcal{D}'_5$	$\mathcal{D}'_6$	$\mathcal{D}'_7$	
Performance of total 96 three-source tasks on 20Newsgroups								
<i>Accuracy</i>	82.12	87.47	89.37	89.10	92.17	89.48	91.53	95.23
<i>TypeAverage</i>	86.32			90.25			91.53	–
<i>F1 – Measure</i>	82.06	87.42	89.34	89.02	92.17	89.48	91.49	95.22
<i>TypeAverage</i>	86.27			90.22			91.49	–
<i>NumNT</i>	9	5	2	1	1	2	1	0
Performance of total 4 three-source tasks on Sentiment								
<i>Accuracy</i>	69.4	65.08	69.08	71.45	73.21	71.49	74.96	78.3
<i>TypeAverage</i>	67.85			72.05			74.96	–
<i>F1 – Measure</i>	69.29	64.94	69.05	71.22	73.17	71.46	74.90	78.28
<i>TypeAverage</i>	67.76			71.95			74.90	–
<i>NumNT</i>	3	3	3	3	4	3	2	0

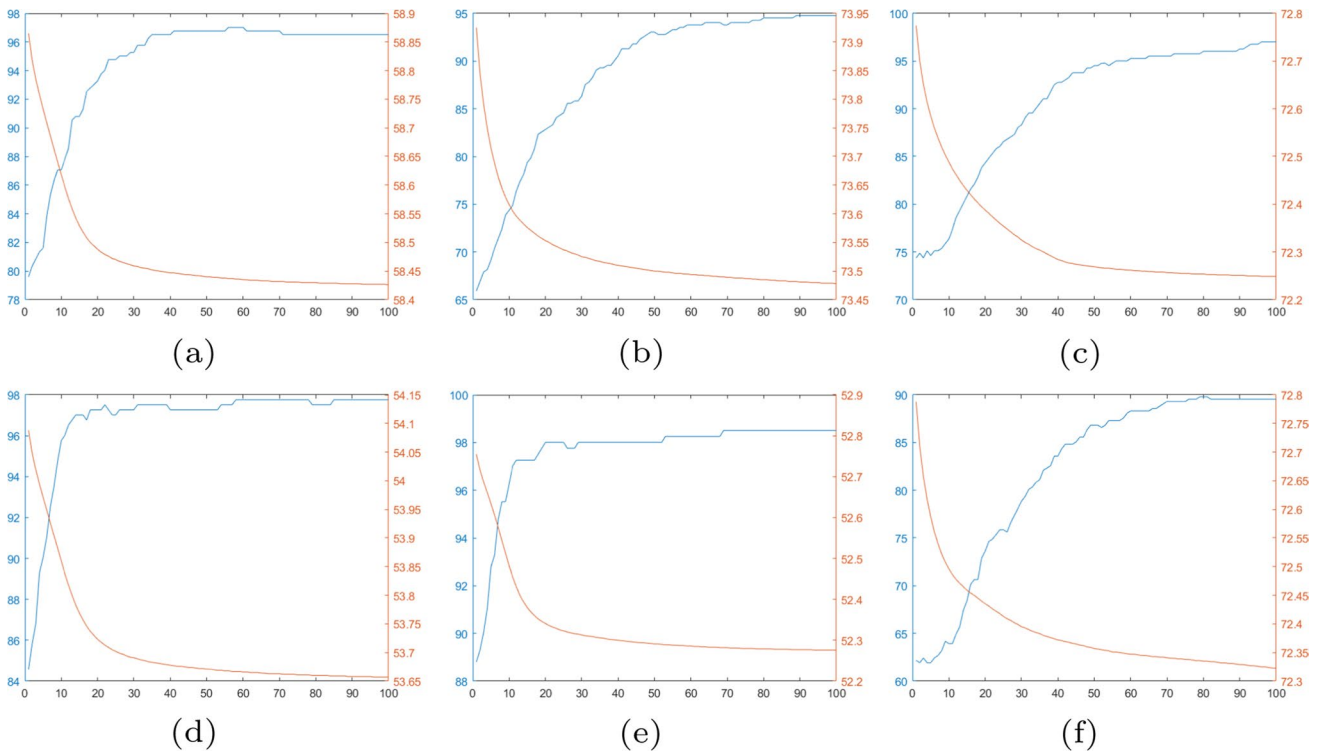
<sup>5</sup> The configuration of computing platform: Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, RAM 8.0GB.

**Table 8** Running time of PSF-MSTL and other compared methods(s)

	LR	NMTF	TSM-DNN	RCD-PLSA	MST3L	MCPC	SDA	PSF-MSTL
<i>20Newsgroups Task</i>	2.0	10.0	138.24	14.2	30.9	48.0	25.3	259.2
<i>Sentiment Task</i>	3.8	11.8	163.19	33.5	61.1	118.5	91.9	321.5

**Table 9** The parameter influence on performance (%) of algorithm PSF-MSTL

Sampling ID	$k_1$	$k_2$	Problem ID									
			1	2	3	4	5	6	7	8	9	10
1	18	22	97.26	98.76	97.01	98.51	96.27	98.01	96.02	97.76	98.01	98.26
2	25	20	96.27	98.51	97.01	98.76	96.77	98.26	96.52	96.52	98.76	98.26
3	17	25	96.52	98.01	96.77	98.51	97.26	98.26	96.27	97.76	98.76	98.01
4	19	17	96.52	98.51	96.27	98.26	96.52	98.26	96.27	97.76	98.76	97.76
5	20	19	97.01	98.51	97.51	98.76	96.77	98.26	96.02	97.26	98.01	98.01
6	24	21	96.52	98.51	97.01	98.76	97.01	98.26	96.52	97.01	98.01	97.76
7	23	19	96.52	98.76	96.77	98.26	96.27	97.76	96.27	97.76	98.51	98.01
8	16	18	97.26	98.76	96.77	97.51	97.26	97.76	96.52	98.26	98.51	98.51
9	15	23	97.01	98.26	96.52	97.76	97.01	98.01	96.27	97.01	98.51	98.51
10	24	17	96.02	98.26	96.77	98.76	96.52	98.26	95.77	97.26	98.26	98.26
Mean			96.69	98.48	96.84	98.38	96.77	98.11	96.24	97.44	98.41	98.13
Variance			0.179	0.061	0.111	0.196	0.138	0.044	0.061	0.262	0.099	0.072
This paper	20	20	96.52	98.48	96.59	98.53	96.79	97.71	96.52	97.56	98.48	97.71



**Fig. 2** The performance of PSF-MSTL and objective value vs. the number of iterations

## 5.6 Parameter Sensitivity

Here, we analyze the parameter sensitivity of PSF-MSTL. To verify that PSF-MSTL is hardly affected by changing parameters, we sample in a larger range of both  $k_1$  and  $k_2$ . To be specific, we randomly sample ten pairs of  $k_1$  and  $k_2$  when  $k_1 \in [15, 25]$ ,  $k_2 \in [15, 25]$ , and randomly choose ten three-source tasks on 20Newsgroups to make verification. The experimental results are shown in Table 9. Apparently, the mean accuracy of ten pairs of parameters is almost equal to the accuracy trained from default parameters, and the variance is small. As a result, PSF-MSTL is not sensitive to parameters.

## 5.7 Algorithm Convergence

In this section, we check the convergence of PSF-MSTL by randomly choosing six three-source tasks on rec vs. sci. The experimental results are shown in Fig. 2. The left y-axis represents the accuracy of PSF-MSTL, the right y-axis represents the objective value in Eq.(4), and the x-axis represents the number of iterations. Obviously, as the iteration develops, the accuracy of PSF-MSTL increases while the objective value decreases, and both of them converge within 100 iterations.

## 6 Conclusion

In this paper, we propose a novel multi-source transfer learning method called PSF-MSTL. First, our method can combine different source domains into various training datasets based on the power set concept, forming a source domain framework. Second, PSF-MSTL integrates all the training datasets and transfers complementary knowledge from an integral framework using a dual-promotion strategy. To solve the optimization problem, we propose an iterative algorithm based on non-negative matrix tri-factorization. Finally, we conduct extensive multi-source experiments to demonstrate that PSF-MSTL can outperform other state-of-the-art multi-source text classification methods.

It is worth mentioning that although PSF-MSTL has excellent performance, the parameters of the high-level concepts are set empirically, and the weights of different training datasets are defaulted to be equal. In the future, we will explore the automatic adjustment of the optimal parameters and consider how to weigh different types of training datasets in different scenarios to obtain better results.

**Author Contributions** Conceptualization, methodology, data curation, writing—original draft, and software were performed by BS. Conceptualization, methodology, writing—review and editing, supervision, and funding acquisition were performed by JP. Resources and writing—review and editing were performed by QQ. Validation and writing—review and editing were performed by ZL.

**Funding** This work is supported by the National Natural Science Foundation of China (No.61703187), Jiangsu Normal University (No.2021XKT1385).

**Data Availability** The datasets and derived data supporting the findings of this study are available from the corresponding author on request.

## Declarations

**Conflict of Interest** The authors declare no conflicts of interest.

**Ethical Approval** Not applicable.

**Consent to Participate** Not applicable.

**Consent for Publication** Not applicable.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Pan, S.J., Yang, Q.: A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **22**(10), 1345–1359 (2010)
- Zhuang, F. Z., Qi, Z. Y., Duan, K. Y., Xi, D. B., Zhu, Y. C., Zhu, H. S., Xiong, H., He, Q.: A comprehensive survey on transfer learning. *Proc. IEEE* **109**(1), 43–76 (2020)
- Dai, W., Xue, G. R., Duan, Q. Yang., Yu, Y.: Co-clustering based classification for out-of-domain documents, *Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp.210-219(2007)
- Jiang, J., Zhai, C. X.: A two-stage approach to domain adaptation for statistical classifiers, *Proc. of the 16th ACM conference on Conference on information and knowledge management (CIKM)*, pp.401-410(2007)
- Fang, Z., Li, Y. X. Lu, J., Dong, J. H., Han, B., Liu, F.: Is out-of-distribution detection learnable?, *Arxiv: abs/2210.14707(2022)*
- Uguroglu, S., Carbonell, J.: Feature selection for transfer learning, *Machine Learning and Knowledge Discovery in Databases*, **6913**, 430-442(2011)

7. Blitzer, J., McDonald, R., Pereira, F.: Domain adaptation with structural correspondence learning. Proc. of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.120-128(2006)
8. Zhuang, F.Z., Luo, P., Xiong, H., He, Q., Xiong, Y.H., Shi, Z.Z.: Exploiting associations between word clusters and document classes for cross-domain text categorization. *Stat. Anal. Data Min.* **4**(1), 100–114 (2011)
9. Chen, Z., Zhang W. X. : Domain Adaptation With Topic Correspondence Learning, Proc. of the 23th international joint conference on Artificial Intelligence (IJCAI), pp.1280-1286 (2013)
10. Pan, J., Hu, X., Zhang, Y., Lin, Y., Li, P., Li, H., He, W., Li, L.: Quadruple transfer learning: exploiting both shared and non-shared concepts for text classification. *Knowl. Based Syst.* **99**, 199–210 (2015)
11. Long, M., Wang, J., Ding, G., Cheng, W., Zhang, X., Wang, W.: Dual transfer learning, Proc. of the 12th SDM: SIAM International Conference on Data Mining, pp.540-551(2012)
12. Zhuang, F. Z., Luo, P., Shen, Z., He, Q., Xiong, Y., Shi, Z., Xiong, H.: Triplex transfer learning: exploiting both shared and distinct concepts for text classification. *IEEE Trans. Cybern.* **44**(7), 1191–1203 (2014)
13. Zhuang, F. Z., Luo, P., Yin, P. F., He, Q., Shi, Z. Z.: Concept Learning for Cross-Domain Text Classification: A General Probabilistic Framework, Proc. of the 23th international joint conference on Artificial Intelligence(IJCAI), pp.1960-1966(2013)
14. Zhu, Y., Chen, Y., Lu, Z., Pan, S. J., Xue, G. R., Yu, Y., Yang, Q.: He terogeneous transger learning for image classification. Proc. of the 25th AAAI **25**(1), 1304–1309 (2011)
15. Pan, J., Hu, X., Li, P., Li, H., He, W., Zhang, Y., Lin, Y.: Domain adaptation via multi-layer transfer learning. *Neurocomputing* **190**, 10–24 (2016)
16. Hu, X., Pan, J., Li, P., Li, H., He, W., Zhang, Y.: Multi-Bridge transfer learning. *Knowl. Based Syst.* **97**, 60–74 (2016)
17. Fang, Z. X., Lu, J., Liu, F., Zhang, G. Q.: Semi-supervised heterogeneous domain adaptation: theory and algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 1087–1105 (2022)
18. Fang, Z., Lu, J., Liu, F., Xuan, J.Y., Zhang, G. Q.: Open set domain adaptation: theoretical bound and algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **32**, 4309–4322 (2019)
19. Z. Fang, J. Lu, A. Liu, F. Liu, and G. Q. Zhang, Learning Bounds for Open-Set Learning, International Conference on Machine Learning (2021)
20. Xu, R. J., Chen, Z. L., Zuo, W. M., Yan, J.J., Lin, L.: Deep Cocktail Network: Multi-source Unsupervised Domain Adaptation with Category Shift, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.3964-3973(2018)
21. Zhao, H., Zhang, S.H., Wu, G. H., Costeira, J. P., Moura, J. M. F., Gordon, G. J.: Multiple source domain adaptation with adversarial learning, International Conference on Learning Representations (ICLR), pp.1-24(2018)
22. Zhu, Y. C., Zhuang, F. Z., Wang, D. Q. : Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources, The Thirty-Third AAAI Conference on Artificial Intelligence 33, pp.5989-5996(2019)
23. Cheng, Y.H., Wang, X.S., Cao, G.: Multi-source tri-training transfer learning. *IEICE Trans. Inf. Syst.* **97**, 1668–1672 (2014)
24. Dai, Y., Liu, J., Zhang, J., Fu, H. G., Xu, Z. L.: Unsupervised sentiment analysis by transferring multi-source knowledge. *Cogn. Comput.* **13**, 1185–1197 (2021)
25. Zhang, D., Ye, M., Liu, Y., Xiong, L., Zhou, L.: Multi-source unsupervised domain adaptation for object detection. *Inf. Fusion* **78**, 138–148 (2022)
26. Dredze, M., Kulesza, A., Crammer, K.: Multi-domain learning by confidence-weighted parameter combination. *Mach. Learn.* **79**(1–2), 123–149 (2010)
27. Xu, Y., Kan, M., Shan, S., Chen, X.: Mutual Learning of Joint and Separate Domain Alignments for Multi-Source Domain Adaptation, Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp.1890-1899(2022)
28. Sun, S., Shi, H., Wu, Y.: A survey of multi-source domain adaptation. *Inf. Fusion* **24**, 84–92 (2015)
29. Liu, Z., Yang, J.A., Liu, H., Liu, J.: Learning from multiple sources via multiple domain relationship. *IEICE Trans. Inf. Syst.* **E99D**, 1941–1944 (2016)
30. Huang, P.P., Wang, G., Qin, S.Y.: Boosting for transfer learning from multiple data sources. *Pattern Recogn. Lett.* **33**, 568–579 (2010)
31. Ding, Z.M., Shao, M., Fu, Y.: Incomplete multisource transfer learning. *IEEE Trans. Neural Netw. Learn. Syst.* **29**, 310–323 (2016)
32. Fang, M., Guo, Y., Zhang, X.S., Li, X.: Multi-source transfer learning based on label shared subspace. *Pattern Recogn. Lett.* **51**, 101–106 (2015)
33. Lu, J., Zuo, H., Zhang, G.Q.: Fuzzy multiple-source transfer learning. *IEEE Trans. Fuzzy Syst.* **28**, 3418–3431 (2019)
34. Chandra, R., Kapoor, A.: Bayesian neural multi-source transfer learning. *Neurocomputing* **378**, 54–64 (2020)
35. Wu, Q.Y., Zhou, X.M., Yan, Y.G., Wu, H.R., Min, H.Q.: Online transfer learning by leveraging multiple source domains. *Knowl. Inf. Syst.* **52**, 687–707 (2017)
36. F. Wu and Y. Huang.: Sentiment domain adaptation with multiple sources, Proceedings of the 54th annual meeting of the association for computational linguistics 1: long papers, pp.301-310 (2016)
37. S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira and J.W. Vaughan.: A theory of learning from different domains, Proceedings of the 54th annual meeting of the association for computational linguistics 79(1-2), pp.151-175 (2010)
38. Zhuang, F.Z., Luo, P., Shen, Z., He, Q., Xiong, Y., Shi, Z., Xiong, H.: Mining distinction and commonality across multiple domains using generative model for text classification. *IEEE Trans. Knowl. Data Eng.* **24**(11), 2025–2039 (2012)
39. Li, K., Lu, J., Zuo, H., Zhang, G.: Dynamic classifier alignment for unsupervised multi-source domain adaptation. *IEEE Trans. Knowl. Data Eng.* **35**(5), 4727–4740 (2023)
40. Wu, H., Ng, M.K.: Multiple graphs and low-rank embedding for multi-source heterogeneous domain adaptation. *ACM Trans. Knowl. Discov. Data (TKDD)* **16**(4), 1–25 (2022)
41. Kim, H., Lim, P.: A tensor space model-based deep neural network for text classification. *Appl. Sci.* **9703**(11), 177–196 (2021)
42. Pan, J., Cui, T., Le Duy, T., Li, X., Zhang, J.: Multi-group transfer learning on multiple latent spaces for text classification. *IEEE Access* **8**, 64120–64130 (2020)
43. Hosmer, D., Lemeshow, S.: Applied logistic regression. John Wiley, Hoboken (2004)
44. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.* **42**(1–2), 177–196 (2001)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations