Learning the Shared Subspace for Multi-Task Clustering and Transductive Transfer Classification

Quanquan Gu and Jie Zhou
State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Automation, Tsinghua University, Beijing, China 100084
gqq03@mails.tsinghua.edu.cn, jzhou@tsinghua.edu.cn

***Abstract*—There are many clustering tasks which are closely related in the real world, e.g. clustering the web pages of different universities. However, existing clustering approaches neglect the underlying relation and treat these clustering tasks either individually or simply together. In this paper, we will study a novel clustering paradigm, namely multi-task clustering, which performs multiple related clustering tasks together and utilizes the relation of these tasks to enhance the clustering performance. We aim to learn a subspace shared by all the tasks, through which the knowledge of the tasks can be transferred to each other. The objective of our approach consists of two parts: (1) Within-task clustering: clustering the data of each task in its input space individually; and (2) Cross-task clustering: simultaneous learning the shared subspace and clustering the data of all the tasks together. We will show that it can be solved by alternating minimization, and its convergence is theoretically guaranteed. Furthermore, we will show that given the labels of one task, our multi-task clustering method can be extended to transductive transfer classification (a.k.a. cross-domain classification, domain adaption). Experiments on several cross-domain text data sets demonstrate that the proposed multi-task clustering outperforms traditional single-task clustering methods greatly. And the transductive transfer classification method is comparable to or even better than several existing transductive transfer classification approaches.**

***Keywords*-multi-task clustering; transductive transfer classification; multi-task learning; transfer learning; cross domain classification; domain adaption**

I. Introduction

Clustering has a long history in the machine learning literature. It aims to partition data points into groups, so that the data points in the same group are relatively similar, while the data points in different groups are relatively dissimilar. In the past decades, incorporating prior knowledge into clustering has witnessed increasing interest, e.g. semi-supervised clustering [1] [2] [3] [4] [5] and co-clustering [6] [7] [8] [9]. However, all these methods are limited to a single task, where i.i.d. assumption of the data samples holds. We refer them as *single-task clustering*.

There are many different but related data sets in real applications. For example, we have web pages from 4 universities, e.g. Cornell, Texas, Wisconsin and Washington. And we are going to cluster the web pages of each university into 7 categories, e.g. student, faculty, staff, department, course, project and the other. In this scenario, clustering the web pages of each university can be seen as a task. Our intuition tells us that the 4 clustering tasks are related, since the sources and contents of their data are similar. However, the distributions of their data should be different, since different universities exhibit different features. Imagine that if we have limited web pages in one university, typical clustering methods may fail to discover the correct clusters. In this case, one may argue to use the web pages from the other universities as an auxiliary data to inform the correct clusters. However, simply combining them together followed with traditional single-task clustering approach does not necessarily lead to performance improvement, because their distributions are different, which violates the i.i.d. assumption in single-task clustering. To address this problem, new clustering paradigm is imperative, which can utilize the relation of different tasks to enhance clustering as well as overcome the non i.i.d. problem.

In this paper, based on the observations mentioned above, we will study a novel clustering paradigm, namely *multi-task clustering*, which can exploit the knowledge shared by multiple related tasks. It falls in the field of multi-task learning [10] [11] [12] [13] [14] [15], which says learning multiple related tasks together may achieve better performance than learning these tasks individually, provided that we can exploit the underlying relation. The *assumption* of our multi-task clustering is that there is a common underlying subspace shared by the multiple related tasks. This underlying subspace can be seen as a new feature representation, in which the data distributions of the related tasks are close to each other. Hence single-task clustering algorithm can be applied in this shared subspace. Similar assumption has also been made in several multi-task classification approaches [11] [13] [14] [15]. Based on the above assumption, we propose a multi-task clustering method. It aims to learn a subspace shared by all the tasks, through which the knowledge of one task can be transferred to another. And the objective of our approach consists of two parts: (1) *Within-task* clustering: clustering the data of each task in its input space individually; and (2) *Cross-task* clustering: simultaneous learning the shared subspace

and clustering the data of all the tasks. Our approach not only utilizes the knowledge in each individual task as traditional clustering method does, but also make use of the knowledge shared by the related tasks which may benefit the clustering performance. We will show that it can be solved via alternating minimization, and its convergence is theoretically guaranteed. To the best of our knowledge, this is the *first* work addressing multi-task clustering.

Furthermore, we will show that provided with the labels of one task, our multi-task clustering method turns out to be a transductive transfer classification method (a.k.a. cross-domain classification or domain adaption), in which the label of a source task (in-domain) is available as prior knowledge, and we aim to utilize this prior knowledge to predict the labels of the data in a related target task (out-of-domain). Experiments on several cross-domain text data sets demonstrate that the proposed multi-task clustering method outperforms traditional single-task clustering methods greatly. And the transductive transfer classification method is comparable to or even better than several existing transductive transfer classification approaches.

The remainder of this paper is organized as follows. In Section II, we will review some related works. In Section III we will propose the multi-task clustering algorithm. In Section IV, we will extend the multi-task clustering method to transfer clustering setting. Experiments on text data sets are demonstrated in Section V. Finally, we draw a conclusion in Section VI and point out the future works.

II. Related Works

In this section, we will review some works related with ours.

A. Multi-Task Learning

Empirical work has shown that learning multiple related tasks from data simultaneously can be advantageous in terms of predictive performance, relative to learning these tasks independently. This motivates multi-task learning [10] [11] [12] [13] [14] [15]. However, existing multi-task learning methods all tackle classification, in which each task has both labeled and unlabeled data, and the goal is to predict the class labels of unlabeled data in each task by utilizing within-task and cross-task knowledge. In this paper, we consider multi-task clustering, where the data in each task are all unlabeled, and it aims at predicting the cluster labels of the data in each task.

B. Transfer Learning

Transfer learning [16] [17] is closely related with multi-task learning. It tackles the transfer of knowledge across tasks, domains, categories and distributions that are similar but not the same. In this paper, we refer task and domain as the same thing. Transfer learning is closely related with multi-task learning, with the difference that in multi-task learning, the learner focuses on enhancing the performance of all the tasks, while in transfer learning, the learner only focuses on improving the performance of a so-called target task (out-of-domain) by using the knowledge from a so-called source task (in-domain). Transfer learning can be categorized as (1) inductive transfer: there are a few labeled data in the target task, while there are a large amount of labeled [18] [19] [20] or unlabeled [21] data in the source task, (2) transductive transfer: there are no labeled data in the target task, while there are large amount of labeled data in the source task [22] [23] [24] [25], this is also called cross-domain classification or domain adaption, and (3) Unsupervised transfer: there are no labeled data in the target task, while there are large amount of unlabeled data in the source task [26]. In our study, we focus on transductive transfer classification, which belongs to the second category.

C. Clustering with Background and Prior Knowledge

Improving clustering performance using the background and prior knowledge has witnessed increasing interest in the past decade. One direction is co-clustering [6] [7] [8] [9], which clusters the data and features simultaneously to enhance the clustering performance. Another direction is semi-supervised clustering [1] [2] [3] [4] [5], which incorporates pairwise constraints, e.g. must-link and cannot-link constraints, to assist clustering. Both co-clustering and semi-supervised clustering use either the background or prior knowledge within a single task. However, multi-task clustering exploits both in-task and out-of-task knowledge.

D. Semi-Supervised Learning

In many practical machine learning problems, the acquisition of sufficient labeled data is often expensive and/or time consuming. On the contrary, in many cases, large number of unlabeled data are far easier to obtain. Consequently, semi-supervised learning [27] [28] [29], which aims to learn from both labeled and unlabeled data points, has received significant attention in the past decade. Semi-supervised learning is different from transductive transfer classification. In semi-supervised learning, the labeled and unlabeled samples are drawn from the same task, so their distributions are the same. However, in transductive transfer classification, the labeled samples are from the source task, while the unlabeled samples are from the target task. So their distributions are different.

III. Multi-Task Clustering

In this section, we first present the problem setting of multi-task clustering. Then we propose a multi-task clustering method and the optimization algorithm, followed with its convergence analysis.

A. Problem Formulation

Suppose we are given m clustering tasks, each with a set of data points, i.e. $\mathcal{X}^{(k)} = \{\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \ldots, \mathbf{x}_{n_k}^{(k)}\} \in \mathbb{R}^d, 1 \leq k \leq m$, where n_k is the number of data points in the k-th task. The goal of multi-task clustering is to partition the data set $\mathcal{X}^{(k)}$ of each task into c clusters $\{\mathcal{C}_j^{(k)}\}_{j=1}^c$. Note that we assume the dimensionality of the feature vector of all the tasks is the same, i.e. d. It is appropriate since we could augment the feature vectors of all the tasks to make the dimensionality same. In fact, the bag-of-words document representation used in our experiments implicitly does the augmentation. Moreover, we assume that the number of clusters in each task is the same, i.e. $c_1 = c_2 = \ldots = c_m = c$, which is also assumed in existing multi-task learning literature.

B. Objective

Let us consider the case of single-task clustering first. Take the k-th task for example. We are going to partition the k-th data set into c clusters. The classical K-means algorithm achieves this goal by minimizing the following objective

$$J_{st} = \sum_{j=1}^{c} \sum_{\mathbf{x}_i^{(k)} \in \mathcal{C}_j^{(k)}} ||\mathbf{x}_i^{(k)} - \mathbf{m}_j^{(k)}||_2^2, \tag{1}$$

where $||\cdot||_2$ is 2-norm and $\mathbf{m}_j^{(k)}$ is the mean of cluster $\mathcal{C}_j^{(k)}$ in the k-th task. If we define $\mathbf{M}^{(k)} = [\mathbf{m}_1^{(k)}, \ldots, \mathbf{m}_c^{(k)}] \in \mathbb{R}^{d \times c}$, then Eq.(1) can be rewritten as

$$\begin{aligned} J_{st} &= ||\mathbf{X}^{(k)} - \mathbf{M}^{(k)}\mathbf{P}^{(k)T}||_F^2, \\ &\text{s.t.} \quad \mathbf{P}^{(k)} \in \{0,1\}^{n_k \times c} \end{aligned} \tag{2}$$

where $\mathbf{X}^{(k)} = [\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_{n_k}^{(k)}], 1 \leq k \leq m$, $||\cdot||_F$ is Frobenius norm and $\mathbf{P}^{(k)} \in \{0,1\}^{n_k \times c}$ is called partition matrix, which represents the clustering assignment, such that $\mathbf{P}_{ij}^{(k)} = 1$ if $\mathbf{x}_i^{(k)}$ belongs to cluster $\mathcal{C}_j^{(k)}$ and $\mathbf{P}_{ij}^{(k)} = 0$ otherwise. This is also known as *hard* clustering, i.e. the cluster assignment is binary.

When it comes to multi-task clustering setting, we are going to learn a shared subspace, which is obtained by an orthonormal projection $\mathbf{W} \in \mathbb{R}^{d \times l}$, across all the related tasks, in which we perform all the clustering tasks together. This shared subspace can be seen as a new feature space, in which the data distribution from all the tasks are similar with each other. As a result, we can cluster them together in this shared subspace using traditional single-task clustering algorithm, i.e. K-means. Furthermore, we add a constraint that the clustering result of each task in the shared subspace is the same as that in the input subspace, which intertwines the clustering in the input space and clustering in the shared subspace. Then it is formulated as minimizing

$$\begin{aligned} J_{mt} &= \lambda \sum_{k=1}^{m} ||\mathbf{X}^{(k)} - \mathbf{M}^{(k)}\mathbf{P}^{(k)T}||_F^2 \\ &+ (1-\lambda) \sum_{k=1}^{m} ||\mathbf{W}^T\mathbf{X}^{(k)} - \mathbf{M}\mathbf{P}^{(k)T}||_F^2 \\ &\text{s.t.} \quad \mathbf{W}^T\mathbf{W} = \mathbf{I}, \mathbf{P}^{(k)} \in \{0,1\}^{n_k \times c} \end{aligned} \tag{3}$$

where $\lambda \in [0,1]$ is a regularization parameter balancing the clustering in the input space and the clustering in the shared subspace, $\mathbf{I}$ is an identity matrix, and $\mathbf{M} = [\mathbf{m}_1, \ldots, \mathbf{m}_c] \in \mathbb{R}^{m \times c}$ with $\mathbf{m}_j$ is the mean of cluster $\mathcal{C}_j$ of all the tasks in the shared subspace.

The objective in Eq.(3) consists of two terms. The first term is *Within-task* clustering, which includes k independent k-means clustering of each task in the input space. The second term is *Cross-task* clustering, which simultaneously learns the shared subspace and clusters the data of all the tasks together in the shared subspace. It is worth noting that the second term is similar with clustering the data of all the tasks together via *Adaptive Subspace Iteration* (ASI) clustering method [30]. The first term and the second term are intertwined through the partition matrices. When letting $\lambda = 1$, Eq.(3) degenerates to m independent K-means clustering. And When letting $\lambda = 0$, Eq.(3) turns out to be clustering data of all the tasks via ASI. In general case, the more related the tasks are, the smaller λ we will set.

By its definition, the elements in $\mathbf{P}^{(k)}$ can only take binary values, which makes the minimization in Eq.(3) very difficult, therefore we relax $\mathbf{P}^{(k)}$ into nonnegative continuous domain. Then the objective of multi-task clustering in Eq.(3) turns out to be

$$\begin{aligned} J_{mt} &= \lambda \sum_{k=1}^{m} ||\mathbf{X}^{(k)} - \mathbf{M}^{(k)}\mathbf{P}^{(k)T}||_F^2 \\ &+ (1-\lambda) \sum_{k=1}^{m} ||\mathbf{W}^T\mathbf{X}^{(k)} - \mathbf{M}\mathbf{P}^{(k)T}||_F^2 \\ &\text{s.t.} \quad \mathbf{W}^T\mathbf{W} = \mathbf{I}, \mathbf{P}^{(k)} \geq 0. \end{aligned} \tag{4}$$

We call Eq.(4) Learning the Shared Subspace for Multi-Task Clustering (LSSMTC).

C. Optimization

As we see, minimizing Eq.(4) is with respect to $\mathbf{M}^{(k)}, \mathbf{P}^{(k)}, \mathbf{W}$ and $\mathbf{M}$. And we cannot give a closed-form solution. In the following, we will present an alternating minimization algorithm to optimize the objective. In other words, we will optimize the objective with respect to one variable when fixing the other variables.

1) Computation of $\mathbf{M}$*:* Given $\mathbf{W}$ and $\mathbf{P}^{(k)}$, optimizing Eq.(4) with respect to $\mathbf{M}$ is equivalent to optimizing

$$\begin{aligned} J_1 &= \sum_{k=1}^{m} ||\mathbf{W}^T\mathbf{X}^{(k)} - \mathbf{M}\mathbf{P}^{(k)T}||_F^2 \\ &= ||\mathbf{W}^T\mathbf{X} - \mathbf{M}\mathbf{P}^T||_F^2 \end{aligned} \quad (5)$$

where $\mathbf{X} = [\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m)}]$ and $\mathbf{P} = [\mathbf{P}^{(1)T}, \ldots, \mathbf{P}^{(m)T}]^T$.

Setting $\frac{\partial J_1}{\partial \mathbf{M}} = 0$, we obtain

$$\mathbf{M} = \mathbf{W}^T\mathbf{X}\mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}. \quad (6)$$

2) Computation of $\mathbf{M}^{(k)}$*:* Given $\mathbf{P}^{(k)}$, optimizing Eq.(4) with respect to $\mathbf{M}^{(k)}$ is equivalent to optimizing

$$J_2 = ||\mathbf{X}^{(k)} - \mathbf{M}^{(k)}\mathbf{P}^{(k)T}||_F^2 \quad (7)$$

Setting $\frac{\partial J_2}{\partial \mathbf{M}^{(k)}} = 0$, we obtain

$$\mathbf{M}^{(k)} = \mathbf{X}^{(k)}\mathbf{P}^{(k)}(\mathbf{P}^{(k)T}\mathbf{P}^{(k)})^{-1}. \quad (8)$$

3) Computation of $\mathbf{P}^{(k)}$*:* Given $\mathbf{W}, \mathbf{M}, \mathbf{M}^{(k)}$, optimizing Eq.(4) with respect to $\mathbf{P}^{(k)}$ is equivalent to optimizing

$$\begin{aligned} J_3 &= \lambda||\mathbf{X}^{(k)} - \mathbf{M}^{(k)}\mathbf{P}^{(k)T}||_F^2 \\ &+ (1-\lambda)||\mathbf{W}^T\mathbf{X}^{(k)} - \mathbf{M}\mathbf{P}^{(k)T}||_F^2 \\ &\text{s.t.} \quad \mathbf{P}^{(k)} \geq 0, \end{aligned} \quad (9)$$

For the constraint $\mathbf{P}^{(k)} \geq 0$, we cannot get a closed-form solution of $\mathbf{P}^{(k)}$. In the following, we will present an iterative solution. We introduce the Lagrangian multiplier $\boldsymbol{\gamma} \in \mathbb{R}^{n_k \times c}$, and the Lagrangian function is

$$\begin{aligned} L(\mathbf{P}^{(k)}) &= \lambda||\mathbf{X}^{(k)} - \mathbf{M}^{(k)}\mathbf{P}^{(k)T}||_F^2 \\ &+ (1-\lambda)||\mathbf{W}^T\mathbf{X}^{(k)} - \mathbf{M}\mathbf{P}^{(k)T}||_F^2 \\ &- \text{tr}(\boldsymbol{\gamma}\mathbf{P}^{(k)T}) \end{aligned} \quad (10)$$

Setting $\frac{\partial L(\mathbf{P}^{(k)})}{\partial \mathbf{P}^{(k)}} = 0$, we obtain

$$\boldsymbol{\gamma} = -2\mathbf{A} + 2\mathbf{P}^{(k)}\mathbf{B} \quad (11)$$

where $\mathbf{A} = \lambda\mathbf{X}^{(k)T}\mathbf{M}^{(k)} + (1-\lambda)\mathbf{X}^{(k)T}\mathbf{W}\mathbf{M}$ and $\mathbf{B} = \lambda\mathbf{M}^{(k)T}\mathbf{M}^{(k)} + (1-\lambda)\mathbf{M}^T\mathbf{M}$.

Using the Karush-Kuhn-Tucker condition [31] $\boldsymbol{\gamma}_{ij}\mathbf{P}_{ij}^{(k)} = 0$, we get

$$[-\mathbf{A} + \mathbf{P}^{(k)}\mathbf{B}]_{ij}\mathbf{P}_{ij}^{(k)} = 0 \quad (12)$$

Introduce $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-$ and $\mathbf{B} = \mathbf{B}^+ - \mathbf{B}^-$ where $\mathbf{A}_{ij}^+ = (|\mathbf{A}_{ij}| + \mathbf{A}_{ij})/2$ and $\mathbf{A}_{ij}^- = (|\mathbf{A}_{ij}| - \mathbf{A}_{ij})/2$ [32], we obtain

$$[\mathbf{A}^- + \mathbf{P}^{(k)}\mathbf{B}^+ - \mathbf{A}^+ - \mathbf{P}^{(k)}\mathbf{B}^-]_{ij}\mathbf{P}_{ij}^{(k)} = 0 \quad (13)$$

Eq.(13) leads to the following updating formula

$$\mathbf{P}_{ij}^{(k)} \leftarrow \mathbf{P}_{ij}^{(k)}\sqrt{\frac{[\mathbf{A}^+ + \mathbf{P}^{(k)}\mathbf{B}^-]_{ij}}{[\mathbf{A}^- + \mathbf{P}^{(k)}\mathbf{B}^+]_{ij}}} \quad (14)$$

4) Computation of $\mathbf{W}$*:* Given $\mathbf{M}, \mathbf{M}^{(k)}, \mathbf{P}^{(k)}$, optimizing Eq.(4) with respect to $\mathbf{W}$ is equivalent to optimizing

$$\begin{aligned} J_4 &= \sum_{k=1}^{m} ||\mathbf{W}^T\mathbf{X}^{(k)} - \mathbf{M}\mathbf{P}^{(k)T}||_F^2 \\ &= ||\mathbf{W}^T\mathbf{X} - \mathbf{M}\mathbf{P}^T||_F^2 \\ &\text{s.t.} \quad \mathbf{W}^T\mathbf{W} = \mathbf{I} \end{aligned} \quad (15)$$

where $\mathbf{X} = [\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m)}]$ and $\mathbf{P} = [\mathbf{P}^{(1)T}, \ldots, \mathbf{P}^{(m)T}]^T$.

Substituting $\mathbf{M} = \mathbf{W}^T\mathbf{X}\mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}$ into the above equation, we obtain

$$\begin{aligned} J_5 &= \text{tr}(\mathbf{W}^T\mathbf{X}(\mathbf{I} - \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T)\mathbf{X}^T\mathbf{W}) \\ &\text{s.t.} \quad \mathbf{W}^T\mathbf{W} = \mathbf{I} \end{aligned} \quad (16)$$

It is easy to show that the optimal $\mathbf{W}$ minimizing Eq.(16) is composed of the eigenvectors of $\mathbf{X}(\mathbf{I} - \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T)\mathbf{X}^T$ corresponding to the l smallest eigenvalues.

In summary, we present the algorithm of optimizing Eq.(4) in Algorithm 1.

Algorithm 1 Learning the Shared Subspace for Multi-Task Clustering (LSSMTC)

Input:m tasks, $\{\mathbf{X}^{(k)}\}_{k=1}^m$, the dimensionality of the shared subspace l, maximum number of iterations T;
Output: Partitions $\mathbf{P}^{(k)} \in \mathbb{R}^{n\times c}, 1 \leq k \leq m$;
Initialize $t = 0$ and $\mathbf{P}^{(k)}, 1 \leq k \leq m$ using K-means;
Initialize $\mathbf{W} \in \mathbb{R}^{d\times l}$ using any orthonormal matrix.
while not convergent **and** $t \leq T$ **do**
  $\mathbf{M} = \mathbf{W}^T(\mathbf{X}\mathbf{P})(\mathbf{P}^T\mathbf{P})^{-1}$;
  for $k = 1$ **To** m **do**
    Compute $\mathbf{M}^{(k)} = \mathbf{X}^{(k)}\mathbf{P}^{(k)}(\mathbf{P}^{(k)T}\mathbf{P}^{(k)})^{-1}$;
    Update $\mathbf{P}_{ij}^{(k)} \leftarrow \mathbf{P}_{ij}^{(k)}\sqrt{\frac{[\mathbf{A}^+ + \mathbf{P}^{(k)}\mathbf{B}^-]_{ij}}{[\mathbf{A}^- + \mathbf{P}^{(k)}\mathbf{B}^+]_{ij}}}$;
  end for
  Compute $\mathbf{W}_{ij}$ by eigen-decomposition of $\mathbf{X}(\mathbf{I} - \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T)\mathbf{X}^T$;
  $t = t + 1$
end while

D. Convergence Analysis

In the following, we will investigate the convergence of Algorithm 1. We use the auxiliary function approach [33] to analyze the multiplicative updating formulas. Here we first introduce the definition of auxiliary function [33].

Definition III.1. *[33]* $Z(h, h')$ *is an auxiliary function for* $F(h)$ *if the conditions*

$$Z(h, h') \geq F(h), Z(h, h) = F(h),$$

are satisfied.

Lemma III.2. *[33] If Z is an auxiliary function for F, then F is non-increasing under the update*

$$h^{(t+1)} = \arg\min_{h} Z(h, h^{(t)})$$

Proof: $F(h^{(t+1)}) \leq Z(h^{(t+1)}, h^{(t)}) \leq Z(h^{(t)}, h^{(t)}) = F(h^{(t)})$ ■

Lemma III.3. *[32] For any nonnegative matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{k\times k}$, $\mathbf{S} \in \mathbb{R}^{n\times k}$,$\mathbf{S}' \in \mathbb{R}^{n\times k}$, and $\mathbf{A}$, $\mathbf{B}$ are symmetric, then the following inequality holds*

$$\sum_{i=1}^{n}\sum_{p=1}^{k} \frac{(\mathbf{AS}'\mathbf{B})_{ip}\mathbf{S}_{ip}^2}{\mathbf{S}'_{ip}} \geq tr(\mathbf{S}^T\mathbf{ASB})$$

Theorem III.4. *Let*

$$J(\mathbf{P}^{(k)}) = tr(\mathbf{P}^{(k)}\mathbf{B}\mathbf{P}^{(k)T} - 2\mathbf{A}\mathbf{P}^{(k)T}) \tag{17}$$

Then the following function

$$\begin{aligned} Z(\mathbf{P}^{(k)}, \mathbf{P}'^{(k)}) &= \sum_{ij} \frac{(\mathbf{P}'^{(k)}\mathbf{B}^+)_{ij}\mathbf{P}_{ij}^{(k)2}}{\mathbf{P}'^{(k)}_{ij}} \\ &- \sum_{ijk} \mathbf{B}^-_{jk}\mathbf{P}'^{(k)}_{ij}\mathbf{P}'^{(k)}_{ik}(1 + \log\frac{\mathbf{P}^{(k)}_{ij}\mathbf{P}^{(k)}_{ik}}{\mathbf{P}'^{(k)}_{ij}\mathbf{P}'^{(k)}_{ik}}) \\ &- 2\sum_{ij} \mathbf{A}^+_{ij}\mathbf{P}'^{(k)}_{ij}(1 + \log\frac{\mathbf{P}^{(k)}_{ij}}{\mathbf{P}'^{(k)}_{ij}}) \\ &+ 2\sum_{ij} \mathbf{A}^-_{ij}\frac{\mathbf{P}_{ij}^{(k)2} + \mathbf{P}'^{(k)2}_{ij}}{2\mathbf{P}'^{(k)}_{ij}} \end{aligned}$$

is an auxiliary function for $J(\mathbf{P}^{(k)})$. Furthermore, it is a convex function in $\mathbf{P}^{(k)}$ and its global minimum is

$$\mathbf{P}^{(k)}_{ij} = \mathbf{P}^{(k)}_{ij}\sqrt{\frac{[\mathbf{A}^+ + \mathbf{P}^{(k)}\mathbf{B}^-]_{ij}}{[\mathbf{A}^- + \mathbf{P}^{(k)}\mathbf{B}^+]_{ij}}} \tag{18}$$

Proof: Please see Appendix. ■

Theorem III.5. *Updating $\mathbf{P}^{(k)}$ using Algorithm 1 will monotonically decrease the value of the objective in Eq.(4), the objective is invariant under the updating if and only if $\mathbf{P}^{(k)}$ is at a stationary point.*

Proof: By Lemma III.2 and Theorem III.4, we can get that $J(\mathbf{P}^{(k)0}) = Z(\mathbf{P}^{(k)0}, \mathbf{P}^{(k)0}) \geq Z(\mathbf{P}^{(k)1}, \mathbf{P}^{(k)0}) \geq J(\mathbf{P}^{(k)1}) \geq \dots$ So $J(\mathbf{P}^{(k)})$ is monotonically decreasing. Since $J(\mathbf{P}^{(k)})$ is obviously bounded below, we prove this theorem. ■

In addition to Theorem III.5, since the computation of $\mathbf{W}$ in Eq.(16) also monotonically decreases the value of the objective in Eq.(4), Algorithm 1 is guaranteed to converge.

IV. Transductive Transfer Classification

In this section, we will show that given the labels of one task, the proposed multi-task clustering method turns out to be a transductive transfer classification method.

For simplicity, we consider the 2 tasks case, $\mathcal{X}^{(k)} = \{\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \dots, \mathbf{x}_{n_k}^{(k)}\}, k = 1, 2$, where n_k is the number of data points in the k-th task. Without loss of generality, we assume the label of the 1st task is given, and we are going to predict the labels of the data in the 2nd task. This problem is exactly transductive transfer classification, which is also known as domain adaption or cross-domain classification. We call the 1st task source task (in-domain), and the 2nd task target task (out-of-domain). Denote $\mathbf{X}^{(k)} = [\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_{n_k}^{(k)}], k = 1, 2$. Again, we assume that the number of classes in each task is the same, i.e. $c_1 = c_2 = c$. Note that it is trivial to generalize our transductive transfer classification method from 1 source task to more than 1 source task.

Based on the above discussion, our multi-task clustering method can be extended to transductive transfer classification as follows,

$$\begin{aligned} J_{tc} &= \lambda||\mathbf{X}^{(2)} - \mathbf{M}^{(2)}\mathbf{P}^{(2)T}||_F^2 \\ &+ (1-\lambda)\sum_{k=1}^{2}||\mathbf{W}^T\mathbf{X}^{(k)} - \mathbf{M}\mathbf{P}^{(k)T}||_F^2 \\ &\text{s.t.} \quad \mathbf{W}^T\mathbf{W} = \mathbf{I}, \mathbf{P}^{(2)} \in \{0,1\}^{n_2\times c} \end{aligned} \tag{19}$$

where the first term is clustering in the input space of the target task, the second and the third term are clustering of the source and the target task together in the shared subspace, $\lambda \in [0, 1]$ is a regularization parameter balancing the clustering in the input space and the clustering in the shared subspace. It should be noted that $\mathbf{P}^{(1)}$ in Eq.(19) is a constant since the label of the source task has been known as prior knowledge.

Again, we relax $\mathbf{P}^{(2)}$ into nonnegative continuous domain. Then the objective of transductive transfer classification in Eq.(19) turns out to be

$$\begin{aligned} J_{tc} &= \lambda||\mathbf{X}^{(2)} - \mathbf{M}^{(2)}\mathbf{P}^{(2)T}||_F^2 \\ &+ (1-\lambda)\sum_{k=1}^{2}||\mathbf{W}^T\mathbf{X}^{(k)} - \mathbf{M}\mathbf{P}^{(k)T}||_F^2 \\ &\text{s.t.} \quad \mathbf{W}^T\mathbf{W} = \mathbf{I}, \mathbf{P}^{(2)} \geq 0, \end{aligned} \tag{20}$$

We call Eq.(20) Learning the Shared Subspace for Transductive Transfer Classification (LSSTTC).

Since the optimization of Eq.(20) is very similar with that of Eq.(4), we omit the derivation of the optimization algorithm, and present it in Algorithm 2 directly.

The convergence of Algorithm 2 is also theoretically guaranteed. The proof of convergence is very similar with that of Algorithm 1.

Algorithm 2 Learning the Shared Subspace for Transductive Transfer Classification (LSSTTC)

Input: $\mathbf{X}^{(1)}, \mathbf{P}^{(1)}, \mathbf{X}^{(2)}$, the dimensionality of the shared subspace l, maximum number of iterations T;
Output: Partitions $\mathbf{P}^{(2)} \in \mathbb{R}^{n \times c}$;
Initialize $\mathbf{P}^{(2)}$ using K-means;
Initialize $\mathbf{W}$ using any orthonormal matrix.
while not convergent **and** $t \leq T$ **do**
  Compute $\mathbf{M} = \mathbf{W}^T\mathbf{X}\mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}$;
  Compute $\mathbf{M}^{(2)} = \mathbf{W}^T\mathbf{X}^{(2)}\mathbf{P}^{(2)}(\mathbf{P}^{(2)T}\mathbf{P}^{(2)})^{-1}$;
  Update $\mathbf{P}_{ij}^{(2)} \leftarrow \mathbf{P}_{ij}^{(2)} \sqrt{\frac{[\mathbf{A}^+ + \mathbf{P}^{(2)}\mathbf{B}^-]_{ij}}{[\mathbf{A}^- + \mathbf{P}^{(2)}\mathbf{B}^+]_{ij}}}$;
  Compute $\mathbf{W}_{ij}$ by eigen-decomposition of $\mathbf{X}(\mathbf{I} - \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T)\mathbf{X}^T$;
end while

V. Experiments

In our experiments, we will evaluate the proposed methods on several cross-domain text data sets.

A. Evaluation Metrics

To evaluate the clustering results, we adopt the performance measures used in [34]. These performance measures are the standard measures widely used for clustering.

Clustering Accuracy: Clustering Accuracy discovers the one-to-one relationship between clusters and classes and measures the extent to which each cluster contained data points from the corresponding class. Clustering Accuracy is defined as follows:

$$Acc = \frac{\sum_{i=1}^{n} \delta(map(r_i), l_i)}{n}, \tag{21}$$

where r_i denotes the cluster label of $\mathbf{x}_i$, and l_i denotes the true class label, n is the total number of documents, $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and $map(r_i)$ is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data set.

Normalized Mutual Information: The second measure is the Normalized Mutual Information (NMI), which is used for determining the quality of clusters. Given a clustering result, the NMI is estimated by

$$NMI = \frac{\sum_{i=1}^{c}\sum_{j=1}^{c} n_{i,j} \log \frac{n_{i,j}}{n_i \hat{n}_j}}{\sqrt{(\sum_{i=1}^{c} n_i \log \frac{n_i}{n})(\sum_{j=1}^{c} \hat{n}_j \log \frac{\hat{n}_j}{n})}}, \tag{22}$$

where n_i denotes the number of data contained in the cluster $\mathcal{C}_i (1 \leq i \leq c)$, $\hat{n}_j$ is the number of data belonging to the $\mathcal{L}_j (1 \leq j \leq c)$, and $n_{i,j}$ denotes the number of data that are in the intersection between the cluster $\mathcal{C}_i$ and the class $\mathcal{L}_j$. The larger the NMI is, the better the clustering result will be.

To evaluate the classification results, we use the classification accuracy.

B. Data Sets

In order to evaluate the proposed methods, we use 2 text data sets, which are widely used in cross-domain classification literature [22] [23] [25].

WebKB[1] The WebKB data set contains webpages gathered from university computer science departments (Cornell, Texas, Washington, Wisconsin). There are about 8280 documents and they are divided into 7 categories, and we choose student, faculty, course and project these four most populous entity-representation categories for clustering, named WebKB4. We consider clustering the web pages of each university as one task. Therefore, we have 4 tasks.

20Newsgroup[2] The 20 Newsgroups is a collection of approximately 20000 newsgroup documents, partitioned across 20 different newsgroups nearly evenly. We generate 2 cross-domain data sets, i.e. Rec.vs.Talk and Comp.vs.Sci, for evaluating multi-task clustering and transductive transfer classification methods. In detail, two top categories are chosen, one as positive and the other as negative. Then the data are split based on sub-categories. The task is defined as top category classification. The splitting ensures the data in different tasks are related but different, since they are drawn from the same top category but different sub-categories. The detailed constitutions of the 2 data sets are summarized in Table I.

Table I
Constitution of the 2 data sets generated from 20Newsgroup

Data set	Task id	Class 1	Class 2
Rec.vs.Talk	Task 1	rec.autos	talk.politics.guns
	Task 2	rec.sport.baseball	talk.politics.mideast
Comp.vs.Sci	Task 1	comp.os.ms-windows.misc	sci.crypt
	Task 2	comp.sys.mac.hardware	sci.space

Table.II summarizes the characteristics of the 3 data sets used in this experiment.

Table II
Description of the data sets

Data set	Task id	#Sample	#Feature	#Class
WebKB4	Task 1	227	2000	4
	Task 2	250	2000	4
	Task 3	248	2000	4
	Task 4	304	2000	4
Rec.vs.Talk	Task 1	1844	2000	2
	Task 2	1545	2000	2
Comp.vs.Sci	Task 1	1875	2000	2
	Task 2	1827	2000	2

C. Experiment 1: Multi-Task Clustering

In this experiment, we study multi-task clustering. We assume that the labels of all the tasks in each data set

[1] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/
[2] http://people.csail.mit.edu/jrennie/20Newsgroups/

are unknown. We compare the proposed multi-task clustering method with typical single-task clustering methods, e.g. Kmeans (KM), *Principal Component Analysis* (PCA)+Kmeans (PCAKM), Normalized Cut (NCut) [35] and adaptive subspace iteration (ASI) [30]. Note that Kmeans can be seen as a special case of the proposed method with $\lambda = 1$. We also present the experimental results of clustering the data of all the tasks together using Kmeans, PCA+Kmeans, NCut and ASI. Note that clustering the data of all the tasks together via ASI corresponds to the proposed method with $\lambda = 0$.

1) Methods & Parameter Settings: We set the number of clusters equal to the true number of classes for all the clustering algorithms. For NCut, the scale parameter of Gaussian kernel is set by the grid $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$. For PCAKM, the reduced dimension of PCA is set to the minimal number that preserves at least 95% of the information. For LSSMTC, we set l by searching the grid $\{2, 2^2, 2^3, 2^4\}$. And the regularization parameter λ is set by searching the grid $\{0.25, 0.5, 0.75\}$. Under each parameter setting, we repeat clustering 5 times, and the mean result as well as the standard deviation is computed. We report the mean and standard deviation result corresponding to the best parameter setting for each method to compare with each other. Since our algorithm is iterative, in our experiments, we prescribe the maximum number of iterations as $T = 20$.

2) Clustering Results: We repeat each experiment 5 times, and the average results are shown in Table III, Table IV and Table V.

"All" refers to clustering the data of all the tasks together. We can see that LSSMTC indeed improves the clustering result, and outperforms Kmeans greatly, which is its single-task degeneration. This improvement owes to exploiting the relation among the tasks by learning the shared subspace. In Task 2 of WebKB4 data set, NCut achieves better clustering result than our method. This is because NCut considers the geometric structure in the data, which is suitable for data sampled from manifold, while our method does not take this into account.

In addition, it is worthwhile noticing that although our method involves combining all the tasks together and doing dimensionality reduction, it far exceeds these simple operations. As we see, simply clustering the data of all the tasks together does not necessarily improve the clustering result, because the data distributions of different tasks are different, and combining the data together directly will violate the i.i.d. assumption in single-task clustering. Moreover, the clustering result of doing dimensionality reduction followed with clustering is also not as good as LSSMTC, because it treats learning the subspace and clustering independently, while learning the subspace and clustering could benefit from each other.

D. Experiment 2: Transductive Transfer Classification

In this experiment, we study transductive transfer classification. We do experiments on any two tasks of each data set. One task is used as source task, in which the class labels of all the data are known. The other is used as target task, where the class labels of all the data are unknown and to be predicted. We compare the proposed transductive transfer classification method with support vector machine (SVM) [36], three semi-supervised learning methods, i.e. Gaussian Field Harmonic Function (GFHF) [27], Learning with Local and Global Consistency (LLGC) [28] and transductive SVM (TSVM) [29]. We also compare it with several existing transductive transfer classification methods, Co-Clustering based Classification (CoCC) [22] and Cross-Domain Spectral Classification (CDSC) [23].

1) Methods & Parameter Settings: For SVM, TSVM, CDSC, since they are designed originally for binary classification, we address the multi-class classification via 1-vs-rest strategy. For SVM, it is trained on the source task, and tested on the target task. For TSVM, GFHF, LLGC and our method, they are trained using both labeled (source task) and unlabeled (target task) data, and are tested on the unlabeled data. SVM is implemented by LibSVM[3] [37], while TSVM is implemented by SVM$^{Light6.01}$ [4], and linear kernel is used. The implementation of GFHF is the same as in [27]. The width of the Gaussian similarity is set via the grid $\{2^{-3}\sigma_0^2, 2^{-2}\sigma_0^2, 2^{-1}\sigma_0^2, \sigma_0^2, 2\sigma_0^2, 2^2\sigma_0^2, 2^3\sigma_0^2\}$, where σ_0 is the mean distance between any two samples in the training set. And the size of neighborhood is searched by the grid $\{5, 10, 50, 80, n-1\}$. The implementation of LLGC is the same as in [28], in which the width of the Gaussian similarity and the size of neighborhood are also determined the same as that in GFHF, and the regularization parameter is set by searching the grid $\{0.1, 1, 10, 100\}$. The implementation and parameter settings of CoCC and CDSC are the same as that in their papers. For LSSTTC, we set l by searching the grid $\{100, 200, \ldots, 900, 1000\}$. And the regularization parameter λ is set by searching the grid $\{0.25, 0.5, 0.75\}$. Under each parameter setting, we repeat LSSTTC 5 times, and the mean result is computed.

2) Classification Results: The classification results are reported in Table VI, Table VII and Table VIII. It is obvious that the proposed transductive transfer classification method outperforms traditional single-task classification methods, e.g. SVM, TSVM, GFHF and LLGC, greatly on most transfer settings. This improvement is due to the prior knowledge, i.e. label information, in the related source task which is transferred to the target task by our method. It is also comparable to or even better than existing transductive transfer classification methods, e.g. CoCC and CDSC. Note that in Task 4 $\rightarrow$ Task 1 and Task 4 $\rightarrow$ Task 2 settings of

[3] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[4] http://svmlight.joachims.org/

Table III
CLUSTERING RESULTS ON WEBKB4

Method	Task 1		Task 2		Task 3		Task 4	
	Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI
KM	0.5784±0.0996	0.2760±0.0753	0.5670±0.0697	0.2552±0.0551	0.5671±0.0903	0.2814±0.0603	0.6770±0.0773	0.3552±0.0949
PCAKM	0.5938±0.1006	0.3085±0.0822	0.5616±0.0595	0.2446±0.0534	0.6105±0.0659	0.3224±0.0588	0.6882±0.0824	0.4187±0.0975
NCut	0.4907±0.0188	0.2816±0.0342	**0.6720±0.0000**	**0.3632±0.0000**	0.5282±0.0000	0.3466±0.0000	0.6079±0.0015	0.2555±0.0068
ASI	0.5119±0.0612	0.2374±0.0365	0.5512±0.0613	0.2947±0.0438	0.6097±0.0569	0.2933±0.0265	0.6418±0.0254	0.3591±0.0157
All KM	0.5476±0.0837	0.1846±0.0736	0.5944±0.0447	0.3178±0.0514	0.5980±0.0914	0.2147±0.1255	0.5898±0.1158	0.3108±0.1135
All PCAKM	0.5912±0.0841	0.2318±0.0907	0.5952±0.0565	0.2059±0.0182	0.5718±0.0789	0.1413±0.1206	0.6753±0.1060	0.3812±0.1094
All NCut	0.5683±0.0000	0.2505±0.0000	0.5920±0.0000	0.2721±0.0000	0.4960±0.0000	0.2340±0.0000	0.5132±0.0000	0.2620±0.0000
All ASI	0.5731±0.0581	0.1209±0.0515	0.5384±0.0232	0.2296±0.0122	0.5044±0.0790	0.2965±0.0879	0.6234±0.0845	0.3363±0.1297
LSSMTC	**0.6247±0.0336**	**0.3369±0.0144**	0.6304±0.0364	0.3416±0.0101	**0.6677±0.0408**	**0.3552±0.0147**	**0.7329±0.0333**	**0.4240±0.0096**

Table IV
CLUSTERING RESULTS ON REC.VS.TALK

Method	Task 1		Task 2	
	Acc	NMI	Acc	NMI
KM	0.6467±0.0382	0.1884±0.0307	0.6454±0.0967	0.1568±0.1379
PCAKM	0.6757±0.0015	0.2122±0.0020	0.6344±0.1131	0.1416±0.1602
NCut	0.6779±0.0000	0.2216±0.0000	0.6887±0.0000	0.2122±0.0000
ASI	0.6303±0.0607	0.1311±0.1012	0.6170±0.1464	0.1401±0.1893
All KM	0.6551±0.0382	0.1742±0.0160	0.5898±0.0271	0.0558±0.0443
All PCAKM	0.6765±0.0348	0.1796±0.0184	0.6061±0.0262	0.0770±0.0381
All NCut	0.6866±0.0000	0.2604±0.0000	0.6188±0.0000	0.0783±0.0000
All ASI	0.6241±0.0585	0.1133±0.0660	0.5683±0.0376	0.0295±0.0357
LSSMTC	**0.8433±0.0804**	**0.4306±0.0582**	**0.7895±0.0827**	**0.3473±0.0835**

Table V
CLUSTERING RESULTS ON COMP.VS.SCI

Method	Task 1		Task 2	
	Acc	NMI	Acc	NMI
KM	0.6130±0.0202	0.1727±0.0228	0.6716±0.0000	0.2087±0.0000
PCAKM	0.6073±0.0190	0.1661±0.0214	0.6716±0.0000	0.2087±0.0000
NCut	0.6683±0.0000	0.2327±0.0000	0.6678±0.0000	0.1694±0.0000
ASI	0.7404±0.1615	0.3444±0.2041	0.6657±0.0021	0.1282±0.0098
All KM	0.6656±0.0727	0.2330±0.0924	0.5407±0.0211	0.0532±0.0191
All PCAKM	0.6659±0.0726	0.2334±0.0922	0.5409±0.0212	0.0536±0.0193
All NCut	0.6352±0.0000	0.1941±0.0000	0.5506±0.0000	0.0627±0.0000
All ASI	0.6241±0.0585	0.1133±0.0660	0.5683±0.0376	0.0295±0.0357
LSSMTC	**0.8801±0.0076**	**0.5376±0.0155**	**0.8016±0.0614**	**0.3347±0.1407**

WebKB4 data set, "negative transfer" [17] occurred, where transfer learning lowers the learning performance.

VI. CONCLUSIONS AND FUTURE WORKS

The contribution of this paper includes the following aspects. First of all, we initiate a novel clustering paradigm, i.e. multi-task clustering, which utilizes the relation among multiple clustering tasks and outperforms traditional single-task clustering methods greatly. As far as we know, this is the *first* work addressing multi-task clustering. Secondly, we extend our multi-task clustering method to transductive transfer classification, which is comparable to or even better than existing methods.

In our future work, we will extend our method to take into account geometric structure as in [34] [38].

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (No.60721003, No.60673106 and No.60573062) and the Specialized Research Fund for the Doctoral Program of Higher Education. We would like to thank the anonymous reviewers for their helpful comments. And we especially thank one of the anonymous reviewers for pointing out a recent work [39] which also considers exploring the relation among multiple unsupervised domains.

APPENDIX

PROOF OF THEOREM III.4

Proof: We rewrite Eq.(17) as

$$\begin{aligned} L(\mathbf{P}^{(k)}) &= \text{tr}(\mathbf{P}^{(k)}\mathbf{B}^{+}\mathbf{P}^{(k)T} - \mathbf{P}^{(k)}\mathbf{B}^{-}\mathbf{P}^{(k)T} \\ &- 2\mathbf{A}^{+}\mathbf{P}^{(k)T} + 2\mathbf{A}^{-}\mathbf{P}^{(k)T}) \end{aligned}$$

By Lemma III.3, we have

$$\mathbf{tr}(\mathbf{P}^{(k)}\mathbf{B}^{+}\mathbf{P}^{(k)T}) \leq \sum_{ij} \frac{(\mathbf{P}'^{(k)}\mathbf{B}^{+})_{ij}\mathbf{P}_{ij}^{(k)2}}{\mathbf{P}_{ij}'^{(k)}}$$

Table VI
CLASSIFICATION RESULTS ON WEBKB4

Source	Target	SVM	TSVM	GFHF	LLGC	CoCC	CDSC	LSSTTC
Task 1	Task 2	0.6280	0.6320	0.5920	0.6520	0.6400	0.6760	**0.7024**
Task 1	Task 3	0.6371	0.6492	0.7137	0.7379	0.7258	0.7339	**0.7419**
Task 1	Task 4	0.7138	0.7303	0.7467	0.7993	0.7895	0.8092	**0.8125**
Task 2	Task 1	0.5639	0.6167	0.5639	0.6432	0.6520	**0.6828**	0.6745
Task 2	Task 3	0.5645	0.5960	0.5202	0.5403	0.6573	0.7218	**0.7661**
Task 2	Task 4	0.6382	0.6842	0.5132	0.7171	0.7237	0.7336	**0.7513**
Task 3	Task 1	0.6344	0.6564	0.6035	0.6960	0.6916	**0.7048**	0.7022
Task 3	Task 2	0.5920	0.6800	0.5920	0.6520	0.6760	0.6840	**0.6960**
Task 3	Task 4	0.7237	0.7304	0.5132	0.7368	0.7796	0.8257	**0.8414**
Task 4	Task 1	0.7269	0.7313	0.6608	**0.7357**	0.6740	0.7093	0.6493
Task 4	Task 2	0.6360	**0.6420**	0.5920	0.6320	0.6200	0.6360	0.6072
Task 4	Task 3	0.5887	0.5960	0.5202	0.5726	0.6976	0.7177	**0.7339**

Table VII
CLASSIFICATION RESULTS ON REC.VS.TALK

Source	Target	SVM	TSVM	GFHF	LLGC	CoCC	CDSC	LSSTTC
Task 1	Task 2	0.7605	0.8395	0.8220	0.7625	0.8544	0.8628	**0.8841**
Task 2	Task 1	0.7310	0.7988	0.7039	0.7055	0.8574	0.8829	**0.9170**

Table VIII
CLASSIFICATION RESULTS ON COMP.VS.SCI

Source	Target	SVM	TSVM	GFHF	LLGC	CoCC	CDSC	LSSTTC
Task 1	Task 2	0.6902	0.8336	0.6825	0.7170	0.9063	0.9196	**0.9489**
Task 2	Task 1	0.7803	0.8864	0.8955	0.8800	0.8960	0.9003	**0.9056**

Moreover, by the inequality $a \leq \frac{(a^2+b^2)}{2b}, \forall a, b > 0$, we have

$$\text{tr}(\mathbf{A}^{-}\mathbf{P}^{(k)T}) = \sum_{ij} \mathbf{A}^{-}_{ij}\mathbf{P}^{(k)}_{ij} \leq \sum_{ij} \mathbf{A}^{-}_{ij} \frac{\mathbf{P}^{(k)2}_{ij} + \mathbf{P}'^{(k)2}_{ij}}{2\mathbf{P}'^{(k)}_{ij}}$$

To obtain the lower bound for the remaining terms, we use the inequality that $z \geq 1 + \log z, \forall z > 0$, then

$$\begin{aligned} & \text{tr}(\mathbf{A}^{+}\mathbf{P}^{(k)T}) \geq \sum_{ij} \mathbf{A}^{+}_{ij}\mathbf{P}'^{(k)}_{ij}(1 + \log \frac{\mathbf{P}^{(k)}_{ij}}{\mathbf{P}'^{(k)}_{ij}}) \\ & \text{tr}(\mathbf{P}^{(k)}\mathbf{B}^{-}\mathbf{P}^{(k)T}) \\ \geq & \sum_{ijk} \mathbf{B}^{-}_{jk}\mathbf{P}'^{(k)}_{ij}\mathbf{P}'^{(k)}_{ik}(1 + \log \frac{\mathbf{P}^{(k)}_{ij}\mathbf{P}^{(k)}_{ik}}{\mathbf{P}'^{(k)}_{ij}\mathbf{P}'^{(k)}_{ik}}) \end{aligned}$$

By summing over all the bounds, we can get $\mathbf{Z}(\mathbf{P}^{(k)}, \mathbf{P}'^{(k)})$, which obviously satisfies (1) $\mathbf{Z}(\mathbf{P}^{(k)}, \mathbf{P}'^{(k)}) \geq J_{mt}(\mathbf{P}^{(k)})$; (2)$\mathbf{Z}(\mathbf{P}^{(k)}, \mathbf{P}^{(k)}) = J_{mt}(\mathbf{P}^{(k)})$

To find the minimum of $\mathbf{Z}(\mathbf{P}^{(k)}, \mathbf{P}'^{(k)})$, we take the Hessian matrix of $Z(\mathbf{P}^{(k)}, \mathbf{P}'^{(k)})$

$$\begin{aligned} \frac{\partial^2 Z(\mathbf{P}^{(k)}, \mathbf{P}'^{(k)})}{\partial \mathbf{P}^{(k)}_{ij} \partial \mathbf{P}^{(k)}_{kl}} &= \delta_{ik}\delta_{jl}\Big(\frac{2(\mathbf{P}'^{(k)}\mathbf{B}^{-} + \mathbf{A}^{+})_{ij}\mathbf{P}'^{(k)}_{ij}}{\mathbf{P}^{(k)2}_{ij}} \\ &+ \frac{2(\mathbf{P}'^{(k)}\mathbf{B}^{+} + \mathbf{A}^{-})_{ij}}{\mathbf{P}'^{(k)}_{ij}}\Big) \end{aligned}$$

which is a diagonal matrix with positive diagonal elements.

Thus $Z(\mathbf{P}^{(k)}, \mathbf{P}'^{(k)})$ is a convex function of $\mathbf{P}^{(k)}$. Therefore, we can obtain the global minimum of $Z(\mathbf{P}^{(k)}, \mathbf{P}'^{(k)})$ by setting $\frac{\partial Z(\mathbf{P}^{(k)}, \mathbf{P}'^{(k)})}{\partial \mathbf{P}^{(k)}_{ij}} = 0$ and solving for $\mathbf{P}^{(k)}$, from which we can get Eq.(18). ■

REFERENCES

[1] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in *ICML*, 2001, pp. 577–584.

[2] S. Basu, M. Bilenko, and R. J. Mooney, "A probabilistic framework for semi-supervised clustering," in *KDD*, 2004, pp. 59–68.

[3] B. Kulis, S. Basu, I. S. Dhillon, and R. J. Mooney, "Semi-supervised graph clustering: a kernel approach," in *ICML*, 2005, pp. 457–464.

[4] T. Li, C. Ding, and M. I. Jordan, "Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization," in *ICDM*, 2007, pp. 577–582.

[5] F. Wang, T. Li, and C. Zhang, "Semi-supervised clustering via matrix factorization," in *SDM*, 2008, pp. 1–12.

[6] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *KDD*, 2001, pp. 269–274.

[7] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *KDD*, 2003, pp. 89–98.

[8] C. H. Q. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *KDD*, 2006, pp. 126–135.

[9] Q. Gu and J. Zhou, "Co-clustering on manifolds," in *KDD*, 2009, pp. 359–368.

[10] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[11] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *KDD*, 2004, pp. 109–117.

[12] C. A. Micchelli and M. Pontil, "Kernels for multi–task learning," in *NIPS*, 2004.

[13] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *Journal of Machine Learning Research*, vol. 6, pp. 1817–1853, 2005.

[14] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *NIPS*, 2006, pp. 41–48.

[15] J. Chen, L. Tang, J. Liu, and J. Ye, "A convex formulation for learning shared structures from multiple tasks," in *ICML*, 2009, p. 18.

[16] S. J. Pan and Q. Yang, "A survey on transfer learning," Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China, Tech. Rep. HKUST-CS08-08, November 2008. [Online]. Available: http://www.cse.ust.hk/ sinnopan/publications/TLsurvey_0822.pdf

[17] W. Dai, O. Jin, G.-R. Xue, Q. Yang, and Y. Yu, "Eigentransfer: a unified framework for transfer learning," in *ICML*, 2009, p. 25.

[18] P. Wu and T. G. Dietterich, "Improving svm accuracy by training on auxiliary data sources," in *ICML*, 2004.

[19] X. Liao, Y. Xue, and L. Carin, "Logistic regression with an auxiliary data source," in *ICML*, 2005, pp. 505–512.

[20] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *ICML*, 2007, pp. 193–200.

[21] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *ICML*, 2007, pp. 759–766.

[22] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Co-clustering based classification for out-of-domain documents," in *KDD*, 2007, pp. 210–219.

[23] X. Ling, W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Spectral domain-transfer learning," in *KDD*, 2008, pp. 488–496.

[24] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *AAAI*, 2008, pp. 677–682.

[25] J. Gao, W. Fan, J. Jiang, and J. Han, "Knowledge transfer via multiple model local structure mapping," in *KDD*, 2008, pp. 283–291.

[26] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Self-taught clustering," in *ICML*, 2008, pp. 200–207.

[27] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003, pp. 912–919.

[28] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *NIPS*, 2003.

[29] T. Joachims, "Transductive inference for text classification using support vector machines," in *ICML*, 1999, pp. 200–209.

[30] T. Li, S. Ma, and M. Ogihara, "Document clustering via adaptive subspace iteration," in *SIGIR*, 2004, pp. 218–225.

[31] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge: Cambridge University Press, 2004.

[32] C. H. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, no. 1, 2008.

[33] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2000, pp. 556–562.

[34] D. Cai, X. He, X. Wu, and J. Han, "Non-negative matrix factorization on manifold," in *ICDM*, 2008, pp. 63–72.

[35] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[36] V. Vapnik, *Statistical Learning Theory*. Wiley, 1998.

[37] R.-E. Fan, P.-H. Chen, and C.-J. Lin, "Working set selection using second order information for training support vector machines," *Journal of Machine Learning Research*, vol. 6, pp. 1889–1918, 2005.

[38] Q. Gu and J. Zhou, "Local learning regularized nonnegative matrix factorization," in *IJCAI*, 2009, pp. 1046–1051.

[39] J. Gao, W. Fan, Y. Sun, and J. Han, "Heterogeneous source consensus learning via decision propagation and negotiation," in *KDD*, 2009, pp. 339–348.