NON-CONVEX TRANSFER SUBSPACE LEARNING FOR UNSUPERVISED DOMAIN ADAPTATION

Zhipeng Lin[1], *Zhenyu Zhao*[2], *Tingjin Luo*[2,*] , *Wenjing Yang*[1], *Yongjun Zhang*[3], *Yuhua Tang*[1]
[1]*State Key Laboratory of High Performance Computing,*
College of Computer, National University of Defense Technology, Changsha, China
[2]*College of Liberal Arts and Sciences, National University of Defense Technology, Changsha, China*
[3]*National Innovation Institute of Defense Technology, Beijing, China*

ABSTRACT

Transfer subspace learning aims to learn robust subspace for the target domain by leveraging knowledge from the source domain. The traditional methods often adopt the convex norm to approximate the original sparse and low-rank constraints, which make the optimization problem be easily solved. However, such relax approximation leads to the performance deviation of the original non-convex model. In this paper, we propose a novel Non-convex Transfer Subspace Learning (NTSL) method to provide a tighter approximation to the original sparse and low-rank constraints. Specifically, we design an objective function that leverages the Schatten p-norm and $\ell_{2,p}$-norm to preserve the structure between the source and target domains. With Schatten p-norm, the objective function better approximates the rank minimization problem than the nuclear norm and preserves the structure of domains. Besides, the $\ell_{2,p}$-norm can reduce the effect of noise and improve the robustness to outliers. Meanwhile, we develop an efficient algorithm to solve the non-convex minimization problem. Extensive experimental results on cross-domain tasks show the effectiveness of our proposed method.

Index Terms— Domain adaptation, Subspace learning, Non-convex optimization, Schatten p-norm, Low-rank model

1. INTRODUCTION

The lack of labeled data and the price for labeling samples drive people to use labeled and relevant data from another distinct database or domain. The urgent needs boost the development of domain adaptation. Unsupervised domain adaptation aims at training classifiers with samples from source domain and then applies classifiers to unlabeled samples from target domain [1]. In unsupervised domain adaptation, the direct application of traditional statistical learning models probably obtains unsatisfactory performance due to the distribution shift problem, which is caused by the different distribution between domains. The transfer subspace learning approaches are proposed to tackle the distribution shift problem.

Recently, transfer subspace learning shows its potential in unsupervised domain adaptation. The basic approaches for transfer subspace learning seek subspace where the distribution divergence between domains gets decreased [2, 3] or the relationship among distributions are modeled [4, 5, 6, 7]. In the literature, there are two main ideas for transfer subspace learning. The first idea is based on manifold learning, which could effectively extract the global and local structure of data in domains. For instance, Geodesic Flow Kernel (GFK) [3] finds a geodesic from the source domain to the target domain based on the kernel methods. The other idea is based on the statistical feature alignment on the low dimensional subspace, which minimizes the divergence between two distributions. In this paper, we focus on the second type of transfer subspace learning methods.

Among the second type of methods, there are many transfer subspace learning methods, such as Transfer Subspace Learning (TSL) [2], Subspace Alignment (SA) [4], Low-rank Transfer Subspace Learning (LTSL) [5], and so on. TSL and SA, which are two representative methods of them, adopt subspace learning to obtain the statistical alignment of source domain and target domain. However, due to the lack of low-rank and sparse constraints, the alignment of source domain and target domain may be not robust under noise [5, 8]. Many studies [5, 6, 7, 9, 10] find that importing the low-rank and sparse constraints into transfer subspace learning could effectively improve the performance of domain adaption. Shao *et al.* [5] gave a concise proof of the boundary of the objective function for low-rank transfer subspace learning. Furthermore, Xu *et al.* [6] used the ℓ_1-norm sparse constraint to obtain the impressive improvement. The ideal low-rank and sparse constraints can be solved by the minimization of the rank and the $\ell_{2,0}$-norm. However, they are well known NP-hard problems and difficult to solve directly [11, 12]. Therefore, to solve the optimization problem efficiently, these traditional methods adopt nuclear norm and $\ell_{2,1}$-norm to replace them, respectively. And then they obtain the approximate optimal solution by solving the relaxed convex objective func-

* Corresponding author is Tingjin Luo (E-mail: luottingjin03@nudt.edu.cn). This work was supported by the National Natural Science Foundation of China (Grant No. 11701566 and 91648204).

978-1-5386-9552-4/19/$31.00 ©2019 IEEE
DOI 10.1109/ICME.2019.00254

tions [13, 14]. Although these models have exhibited promising results, there is still room for improvement, mainly because the relaxation for these methods may result in the serious deviation from the original solution [15, 16].

In this paper, to approximate the original low-rank and sparse model better, we propose a novel model named Non-convex Transfer Subspace Learning (NTSL). NTSL preserves the structural information of different domains by using non-convex Schatten p-norm and $\ell_{2,p}$-norm. By analysis, the Schatten p-norm and $\ell_{2,p}$-norm are the closer approximation to the low rank and sparsity of a matrix than the nuclear norm and $\ell_{2,1}$-norm respectively when $p < 1$. When $p \to 0$, our relaxation is more robust and effective than the traditional low-rank transfer subspace learning methods, which is a special case of our methods for $p = 1$. Therefore, our NTSL can obtain a tighter approximation of original low-rank and sparse constraints, which help us achieve better performance. We derive an efficient algorithm based on ADMM [17] to solve our formulated non-convex problem (when $p < 1$). In addition, we give a theoretical analysis on the computational complexity. Experiments on real-world datasets present that our non-convex method improves the accuracy of visual domain adaptation comparing with the-state-of-art methods.

We highlight the contributions of this paper as follows:

- We propose a novel Non-convex Transfer Subspace Learning method (NTSL) to obtain a better reconstruction and transformation between different domains. NTSL can adaptively adjust the relaxation of low-rank and sparse constraints.
- To optimize the new objective function, we derive an efficient algorithm based on ADMM and give the theoretical analysis.
- Extensive experiments on four typical cross-domain datasets illustrate the effectiveness and superiority of our proposed method.

2. PRELIMINARIES

In this work, we denote $\boldsymbol{X}_s \in \mathbb{R}^{l \times n_s}$ and $\boldsymbol{X}_t \in \mathbb{R}^{l \times n_t}$ as the source and target data matrix respectively, where l is the dimension of each sample. While n_s and n_t are the corresponding number of samples in domains. Denote low-rank transformation matrix by $\boldsymbol{P} \in \mathbb{R}^{l \times d}$ and reconstruction matrix by $\boldsymbol{Z} \in \mathbb{R}^{n_s \times n_t}$, where d is the dimension of discriminative subspace. Let $\boldsymbol{E} \in \mathbb{R}^{d \times n_t}$ be the sparse noise matrix. Matrices are written as boldface uppercase and vectors as boldface lowercase. For matrix $\boldsymbol{M}$, the i, j-th entry, the i-th row, the i-th column of matrix $\boldsymbol{M}$ are respectively denoted by $\boldsymbol{M}_{i,j}$, $\boldsymbol{m}^i$, and $\boldsymbol{m}_i$. In addition, $\boldsymbol{M}^{-1}$, $\boldsymbol{M}^{\mathrm{T}}$ and $\mathrm{Tr}\,(\boldsymbol{M})$ represent the inverse, transpose and trace of $\boldsymbol{M}$ respectively.

The $\ell_{2,p}$-norm and Schatten p-norm $(0 < p < \infty)$ have been successfully applied in the subspace learning [15, 18], both of which are non-convex norm. The $\ell_{2,p}$-norm of a matrix $\boldsymbol{M} \in \mathbb{R}^{n \times m}$ is defined as:

$$\|\boldsymbol{M}\|_{2,p} \triangleq \left(\sum_{i=1}^{n} \left\| \boldsymbol{m}^i \right\|_2^p \right)^{\frac{1}{p}} = \left(\sum_{i=1}^{n} \left(\sum_{j=1}^{m} |M_{i,j}|^2 \right)^{\frac{p}{2}} \right)^{\frac{1}{p}}.$$

The Schatten p-norm $(0 < p < \infty)$ of a matrix $\boldsymbol{M} \in \mathbb{R}^{n \times m}$ is defined as

$$\|\boldsymbol{M}\|_{S_p} \triangleq \left(\sum_{i=1}^{\min\{n,m\}} \sigma_i^p \right)^{\frac{1}{p}} = \left(Tr\left(\left(\boldsymbol{M}^{\mathrm{T}} \boldsymbol{M} \right)^{\frac{p}{2}} \right) \right)^{\frac{1}{p}},$$

where σ_i is the i-th singular value of matrix $\boldsymbol{M}$.

Note that, when $p = 1$, the $\ell_{2,p}$-norm is $\ell_{2,1}$-norm and the Schatten p-norm is the trace norm. For $p = 0$, the Schatten 0-norm is the rank.

3. NON-CONVEX TRANSFER SUBSPACE LEARNING METHOD

3.1. Problem Formulation

Transfer subspace learning seeks to learn the projection that reconstructs common subspace where the distributions of source data and target data approaches the same independent-identical-distribution. To begin with, we give the basic assumption and define the basic problem.

The classical methods assume that samples in source domain $\boldsymbol{X}_s$ and target domain $\boldsymbol{X}_t$ lie in a union of common subspace, in which the target data can be linearly represented by source data. Such basic assumption has been adopted by many transfer subspace transfer learning methods [4, 5, 6, 7, 9, 10]. Based on the assumption, we can reconstruct the target data by the source data in the common subspace. Hence, we have the following basic problem.

Basic Problem: Given the source data matrix $\boldsymbol{X}_s$ and target data matrix $\boldsymbol{X}_t$, the optimal goal is to find a subspace $\boldsymbol{Z}$ to linearly represent test data $\boldsymbol{X}_t$ by train data $\boldsymbol{X}_s$ in a union of discriminative subspace $\boldsymbol{P}$. The optimal problem is formally given by:

$$\begin{cases} \underset{\boldsymbol{Z},\boldsymbol{P}}{\arg\min}\, F(\boldsymbol{P}, \boldsymbol{X}_s) + \lambda \mathrm{rank}(\boldsymbol{Z}) \\ \text{s.t. } \boldsymbol{P}^{\mathrm{T}} \boldsymbol{X}_t = \boldsymbol{P}^{\mathrm{T}} \boldsymbol{X}_s \boldsymbol{Z}, \end{cases} \tag{1}$$

where $F(\boldsymbol{P}, \boldsymbol{X}_s)$ is a general learning function for the discriminative subspace $\boldsymbol{P}$. The learning function could be specific according to different subspace learning methods. According to Theorem 2 in [5], the $F(\boldsymbol{P}, \boldsymbol{X}_s)$ is lower bounded by $F(\boldsymbol{P}, \boldsymbol{X}_s)$ with a small error term, which guarantees the performance of the learned subspace $\boldsymbol{P}$.

The basic model has two main limitations. First, it assumes that there is no data noise and the target data can be correctly represented by the source data in the common subspace. Obviously, this is a strong assumption in many real-world applications. Second, the traditional approach to imposing low-rank constraint is to use the trace norm as the relaxation of corresponding constraint. The convex norm results in serious deviation from the original problem.

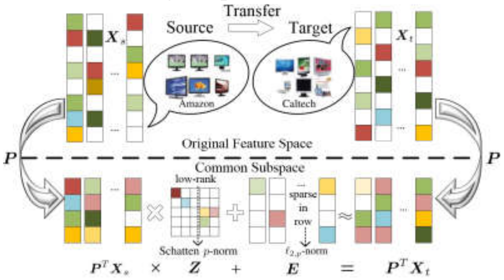


Fig. 1: Framework of the proposed method. The low-rank and sparse constraints are closely approximated by non-convex norm. Some elements in this figure are similar to [5].

Our model: We import the error term $\boldsymbol{E}$ to reduce the noise during the transfer subspace learning. Meanwhile, we use the minimization of $\|\boldsymbol{E}\|_{2,0}$ to directly encourage $\boldsymbol{E}$ to be sparse in rows. There are many norms to match the noise term. We suppose the noise is sample-specific, which means some samples in the source domain are noises or outliers and the others are enough clean for successful transfer learning. Based on the assumption, we need to ensure $\boldsymbol{E}$ is sparse in rows in order to make it more robust to sample-specific noises and outliers. In addition, we reserve the low-rank constraint on matrix $\boldsymbol{Z}$. The benefit of low-rank reconstruction (LRR) is two-fold. First, the LRR has more robustness to data noise and corruption than the direct subspace alignment. Second, the LRR can satisfy the block-wise structure of matrix $\boldsymbol{Z}$, which encourages the target samples to be reconstructed by its neighbor in the source domain. Formally, we have

$$\begin{cases} \underset{\boldsymbol{Z},\boldsymbol{P}}{\arg\min} \;\; F(\boldsymbol{P}, \boldsymbol{X}_s) + \lambda_1 \|\boldsymbol{E}\|_{2,0} + \lambda_2 \mathrm{rank}(\boldsymbol{Z}) \\ \text{s.t. } \boldsymbol{P}^{\mathrm{T}}\boldsymbol{X}_t = \boldsymbol{P}^{\mathrm{T}}\boldsymbol{X}_s\boldsymbol{Z} + \boldsymbol{E}. \end{cases} \tag{2}$$

The problem (2) is difficult as rank minimization and $\ell_{2,0}$-norm minimization are both NP-hard. The traditional transfer subspace learning methods use the convex norm minimization to relax the low-rank and sparse constraints, which results in deviation from the original solution. In this paper, we use Schatten p-Norm and $\ell_{2,p}$-norm as non-convex envelopes of the rank and $\ell_{2,0}$-norm. The value of p is selected in $(0, 1]$. When p is close to 0, the Schatten p-Norm $\|\boldsymbol{X}\|_{S_p}$ is a closer approximation to the rank of $\boldsymbol{X}$ than trace norm. Compared to $\ell_{2,1}$-norm, $\ell_{2,p}$-norm could enforce more sparsity on the row of $\boldsymbol{E}$ for $p \to 0$, hence $\ell_{2,p}$-norm is more robust to outliers. PCA is an effective and efficient subspace learning methods. We replace the learning function $F(P, X_s)$ with PCA to improve computing efficiency. The framework of our method is shown in Fig. 1. We rewrite problem (2) as follows:

$$\begin{cases} \underset{\boldsymbol{Z},\boldsymbol{P}}{\arg\min} \;\; \mathrm{Tr}\left(-\boldsymbol{P}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{P}\right) + \lambda_1 \|\boldsymbol{E}\|_{2,p}^{p} + \lambda_2 \|\boldsymbol{Z}\|_{S_p}^{p} \\ \text{s.t. } \boldsymbol{P}^{T}\boldsymbol{X}_t = \boldsymbol{P}^{T}\boldsymbol{X}_s\boldsymbol{Z} + \boldsymbol{E}, \;\; \boldsymbol{P}^{\mathrm{T}}\boldsymbol{P} = \boldsymbol{I}, \end{cases} \tag{3}$$

where $\boldsymbol{\Sigma}$ is the data covariance matrix. After solving $\boldsymbol{Z}$ and $\boldsymbol{P}$, we could respectively use the projected source and target data matrix as training samples and test samples. It is obvious that problem (3) can be transformed into problem (1) when the parameter λ_1 is relatively lager than parameter λ_2. Thus, the performance of our learned discriminative subspace $\boldsymbol{P}$ is guaranteed by Theorem 2 in [5].

3.2. Algorithm for Solving the Optimization Problem

Optimization problem in formula (3) is non-convex. We use the ADMM to iteratively update each variable by fixing other variables. We can convert formula (3) into following augmented Lagrange multiplier function $\mathcal{L}$:

$$\begin{aligned} \mathcal{L} = \; & \mathrm{Tr}\left(-\boldsymbol{P}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{P}\right) + \lambda_1 \|\boldsymbol{E}\|_{2,p}^{p} + \lambda_2 \|\boldsymbol{J}\|_{S_p}^{p} \\ & + \left\langle \boldsymbol{Y_1}, \; \boldsymbol{P}^{T}\boldsymbol{X}_t - \boldsymbol{P}^{T}\boldsymbol{X}_s\boldsymbol{Z} - \boldsymbol{E} \right\rangle \\ & + \langle \boldsymbol{Y}_2, \; \boldsymbol{Z} - \boldsymbol{J} \rangle + \left\langle \boldsymbol{Y}_3, \; \boldsymbol{P}^{\mathrm{T}}\boldsymbol{P} - \boldsymbol{I} \right\rangle \\ & + \frac{\mu}{2} \left\| \boldsymbol{P}^{T}\boldsymbol{X}_t - \boldsymbol{P}^{T}\boldsymbol{X}_s\boldsymbol{Z} - \boldsymbol{E} \right\|_F^2 + \frac{\mu}{2} \|\boldsymbol{Z} - \boldsymbol{J}\|_F^2 \\ & + \frac{\mu}{2} \left\| \boldsymbol{P}^{\mathrm{T}}\boldsymbol{P} - \boldsymbol{I} \right\|_F^2 . \end{aligned} \tag{4}$$

The main steps of solving (4) are as follows.

Step 1 *(Fixing $\boldsymbol{Z}$, $\boldsymbol{E}$ $\boldsymbol{J}$ and Optimizing $\boldsymbol{P}$)*: When fixing $\boldsymbol{Z}$, $\boldsymbol{E}$ $\boldsymbol{J}$, the problem (4) is equivalent to minimize the following formulation:

$$\begin{aligned} \mathcal{L}(\boldsymbol{P}) = \; & \mathrm{Tr}\left(-\boldsymbol{P}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{P}\right) + \left\langle \boldsymbol{Y}_1, \; \boldsymbol{P}^{T}\boldsymbol{X}_t - \boldsymbol{P}^{T}\boldsymbol{X}_s\boldsymbol{Z} - \boldsymbol{E} \right\rangle \\ & + \left\langle \boldsymbol{Y}_3, \; \boldsymbol{P}^{\mathrm{T}}\boldsymbol{P} - \boldsymbol{I} \right\rangle + \frac{\mu}{2} \left\| \boldsymbol{P}^{T}\boldsymbol{X}_t - \boldsymbol{P}^{T}\boldsymbol{X}_s\boldsymbol{Z} - \boldsymbol{E} \right\|_F^2 . \end{aligned}$$

Take the derivative of $\mathcal{L}$ w.r.t $\boldsymbol{P}$ and set to zero, we can get the optimal solution of $\boldsymbol{P}$ by solving the Sylvester equation:

$$\begin{aligned} & \left\{ \mu \left(\boldsymbol{X}_t - \boldsymbol{X}_s\boldsymbol{Z}\right) \left(\boldsymbol{X}_t^{\mathrm{T}} - \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{X}_s^{\mathrm{T}}\right) - 2\boldsymbol{\Sigma} \right\} \boldsymbol{P} \\ & \quad + 2\boldsymbol{P}\boldsymbol{Y}_3^{\mathrm{T}} = \left(\boldsymbol{X}_t - \boldsymbol{X}_s\boldsymbol{Z}\right)\left(\mu\boldsymbol{E} - \boldsymbol{Y}_1\right)^{\mathrm{T}} . \end{aligned} \tag{5}$$

After updating matrix $\boldsymbol{P}$, we calculate an orthonormal basis for the range of $\boldsymbol{P}$ and assign it to $\boldsymbol{P}$.

Step 2 *(Fixing $\boldsymbol{P}$, $\boldsymbol{E}$ $\boldsymbol{J}$ and Optimizing $\boldsymbol{Z}$)*: Take the derivative of $\mathcal{L}$ w.r.t $\boldsymbol{Z}$ and set to zero, we can obtain:

$$\begin{aligned} \boldsymbol{Z} = & \left(\boldsymbol{I} + \boldsymbol{X}_s^{\mathrm{T}}\boldsymbol{P}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{X}_s\right)^{-1} * \\ & \left(\boldsymbol{X}_s^{\mathrm{T}}\boldsymbol{P}\left(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{X}_t - \boldsymbol{E} + \frac{\boldsymbol{Y}_1}{\mu}\right) - \left(-\boldsymbol{J} + \frac{\boldsymbol{Y}_2}{\mu}\right)\right) . \end{aligned} \tag{6}$$

where the operator $*$ represent the matrix multiplication.

Step 3 *(Fixing $\boldsymbol{P}$, $\boldsymbol{Z}$ $\boldsymbol{E}$ and Optimizing $\boldsymbol{J}$)*: When fixing $\boldsymbol{P}$, $\boldsymbol{Z}$, $\boldsymbol{E}$, the problem in Eq. (3) is simplified as:

$$\underset{\boldsymbol{J}}{\arg\min} \;\; \frac{1}{2} \|\boldsymbol{J} - \boldsymbol{G}\|_F^2 + \frac{\lambda_2}{\mu} \|\boldsymbol{J}\|_{S_p}^{p} , \tag{7}$$

where $\boldsymbol{G} = \boldsymbol{Z} + \frac{\boldsymbol{Y}_2}{\mu}$. Eq. (7) is equivalent to problem (12) of [16], which gives details for the solution.

Step 4 *(Fixing $\boldsymbol{P}$, $\boldsymbol{Z}$, $\boldsymbol{J}$ and Optimizing $\boldsymbol{E}$)*: When fixing

$\boldsymbol{P}$, $\boldsymbol{Z}$, $\boldsymbol{J}$, the problem in Eq. (3) is simplified as:

$$\arg\min_{\boldsymbol{E}} \ \frac{1}{2} \|\boldsymbol{E} - \boldsymbol{H}\|_F^2 + \frac{\lambda_1}{\mu} \|\boldsymbol{E}\|_{2,p}^p , \tag{8}$$

where $\boldsymbol{H} = \boldsymbol{P}^T \boldsymbol{X}_t - \boldsymbol{P}^T \boldsymbol{X}_s \boldsymbol{Z} + \frac{\boldsymbol{Y}_1}{\mu}$. The problem (8) can be separated into multiple sub-problems for each row of $\boldsymbol{E}$:

$$\arg\min_{\boldsymbol{e}^i} \ \frac{1}{2} \left\|\boldsymbol{e}^i - \boldsymbol{h}^i\right\|_2^2 + \lambda \left\|\boldsymbol{e}^i\right\|_2^p . \tag{9}$$

According to the Cauchy-Buniakowsky-Schwarz inequality, we have following inequality:

$$\boldsymbol{e}^{i\mathrm{T}} \boldsymbol{h}^i \leq \left\|\boldsymbol{e}^i\right\|_2 \left\|\boldsymbol{h}^i\right\|_2 . \tag{10}$$

The condition for equality is $\boldsymbol{e}_i$ and $\boldsymbol{h}_i$ share the same direction, or the length of $\boldsymbol{e}_i$ or $\boldsymbol{h}_i$ equals zero. Thus, we have following inequality: $\frac{1}{2} \left\|\boldsymbol{e}^i - \boldsymbol{h}^i\right\|_2^2 \geq \frac{1}{2} \left(\left\|\boldsymbol{e}^i\right\|_2 - \left\|\boldsymbol{h}^i\right\|_2\right)^2$. Therefore, the objective function in problem (9) is minimized when $\boldsymbol{e}^{i*} = \left\|\boldsymbol{e}^i\right\|_2^* \left(\boldsymbol{h}^i / \left\|\boldsymbol{h}^i\right\|_2\right)$, where $\left\|\boldsymbol{e}^i\right\|_2^*$ is the optimal solution for the following problem:

$$\arg\min_{\|\boldsymbol{e}^i\|_2} \ \frac{1}{2} \left(\left\|\boldsymbol{e}^i\right\|_2 - \left\|\boldsymbol{h}^i\right\|_2\right)^2 + \lambda \left\|\boldsymbol{e}^i\right\|_2^p . \tag{11}$$

Eq. (11) is similar to problem (11) in [16]. The details of the algorithm can be find in [16].

Step 5 *(Update Multiplier $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$, $\boldsymbol{Y}_3$ and Parameter μ)*:Multipliers $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$, $\boldsymbol{Y}_3$ and Parameter μ are updated by using (12),

$$\begin{cases} \boldsymbol{Y}_1 = \boldsymbol{Y}_1 + \mu(\boldsymbol{P}^{\mathrm{T}} \boldsymbol{X}_t - \boldsymbol{P}^{\mathrm{T}} \boldsymbol{X}_s \boldsymbol{Z} - \boldsymbol{E}), \\ \boldsymbol{Y}_2 = \boldsymbol{Y}_2 + \mu(\boldsymbol{Z} - \boldsymbol{J}), \\ \boldsymbol{Y}_3 = \boldsymbol{Y}_3 + \mu(\boldsymbol{P}^{\mathrm{T}} \boldsymbol{P} - \boldsymbol{I}), \\ \quad \mu = \min(\phi\mu, \mu_{\max}). \end{cases} \tag{12}$$

In summary the procedure of solving problem (3) is summarized in Algorithm 1.

```
Algorithm 1: Solving Problem (3) by ADMM
Input: $X_t$, $X_s$ $\lambda_1$, $\lambda_2$, $p$.
Output: $P$.
while not converged do
    Update $P$ by solving Eq. (5);
    Calculte $Z$ by Eq. (6);
    Update $J$ by the optimal solution to problem (7);
    Update $E$ by the optimal solution to problem (8);
    Calculte Multiplier and Parameter by Eq. (12) ;
    Check the convergence conditions:
      $\left\|P^{\mathrm{T}} X_t - P^{\mathrm{T}} X_s Z - E\right\|_{\mathrm{inf}} < \epsilon$,
      $\|Z - J\|_{\mathrm{inf}} < \epsilon$, $\|P_{\mathrm{new}} - P_{\mathrm{old}}\|_{\mathrm{inf}} < \epsilon$;
end
return $E_t$.
```

3.3. Computational Complexity

The time-consuming components of Algorithm 1 are the solution of Sylvester equation, orthonormal basis, matrix multiplication, matrix inverse, and SVD computation. Suppose the $\boldsymbol{X_s}$ and $\boldsymbol{X_t}$ are $m \times n$ matrices. The number of iterations and deduced dimension are respectively denoted by N and d. There are k multiplications. The computational complexity of classical solution for Sylvester equation is $O(m^3)$. The computational complexity of calculating orthonormal basis is $O(md^2)$. The general complexity of matrix multiplication is $O(km^3)$ and the complexity of inverse of a $n \times n$ matrix is $O(n^3)$. The complexity of SVD computation of $\boldsymbol{G}$ is $O(n^3)$. The total the computation complexity of Algorithm 1 is $N((k+1)O(m^3) + 2O(n^3) + O(md^2))$.

4. EXPERIMENTS

4.1. Datasets Description

Office + Caltech Data Set: The Office + Caltech dataset is a popular dataset for transfer learning with 10 sharing categories, which contains four domains: Amazon (A), Webcam (W), DSLR (D), and Caltech (C). The 4,096 DeCAF6 features for Office + Caltech dataset are used [19]. We obtain 12 pairs of dataset like C→A, A→D, D→W, and so on.

Yale B + CMU PIE Data Set: The Yale B and CMU PIE are popular datasets in the field of face recognition. We use the 30×30 Yale B + CMU PIE Data Set released in [5]. In this experiment, we have two cross-domain datasets: P→Y and Y→P.

USPS + MNIST Data Set: USPS (U) and MNIST (M) are popular digits recognition datasets. We adopt the public 16×16 USPS + MNIST dataset released by Long *et al.* [20]. We obtain two domain adaptation tasks: U→M and M→U.

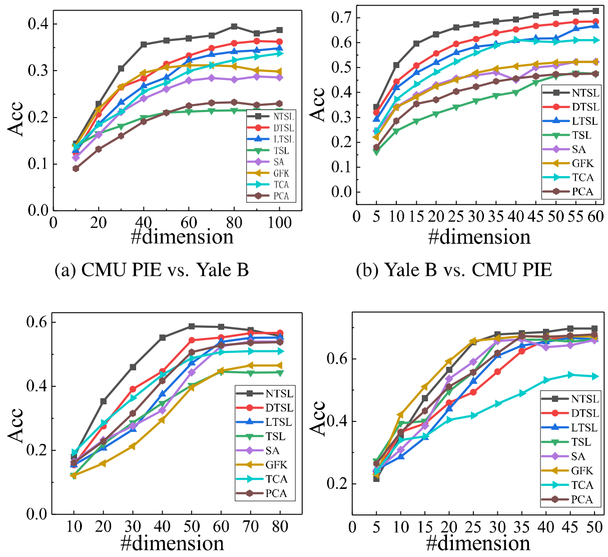
COIL 20 Data Set: The COIL 20 dataset contains 20 classes with 1440 object recognition images in 32×32. We also adopt the public COIL 20 dataset released by Long *et al.* [20], which consists of COIL1 and COIL2 taken in different directions. We have two tasks: C1→C2 and C2→C1.

4.2. Comparison Methods

To show the effectiveness of NTSL, we choose seven related comparison methods, which is from three categories: distribution alignment, transfer subspace learning without low-rank and sparse constraints and transfer subspace learning with constraints, including Principal Component Analysis (PCA), Transfer Component Analysis (TCA) [21], Geodesic Flow Kernel (GFK) [3], Subspace Alignment (SA) [4], Transfer Subspace Learning (TSL) [2], Low-rank Transfer subspace Learning (LTSL) [5], and Discriminative Transfer Subspace Learning via Low-Rank and Sparse Representation [6], which is marked as DTSL. TCA learns the transfer components across domains through minimization of Maximum Mean Discrepancy (MMD) in the Reproducing Kernel Hilbert Space (RKHS) by using an explicit low-rank representation. SA directly obtains a mapping function which aligns the source subspace with the target one without the low-rank and sparse constraints. The main idea of GFK, TSL and LTSL are summarized in Section 1. Similar to LTSL, DTSL also imposes joint low-rank and sparse constraints on the reconstruction coefficient matrix by using the nuclear norm and ℓ_1-norm.

4.3. Implementation Details

For PCA, SA, TCA, TSL, LTSL, DTSL and NTSL, we use the transformation matrix to project source and target data matrix, after which, we use the projected source data matrix as training samples and target data matrix as test samples. We applied 1NN as the classifier. For GFK, 1NN is applied after we obtain geodesic flow kernel. The general subspace learning methods for TSL and LTSL are PCA. The parameters of comparison methods were initialized and tuned according to the corresponding papers to obtain optimal performance. For the transfer tasks not tested, we use the grid-search strategy to obtain the optimal parameter. Specially, we need to tune three parameters in NTSL: λ_1, λ_2, and p. The parameter p is searched in $[0.1, 0.2, \cdots, 1.4, 1.5]$. The search ranges for λ_1 and λ_2 are $[2, 4, \cdots, 18, 20]$ and $[0.01, 0.05, 0.1, 0.5, 1, 5, 10]$ respectively. Similarly, we also search the optimal dimension for different methods on different datasets.



(a) CMU PIE vs. Yale B

(b) Yale B vs. CMU PIE

(c) USPS vs. MNIST

(d) MNIST vs. USPS

Fig. 2: Classification Accuracy (Acc) results on different dimensions on four transfer tasks: P→Y, Y→P, U→M, and M→U. Note that we also choose the optimal dimension for different methods on other tasks.

4.4. Experimental Results

To choose the optimal dimension for different methods, we conduct experiments on different dimensions and use optimal dimensions for different methods in classification tasks. We plot the accuracy curves under different dimension for four tasks: P→Y, Y→P, U→M, and M→U in Fig. 2. The accuracy of all methods goes up as the dimension increasing and finally flattens out. Generally speaking, NTSL outperforms the most of comparison methods under different dimension.

Table 1: Classification accuracy (%) for all methods on the Office + Caltech, Yale B + CMU PIE, USPS + MNIST, and COIL 20 DataSet.

Methods	PCA	TCA	GFK	SA	TSL	LTSL	DTSL	NTSL
C→A	86.0	89.9	89.4	87.0	90.2	89.5	**91.4**	89.8
C→W	74.3	78.6	75.6	71.6	78.3	79	78.6	**83.3**
C→D	81.7	82.8	87.3	80.5	84.7	86	**89.8**	86.6
A→C	78.9	79.7	79.7	78.4	83.5	82.2	**85.1**	81.7
A→W	73.1	**76.3**	68.1	69.3	71.5	71.9	75.6	76.2
A→D	72.4	**87.3**	77.1	75.9	80.3	79.6	83.4	82.8
W→C	72.3	77.4	73.6	74.7	74.5	74.5	72.5	**74.7**
W→A	75.0	**83.8**	77.5	77.1	78.8	78.9	72.1	82.6
W→D	**100.0**	**100.0**	**100.0**	99.6	**100.0**	**100.0**	**100.0**	**100.0**
D→C	73.7	79.3	77.6	76.1	79.0	**79.3**	74.5	78.7
D→A	80.3	**89.1**	83.7	82.6	85.5	86.3	82.3	88.2
D→W	95.3	98.6	99.3	99.3	**99.7**	99.3	99.3	**99.7**
P→Y	22.9	33.7	31.2	28.7	21.5	34.8	36.2	**38.7**
Y→P	47.5	61.7	52.3	52.3	46.9	66.0	68.6	**72.8**
U→M	53.8	51.0	46.5	54.0	44.6	55.2	56.6	**58.8**
M→U	66.2	54.9	67.2	66.2	66.1	66.7	67.3	**69.7**
C1→C2	88.1	88.6	84.4	84.6	84.6	85.7	**89.9**	87.9
C2→C1	88.0	86	84.2	82.9	83.8	82.2	**89.3**	88.5
Average	73.9	77.7	75.3	74.5	75.2	77.7	78.5	**80.1**

Based on the optimal dimension, we conduct experiments ten times at random for each cross-domain task and method. The average classification accuracy on different tasks for all methods is recorded in Table 1.

From Table 1, we can make three key observations as follows. To begin with, we can find that the traditional subspace learning methods like PCA do not obtain comparable performance to that of transfer subspace learning methods. The performance gap between PCA and transfer subspace learning methods validates the hypothesis that reducing divergence between domains through subspace learning can improve the accuracy of transfer learning.

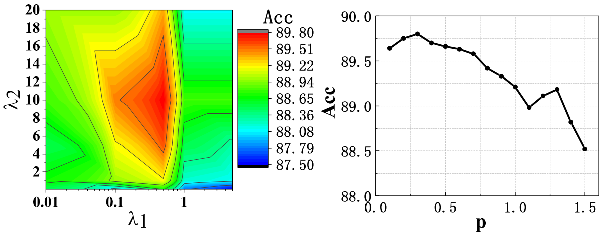
Secondly, although the employment of subspace learning to reduce divergence can improve the performance of domain adaptation, transfer subspace learning methods with low-rank and sparse constraints like LTSL, DTSL, and NTSL can better reduce divergence between domains when compared with the method without constraints like TSL and direct subspace alignment method like SA. In this experiments, TSL captures divergence without the reconstruction from the source domain to the target domain. SA directly learns a linear transformation to realize the subspace alignment. Thus, the learned transformed matrix from TSL and SA may be not so efficient as the expectation. Therefore, by low-rank reconstruction and importing sparse constraint for subspace learning, LTSL, DTSL, and NTSL obtain additional improvements.

Thirdly, as shown in Table 1, our method NTSL outperforms all the transfer subspace learning methods (TSL, LTSL, DTSL). Although low-rank reconstruction and importing sparse constraint are useful, the performance can vary depending on how the constraints are utilized and relaxed. In our model, we propose the joint Schatten p-norm and $\ell_{2,p}$-norm minimization, which allows constraints to be approximated by a closer problem, which can lead to a better performance according to the results. The leverage of Schatten p-norm and $\ell_{2,p}$-norm in objective function can reduce the deviation from the real solution and handle the outliers pursuit.

Overall, the performance of distribution alignment method (TCA), Grassmann manifold-based method (GFK), and convex norm-based transfer subspace learning methods (LTSL, DTSL) were generally worse than NTSL. Compared to the best comparison method (DTSL), our approach gains 2.0% improvement on average based on a wide range of image data sets. The NTSL outperforms all other transfer subspace learning methods in 8 tasks out of 18 tasks and is the only one with an average accuracy of more than 80%.

4.5. Parameter Sensibility and Convergence Analysis

There are three parameters in our objective function: p, λ_1, and λ_2. To demonstrate the effects of these parameters, we first plot the contour figure for parameter λ_1 and λ_2 and then test the effect of parameter p on task C→A. The results are shown in Fig. 3. We can find that our method is robust to different parameter settings within a feasible range. Specially, the curve of parameter p in Fig. 3(b) validates that a smaller p generally obtains a closer relaxation and better prediction.



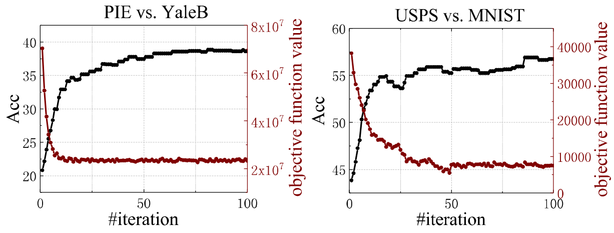
(a) λ_1 and λ_2 (b) p

Fig. 3: Parameter sensitivity analysis on Caltech vs. Amazon.

We run our method for 100 iterations on two tasks and draw the convergence curve of the objective function values and the classification accuracy in Fig. 4. Basically, the value of objective function decreases as the number of iterations going up. The accuracy of recognition varies rapidly and converges within the 50 iteration, which shows that NTSL has a good convergence property.

5. CONCLUSIONS

To preserve the structural information between domains better, we propose a novel unified Non-convex Transfer Subspace Learning (NTSL) model by integrating Schatten p-norm and $\ell_{2,p}$-norm in this paper. Meanwhile, to solve our non-convex minimization problem, an efficient algorithm is developed. Finally, experimental results on popular domain adaptation datasets demonstrate the effectiveness of our method. In the future, we plan to extend our NTSL to handle the cross-class distribution divergence on the subspace.



(a) P→Y (b) U→M

Fig. 4: Convergence sensitivity analysis for NTSL on two transfer tasks: P→Y and U→M.

References

[1] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[2] Si Si, Dacheng Tao, and Bo Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 929, 2010.

[3] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *CVPR*. IEEE, 2012, pp. 2066–2073.

[4] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2960–2967.

[5] Ming Shao, Dmitry Kit, and Yun Fu, "Generalized transfer subspace learning through low-rank constraint," *International Journal of Computer Vision*, vol. 109, no. 1-2, pp. 74–93, 2014.

[6] Yong Xu, Xiaozhao Fang, Jian Wu, Xuelong Li, and David Zhang, "Discriminative transfer subspace learning via low-rank and sparse representation," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 850–863, 2016.

[7] Zhengming Ding, Ming Shao, and Yun Fu, "Latent low-rank transfer subspace learning for missing modality recognition.," in *AAAI*, 2014, pp. 1192–1198.

[8] Zhengming Ding and Yun Fu, "Low-rank common subspace for multi-view learning," in *ICDM*. IEEE, 2014, pp. 110–119.

[9] I-Hong Jhuo, Dong Liu, DT Lee, and Shih-Fu Chang, "Robust visual domain adaptation with low-rank reconstruction," in *CVPR*. IEEE, 2012, pp. 2168–2175.

[10] Parvin Razzaghi, Parisa Razzaghi, and Karim Abbasi, "Transfer subspace learning via low-rank and discriminative reconstruction matrix(in press)," Accessed online:https://doi.org/10.1016/j.knosys.2018.08.026, 2018.

[11] Chen Xu, Zhouchen Lin, and Hongbin Zha, "A unified convex surrogate for the Schatten p-norm norm.," in *AAAI*, 2017, pp. 926–932.

[12] Feiping Nie, Heng Huang, Xiao Cai, and Chris H Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Advances in neural information processing systems*, 2010, pp. 1813–1821.

[13] Jun Liu, Shuiwang Ji, and Jieping Ye, "Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 2009, pp. 339–348.

[14] Bamdev Mishra, Gilles Meyer, Francis Bach, and Rodolphe Sepulchre, "Low-rank optimization with trace norm penalty," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2124–2149, 2013.

[15] Zhao Zhang, Mingbo Zhao, Fanzhang Li, Li Zhang, and Shuicheng Yan, "Robust alternating low-rank representation by joint ℓ_p-and $\ell_{2,p}$-norm minimization," *Neural Networks*, vol. 96, pp. 55–70, 2017.

[16] Feiping Nie, Hua Wang, Xiao Cai, Heng Huang, and Chris Ding, "Robust matrix completion via joint Schatten p-norm and ℓ_p-norm minimization," in *ICDM*. IEEE, 2012, pp. 566–574.

[17] Zhouchen Lin, Risheng Liu, and Zhixun Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in neural information processing systems*, 2011, pp. 612–620.

[18] Hengmin Zhang, Jian Yang, Fanhua Shang, Chen Gong, and Zhenyu Zhang, "LRR for subspace segmentation via tractable Schatten-p norm minimization and factorization," *IEEE Transactions on Cybernetics*, 2018.

[19] Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *MM*. ACM, 2018, pp. 402–410.

[20] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and S Yu Philip, "Transfer feature learning with joint distribution adaptation," in *ICCV*. IEEE, 2013, pp. 2200–2207.

[21] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.