

HETEROGENEOUS TENSOR DOMAIN ADAPTATION

Yicong He^{*} George K. Atia^{*†}

^{*}Department of Electrical and Computer Engineering

[†]Department of Computer Science

University of Central Florida, Orlando, FL 32816 USA

yicong.he@ucf.edu, george.atia@ucf.edu

ABSTRACT

Heterogeneous domain adaptation (HDA) addresses the challenge of domain adaptation in scenarios where the source and target domains exhibit distinct types of features. While numerous studies have considered the HDA problem and demonstrated efficacy in transferring knowledge across diverse feature types, existing methods are predominantly tailored for 1-D feature vectors, leaving the handling of high-order features largely under-explored. In this paper, we address a new problem, namely, heterogeneous tensor domain adaptation (HTDA), where features from either the source or target domains are tensors with different sizes or orders. Our approach involves a multilinear principal component analysis (MPCA) based tensor projection method which projects heterogeneous features onto a shared latent tensor space. Additionally, we leverage a probabilistic class-wise distribution alignment to align feature distributions in the tensor space, followed by label propagation from both the source data and labeled target data to unlabeled target data through graph regularization. Our experimental results on multiple domain adaptation datasets demonstrate the superior performance of our proposed tensor-based method in comparison to existing vector-based HDA methods.

Index Terms— Domain adaptation, heterogeneous domain adaptation, tensor decomposition

1. INTRODUCTION

The increase in data volume poses a significant challenge when training classifiers on new datasets, primarily due to the extensive labeling requirements. However, directly applying an existing classifier trained on one dataset (source) to classify a different dataset (target) is generally ineffective in achieving the desired classification performance. This is attributed to ‘domain shift’, wherein the source and target data exhibit different distributions [1]. This spurred considerable interest in domain adaptation (DA) techniques, which aim to mitigate the

domain discrepancy and enhance the accurate classification of target domain data by leveraging knowledge transfer from a source domain. To date, numerous methodologies for DA have been proposed [2, 3, 4, 5, 6].

In addition to the domain shift problem, several challenging issues need to be addressed in DA. One such challenge arises when the source and target domains exhibit disparities in feature types and dimensions, leading to a more intricate problem known as heterogeneous domain adaptation (HDA) [7]. This is distinct from homogeneous DA, where features share identical dimensions. Numerous HDA algorithms developed in recent years have demonstrated their effectiveness in transferring knowledge across features with different dimensions [8, 9, 10, 11, 12, 6].

However, a less explored challenge is the tensor domain adaptation (TDA) problem, where features are represented as high-order tensors. Most existing DA methods are specifically designed for 1-D feature vectors. Nevertheless, several studies have highlighted the necessity of considering the intrinsic structure of tensor features rather than solely relying on vectorized features [13, 14, 15, 16].

In this paper, we focus on a novel problem called heterogeneous tensor domain adaptation (HTDA), where tensor features in both the source and target domains may vary in size and order. Our approach involves three main components, namely, MPCA-based tensor projection, tensor class-wise distribution alignment, and tensor correlation-based label propagation. Further, we develop a novel algorithm for solving the HTDA problem by leveraging the tensor structure, and conduct an extensive evaluation of our proposed method on multiple domain adaptation datasets. To the best of our knowledge, this work represents the first attempt to tackle the challenge of heterogeneous tensor domain adaptation.

2. PRELIMINARIES AND BACKGROUND

In this paper, uppercase script letters, e.g., $\mathcal{X}$, are used to denote tensors. Boldface letters, e.g., $\mathbf{X}$, to represent matrices. An N -th order tensor is defined as $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$, where I_i denotes the size of the i -th way of the tensor. The Frobenius

This work was supported by NSF under Award CCF-2106339 and DARPA under Agreement No. HR0011-24-9-0427.

norm of tensor $\mathcal{X}$ is defined as $\|\mathcal{X}\|_F = \sqrt{\sum_{i_1 \dots i_N} |\mathcal{X}_{i_1 \dots i_N}|^2}$, where $\mathcal{X}_{i_1 \dots i_N}$ denotes the $(i_1, i_2, \dots, i_N)$ -th entry of $\mathcal{X}$.

2.1. HTDA: Problem statement

In the HTDA problem, we assume that the source data consists of N_S K -dimensional tensors $\mathcal{X}_{s,i} \in \mathbb{R}^{m_1 \times \dots \times m_K}$, $i = 1, \dots, N_S$, and the data in the target domain consists of N_T L -dimensional tensors $\mathcal{X}_{t,j} \in \mathbb{R}^{n_1 \times \dots \times n_L}$, $j = 1, \dots, N_T$. The parameters $\{m_i\}_{i=1}^K$ and $\{n_j\}_{j=1}^L$ are the dimensions of the i -th order of the source data and the j -th order of the target data, respectively. In heterogeneous cases, K , L and the tensor sizes may not be equal. Further, we define the source input tensor $\mathcal{X}_S \in \mathbb{R}^{m_1 \times \dots \times m_K \times N_S}$ and target input tensor $\mathcal{X}_T \in \mathbb{R}^{n_1 \times \dots \times n_L \times N_T}$ by stacking $\mathcal{X}_{s,i}$ and $\mathcal{X}_{t,j}$, respectively.

We follow the common HDA setting and assume that a limited set of labeled data in the target domain is available. The source and target domains share the same set of class labels, and the number of classes is C . Let $\mathbf{Y}_S \in \{0, 1\}^{N_S \times C}$ represent the source label matrix using one-hot encoding. In this matrix, $(\mathbf{Y}_S)_{ic} = 1$ if the i -th source data point belongs to the c -th class. For the target domain, the label matrix for the labeled target data is denoted as $\mathbf{Y}_L \in \{0, 1\}^{N_L \times C}$, where N_L is the number of labeled target data. The objective of the HTDA problem is to accurately classify the unlabeled target data based on the information from $\mathcal{X}_S$, $\mathcal{X}_T$, $\mathbf{Y}_S$ and $\mathbf{Y}_L$.

2.2. Tucker decomposition

Tucker decomposition [17] decomposes an input tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ into a small core tensor $\mathcal{G} \in \mathbb{R}^{J_1 \times \dots \times J_N}$ and a set of orthonormal factor matrices $\{\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times J_n}\}_{n=1}^N$, such that the original tensor can be approximated as

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \dots \times_N \mathbf{A}^{(N)} = \llbracket \mathcal{G}; \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)} \rrbracket \quad (1)$$

where the operator $\times_n$, $n = 1, \dots, N$, is defined as $(\mathcal{X} \times_n \mathbf{A})_{(n)} = \mathbf{A} \mathbf{X}_{(n)}$. The matrix $\mathbf{X}_{(n)}$ of size $I_n \times \prod_{j \neq n} I_j$ denotes the mode- n unfolding matrix defined by

$$(X_{(n)})_{i_n \overline{i_1 \dots i_{n-1} i_{n+1} \dots i_N}} = \mathcal{X}_{i_1 i_2 \dots i_N},$$

with $\overline{i_1 i_2, \dots, i_N} = i_1 + (i_2 - 1)I_1 + \dots + (i_N - 1)I_1 I_2 \dots I_N$. To simply the notation, we use $\mathcal{A} := \{\mathbf{A}^{(n)}\}_{n=1}^N$ to represent the set of orthonormal factor matrices, thus (1) can be simplified to $\mathcal{X} = \llbracket \mathcal{G}; \mathcal{A} \rrbracket$.

3. PROPOSED APPROACH AND ALGORITHM

In this section, we detail each component of the heterogeneous tensor domain adaptation (HTDA) approach. We first present the solution for the scenario where tensors from both the source and target domains share the same order D , i.e., $K = L = D$, albeit having different dimensions within these orders. The handling of tensors with distinct orders will be addressed in Section 3.5.

3.1. MPCA-based tensor projection

Our approach in this work leverages Tucker decomposition to project both the source and target tensor data into a common latent tensor space. Specifically, for a source tensor $\mathcal{X}_s \in \mathbb{R}^{m_1 \times \dots \times m_D}$ and a set of orthonormal factor matrices $\mathcal{P} := \{\mathbf{P}^{(k)}\}_{k=1}^D$, where $\mathbf{P}^{(k)} \in \mathbb{R}^{p_k \times m_k}$ and p_k represents the k -th dimension of the latent tensor space, we perform a projection $\llbracket \mathcal{X}_s; \mathcal{P} \rrbracket$. This projection transforms $\mathcal{X}_s$ from its original source feature space in $\mathbb{R}^{m_1 \times \dots \times m_D}$ to a latent tensor space in $\mathbb{R}^{p_1 \times \dots \times p_D}$. A similar projection can be performed on the target domain data $\mathcal{X}_t \in \mathbb{R}^{n_1 \times \dots \times n_D}$ with a set of orthonormal factor matrices $\mathcal{Q} := \{\mathbf{Q}^{(k)}\}_{k=1}^D$, where $\mathbf{Q}^{(k)} \in \mathbb{R}^{p_k \times n_k}$. This results in the projection of the source and target tensors to a common latent tensor space in $\mathbb{R}^{p_1 \times \dots \times p_D}$.

Such a tensor projection, however, falls short of capturing sufficient information from the original tensors and imposes limited constraints on the factor matrices $\mathcal{P}$ and $\mathcal{Q}$. Inspired by [18, 12], we introduce an additional constraint through MPCA. Specifically, the projection $\llbracket \mathcal{X}_s; \mathcal{P} \rrbracket$ can be further projected back to the source domain tensor space using $\llbracket \llbracket \mathcal{X}_s; \mathcal{P} \rrbracket; \mathcal{P}^\top \rrbracket$, where $\mathcal{P}^\top := \{(\mathbf{P}^{(k)})^\top\}_{k=1}^D$. Therefore, we can gauge the reconstruction error in terms of $\mathcal{X}_s$ as $\|\mathcal{X}_s - \llbracket \llbracket \mathcal{X}_s; \mathcal{P} \rrbracket; \mathcal{P}^\top \rrbracket\|_F^2$. By measuring the reconstruction error on both the source and target domains, we formulate the optimization problem of $\mathcal{P}$ and $\mathcal{Q}$ based on $\mathcal{X}_S$ and $\mathcal{X}_T$ as

$$\begin{aligned} \min_{\mathcal{P}, \mathcal{Q}} \quad & \lambda_1 \|\mathcal{X}_S - \llbracket \llbracket \mathcal{X}_S; \mathcal{P} \rrbracket; \mathcal{P}^\top \rrbracket\|_F^2 + \lambda_2 \|\mathcal{X}_T - \llbracket \llbracket \mathcal{X}_T; \mathcal{Q} \rrbracket; \mathcal{Q}^\top \rrbracket\|_F^2 \\ \text{subject to} \quad & \mathbf{P}^{(k)} \mathbf{P}^{(k)\top} = \mathbf{I}, \mathbf{Q}^{(k)} \mathbf{Q}^{(k)\top} = \mathbf{I}, k = 1, \dots, D \end{aligned} \quad (2)$$

where λ_1 and λ_2 are free parameters. We should remark that the above MPCA-based tensor projection method is different from the existing works [13, 14] where the latent tensor space is constructed by a set of common factor matrices.

3.2. MMD and class-wise distribution alignment

After projecting heterogeneous tensor features to a common tensor space, we are ready to perform class-wise distribution alignment. Rather than applying the widely used maximum mean discrepancy (MMD) [19, 20], we leverage the insights from the probabilistic class-wise distribution alignment approach proposed by [21] to enhance the distribution alignment process within our HTDA approach. In particular, the alignment is performed by optimizing the following problem:

$$\min_{\mathcal{P}, \mathcal{Q}} \sum_{c=1}^C \left\| \frac{1}{N_S^c} \sum_{i=1}^{N_S^c} \llbracket \mathcal{X}_{s,i}^c; \mathcal{P} \rrbracket - \frac{1}{N_T^c} \sum_{j=1}^{N_T^c} (F_T)_{jc} \llbracket \mathcal{X}_{t,j}; \mathcal{Q} \rrbracket \right\|_F^2 \quad (3)$$

$\mathbf{F}_T \in [0, 1]^{N_T \times C}$ is a soft label matrix for the target domain where each element $(F_T)_{jc}$, $j = 1, \dots, N_T$, $c = 1, \dots, C$, represents the probability of the j -th target data point belonging to the c -th class. $\mathcal{X}_{s,i}^c$, $i = 1, \dots, N_S^c$, denotes a source

tensor belonging to the c -th class. N_S^c and N_T^c are the numbers of source and target data in class c , respectively, where N_T^c is estimated as $N_T^c = \sum_{j=1}^{N_T} (F_T)_{jc}$.

3.3. Tensor correlation-based label propagation

We augment our approach by introducing a graph regularizer to propagate the labels from the source data and labeled target data to unlabeled target data. Specifically, we utilize the projected tensors in the latent tensor space of both the source and target domains to construct a neighborhood similarity graph, denoted as $\mathbf{W} \in \mathbb{R}^{(N_S+N_T) \times (N_S+N_T)}$, incorporating the tensor correlation and k -nearest neighbors (KNN). Tensor correlation is employed to measure the similarity between two projected tensors. For two D -th order tensors, $\mathcal{X}_1, \mathcal{X}_2 \in \mathbb{R}^{p_1 \times \dots \times p_D}$, the tensor correlation is defined by [22]

$$\text{Corr}(\mathcal{X}_1, \mathcal{X}_2) = \frac{\langle \mathcal{X}_1, \mathcal{X}_2 \rangle}{\sqrt{\langle \mathcal{X}_1, \mathcal{X}_1 \rangle} \sqrt{\langle \mathcal{X}_2, \mathcal{X}_2 \rangle}},$$

where the inner product $\langle \mathcal{X}_1, \mathcal{X}_2 \rangle = \sum (\mathcal{X}_1)_{i_1 i_2 \dots i_D} (\mathcal{X}_2)_{i_1 i_2 \dots i_D}$. Note that $\text{Corr}(\mathcal{X}_1, \mathcal{X}_2) \in [-1, 1]$ and the closer the two sub-tensors are, the larger the value is. Hence, the similarity W_{12} between $\mathcal{X}_1$ and $\mathcal{X}_2$ can be defined as $W_{12} = \text{Corr}(\mathcal{X}_1, \mathcal{X}_2)$.

After obtaining the similarity matrix $\mathbf{W}$, we can propagate the source and labeled target data to unlabeled target data by

$$\min_{\mathbf{F}_T} \sum_{i=1}^{N_S+N_T} \sum_{j=1}^{N_S+N_T} \mathbf{W}_{ij} \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 \quad (4)$$

where $\mathbf{f}_i$ is the i -th row of $\mathbf{F} = [\mathbf{Y}_S^\top, \mathbf{F}_T^\top]^\top \in \mathbb{R}^{(N_S+N_T) \times C}$.

3.4. Algorithm and optimization

By incorporating the tensor projection, class-wise distribution alignment, and label propagation, we formulate the final optimization problem as follows:

$$\begin{aligned} \min_{\mathcal{P}, \mathcal{Q}, \mathbf{F}_T} & \lambda_1 \|\mathcal{X}_S - \llbracket \mathcal{X}_S; \mathcal{P} \rrbracket\|_F^2 + \lambda_2 \|\mathcal{X}_T - \llbracket \mathcal{X}_T; \mathcal{Q} \rrbracket\|_F^2 \\ & + \sum_{c=1}^C \left\| \frac{1}{N_S^c} \sum_{i=1}^{N_S^c} \llbracket \mathcal{X}_{s,i}^c; \mathcal{P} \rrbracket - \frac{1}{N_T^c} \sum_{j=1}^{N_T^c} (F_T)_{jc} \llbracket \mathcal{X}_{t,j}; \mathcal{Q} \rrbracket \right\|_F^2 \\ & + \gamma \sum_{i=1}^{N_S+N_T} \sum_{j=1}^{N_S+N_T} \mathbf{W}_{ij} \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 \\ \text{s.t., } & \mathbf{P}^{(k)} \mathbf{P}^{(k)\top} = \mathbf{I}, \mathbf{Q}^{(k)} \mathbf{Q}^{(k)\top} = \mathbf{I}, k = 1, \dots, D \\ & \mathbf{F}_L = \mathbf{Y}_L, \mathbf{F}_T \geq 0, \mathbf{F}_T \mathbf{1}_C = \mathbf{1}_{N_T} \end{aligned} \quad (5)$$

where $\mathbf{F}_L$ is the soft label matrix corresponding to labeled target data, γ is a free parameter. Next, we develop an alternating minimization method to solve the problem defined in (5).

1) *Optimization of $\mathbf{F}_T$* : By defining $\mathcal{Z}_S = \llbracket \mathcal{X}_S; \mathcal{P} \rrbracket$ and $\mathcal{Z}_T = \llbracket \mathcal{X}_T; \mathcal{Q} \rrbracket$, we can unfold the projected tensors along the

$(D+1)$ -th dimension, leading to the following subproblem:

$$\begin{aligned} \mathbf{F}_T &= \arg \min_{\mathbf{F}_T} \|\mathbf{Z}_{S(D+1)} \mathbf{F}_S \mathbf{D}_S - \mathbf{Z}_{T(D+1)} \mathbf{F}_T \mathbf{D}_T\|_F^2 \\ &+ \gamma \text{tr}(\mathbf{F}_T^\top \mathbf{L}_{TT} \mathbf{F}_T + 2 \mathbf{Y}_S^\top \mathbf{L}_{ST} \mathbf{F}_T) \\ \text{s.t., } & \mathbf{F}_L = \mathbf{Y}_L, \mathbf{F}_T \geq 0, \mathbf{F}_T \mathbf{1}_C = \mathbf{1}_{N_T} \end{aligned} \quad (6)$$

where $\mathbf{D}_S$ and $\mathbf{D}_T$ are $C \times C$ diagonal matrices with c -th diagonal entries N_S^c and N_T^c , respectively. The Laplacian matrix $\mathbf{L} = [\mathbf{L}_{SS}, \mathbf{L}_{ST}; \mathbf{L}_{TS}, \mathbf{L}_{TT}]$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix with diagonal entries as the column sums of $\mathbf{W}$. Therefore, $\mathbf{F}_T$ can be obtained as [23, 21]

$$\mathbf{F}_T = \text{Proj}_{\mathcal{C}} \left(\mathbf{F}_T \odot \sqrt{\frac{[\mathbf{G}_T]^- + [\mathbf{G}_S]^+ + \mathbf{H}_D}{[\mathbf{G}_T]^+ + [\mathbf{G}_S]^- + \mathbf{H}_W}} \right) \quad (7)$$

where operation $[\cdot]^+$ sets all negative entries to 0,

$$\begin{aligned} \mathbf{G}_S &= \mathbf{Z}_{T(D+1)}^\top (\mathbf{Z}_{S(D+1)} \mathbf{F}_S \mathbf{D}_S) \mathbf{D}_T \\ \mathbf{G}_T &= \mathbf{Z}_{T(D+1)}^\top (\mathbf{Z}_{T(D+1)} \mathbf{F}_T \mathbf{D}_T) \mathbf{D}_T \\ \mathbf{H}_W &= \gamma \mathbf{D}_{TT} \mathbf{F}_T \\ \mathbf{H}_D &= \gamma (\mathbf{W}_{TT} \mathbf{F}_T + \mathbf{W}_{ST}^\top \mathbf{Y}_S) \end{aligned} \quad (8)$$

and $\text{Proj}_{\mathcal{C}}$ is a proximal operator that projects the argument onto the set $\mathcal{C}$, where each row sums up to 1 and $\mathbf{F}_L = \mathbf{Y}_L$.

2) *Optimization of $\mathcal{P}$ and $\mathcal{Q}$* : According to (1) we have

$$\llbracket \llbracket \mathcal{X}_S; \mathcal{P} \rrbracket; \mathcal{P}^\top \rrbracket = \mathcal{X}_S \times_1 (\mathbf{P}^{(1)\top} \mathbf{P}^{(1)}) \times_2 \dots \times_D (\mathbf{P}^{(D)\top} \mathbf{P}^{(D)}).$$

For further computation, it can be reformulated as a multiplication form $(\llbracket \llbracket \mathcal{X}_S; \mathcal{P} \rrbracket; \mathcal{P}^\top \rrbracket)_{(k)} = \mathbf{P}^{(k)\top} \mathbf{P}^{(k)} \mathbf{X}_{S(k)} \mathbf{O}_{\neq k}$, where

$$\begin{aligned} \mathbf{O}_{\neq k} &= (\mathbf{P}^{(D)\top} \mathbf{P}^{(D)}) \otimes \dots \otimes (\mathbf{P}^{(k+1)\top} \mathbf{P}^{(k+1)}) \\ &\otimes (\mathbf{P}^{(k-1)\top} \mathbf{P}^{(k-1)}) \otimes \dots \otimes (\mathbf{P}^{(1)\top} \mathbf{P}^{(1)}). \end{aligned}$$

By fixing $\mathcal{Q}$ and $\mathbf{P}^{(j)}$, $j \neq k$, we can solve $\mathbf{P}^{(k)}$ using

$$\begin{aligned} \min_{\mathbf{P}^{(k)}} & \lambda_1 \left\| \mathbf{X}_{S(k)} - \mathbf{P}^{(k)\top} \mathbf{P}^{(k)} \mathbf{X}_{S(k)} \mathbf{O}_{\neq k} \right\|_F^2 + \sum_{c=1}^C \left\| \mathbf{P}^{(k)} \frac{1}{N_S^c} \right. \\ & \times \sum_{i=1}^{N_S^c} (\mathbf{X}_{s,i}^c)_{(k)} \mathbf{P}_{\neq k}^{(k)} - \mathbf{Q}^{(k)} \frac{1}{N_T^c} \sum_{j=1}^{N_T^c} (F_T)_{jc} (\mathbf{X}_{t,j})_{(k)} \mathbf{Q}_{\neq k} \left. \right\|_F^2 \\ \text{s.t., } & \mathbf{P}^{(k)} \mathbf{P}^{(k)\top} = \mathbf{I}. \end{aligned} \quad (9)$$

We rewrite the above minimization as

$$\begin{aligned} \min_{\mathbf{P}^{(k)}} & \lambda_1 \left\| \mathbf{X}_{S(k)} - \mathbf{P}^{(k)\top} \mathbf{P}^{(k)} \mathbf{X}_{S(k)} \mathbf{O}_{\neq k} \right\|_F^2 \\ & + \left\| \mathbf{P}^{(k)} \mathbf{G} - \mathbf{V} \right\|_F^2, \text{ s.t., } \mathbf{P}^{(k)} \mathbf{P}^{(k)\top} = \mathbf{I}, \end{aligned} \quad (10)$$

where $\mathbf{G}$ and $\mathbf{V}$ are formed by concatenating the matrices $\frac{1}{N_S^c} \sum_{i=1}^{N_S^c} (\mathbf{X}_{s,i}^c)_{(k)} \mathbf{P}_{\neq k}^{(k)}$ and $\frac{1}{N_T^c} \sum_{j=1}^{N_T^c} (F_T)_{jc} (\mathbf{X}_{t,j})_{(k)} \mathbf{Q}_{\neq k}$,

Algorithm 1 HTDA algorithm

Input: $\mathcal{X}_S, \mathcal{X}_T, \mathbf{Y}_S$ and $\mathbf{Y}_L$, parameter $\{p_k\}_{k=1}^D, \lambda_1, \lambda_2, \gamma, \epsilon$, KNN parameter k , maximum iteration number M .

- 1: Reshape the source/target tensor to the same order D .
- 2: Initialize $\mathbf{F}_T$, factor matrices $\mathcal{P}$ and $\mathcal{Q}$, set $J^0 = 0, t = 1$.
- 3: **repeat**
- 4: Compute similarity matrix $\mathbf{W}$ using KNN.
- 5: Update $\mathbf{F}_T, \{\mathbf{P}^{(k)}\}_{k=1}^D, \{\mathbf{Q}^{(k)}\}_{k=1}^D$ using (7) and (11).
- 6: Compute loss J^t using (5) and $e = (J^t - J^{t-1})^2$.
- 7: $t = t + 1$
- 8: **until** $t = M$ or $e < \epsilon$.

Output: $\mathbf{F}_T$.

$c = 1, \dots, C$ along the columns, respectively. Further, the problem above can be simplified as

$$\min_{\mathbf{P}^{(k)}} \left\| \mathbf{P}^{(k)} \mathbf{G} - \mathbf{V} \right\|_F^2 - \lambda_1 \left\| \mathbf{P}^{(k)} \mathbf{X}_{S(k)} \Lambda \right\|_F^2 \quad (11)$$

s.t., $\mathbf{P}^{(k)} \mathbf{P}^{(k)\top} = \mathbf{I}$,

where $\Lambda = (\mathbf{P}^{(D)} \otimes \dots \otimes \mathbf{P}^{(k+1)} \otimes \mathbf{P}^{(k-1)} \otimes \dots \otimes \mathbf{P}^{(1)})^\top$. (11) can be efficiently solved using the algorithm described in [24]. Further, $\mathbf{Q}^{(k)}$ can be solved in the same way.

3.5. Tensor data with different orders

In this section, we describe our approach to handling HTDA in scenarios where tensors from the source and target domains have different orders. In many cases, multi-dimensional features exhibit various types of dimensions, such as ‘spatial’, ‘temporal’, and ‘channel’. When dealing with tensor features of different orders, a straightforward strategy is to align corresponding types. For instance, if the source features are 1-D vector features with size m_1 , while the target features are 3-D features extracted from neural networks with size $n_1 \times n_2 \times n_3$ (where n_1 and n_2 represent ‘spatial’ dimensions and n_3 represents the ‘channel’ dimension), we can treat the vector feature as having only the ‘channel’ dimension and reshape it into a 3-D feature with size $1 \times 1 \times m_1$. This allows the application of the proposed algorithms to tensors of different orders. The feasibility of this strategy will be verified in the experimental results section. Finally, Algorithm 1 outlines the procedure.

4. EXPERIMENTAL RESULTS

We conduct several experiments to evaluate the performance of the proposed algorithm. We compare the performance with existing vector-based HDA algorithms, including CDLS [10], SHFA [8], TNT [9] and ICDM [12]. SVM is used as the baseline method. We also compare with a vector-based version of our algorithm (i.e., the inputs are vectorized features), which we term HDA with label propagation (HDA-LP).

We evaluate the performance on three popular DA datasets, namely *Office-Caltech*, *Office31*, and *ImageCLEF*. The eval-

Table 1. Datasets and features used in the experiments

Dataset	Domain	Features
Office-Caltech	Amazon(A)	SURF (800 × 1) VGG16 (6 × 6 × 512)
	Caltech(C)	
	DSLR(D)	
	Webcam(W)	
Office31	Amazon(A)	Dcaaf6 (4096 × 1) VGG16 (6 × 6 × 512)
	DSLR(D)	
	Webcam(W)	
ImageCLEF	Bing(B)	SIFT (1024 × 1) VGG16 (6 × 6 × 512) ResNet50 (7 × 7 × 2048)
	ImageNet(I)	
	Pascal(P)	

uation involves five distinct types of features: SURF, SIFT, Dcaaf6, VGG16, and ResNet50. Table 1 provides information about the datasets and features used in each dataset. Specifically, the VGG16 feature is obtained by performing average pooling on the output of the ‘conv5_block3’ layer of the VGG16 network. The ResNet50 feature is extracted from the ‘block5_conv3’ layer of the ResNet50 network. For VGG16, the input images are resized to 256 × 256, while for ResNet50, the input images are resized to 224 × 224. For the algorithms that can only deal with vector-based features, we vectorize the tensor into a long vector as the feature representation. It is worth noting that the CDLS algorithm runs out of memory when the source feature is VGG16 and ResNet50; therefore, we only report its results for the *Office31* dataset.

For the proposed HTDA algorithm, we set the parameters as: $p_k = \min\{m_k, n_k, 30\}$ for $k = 1, \dots, D$, $\lambda_1 = \lambda_2 = 0.1$, $\gamma = 5 \times 10^{-4}$, $k = 10$, $M = 20$ and $\epsilon = 1 \times 10^{-5}$. The average classification accuracy is computed over 20 Monte Carlo runs, with different selections of source and labeled target data. The numbers $\{N_S^c\}_{c=1}^C$ of source points from the different classes are set to the same value (denoted N_S^c) for all classes. In each run, we randomly choose N_S^c source data per class to construct the source input tensor $\mathcal{X}_S$ and N_L^c target data per class as labeled target data to construct $\mathbf{Y}_L$. All experiments were conducted using MATLAB R2023b on a desktop PC with a 2.5-GHz processor and 32GB of RAM.

We compare the adaptation performance on tensor-based target features for different numbers of labeled target data. For all experiments, the number of source data per class N_S^c is set to 20. For the *Office-Caltech* dataset, the number of labeled target data per class N_L^c is varied from 0.5 to 3. Here, $N_L^c = 0.5$ means that half of the classes have no labeled target data, while the remaining half have one labeled target data point per class. For the *Office31* and *ImageCLEF* datasets, N_L^c is set to 1 and 3. Fig. 1 depicts the average classification accuracy versus N_L^c on *Office-Caltech*: SURF→VGG16, while the average accuracy on the *Office31*: Dcaaf6→VGG16 and *ImageCLEF*: VGG16→ResNet50 is shown in Table 2 and Table 3. As can be seen, the proposed HTDA consistently outperforms other methods in terms of adaptation performance across all tasks.

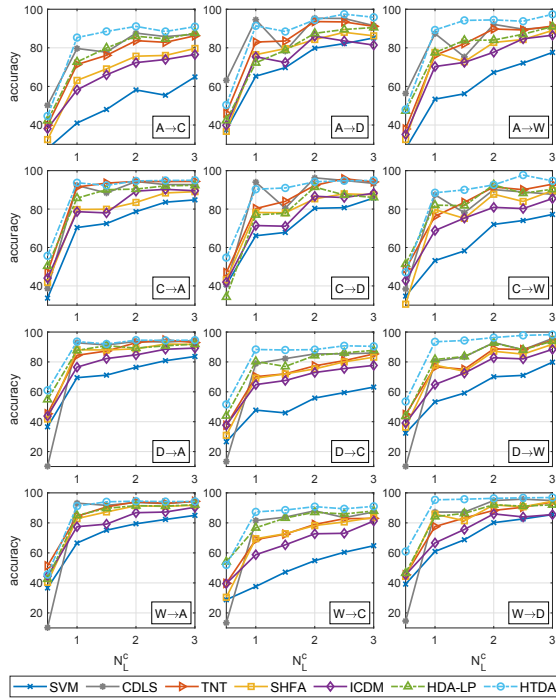
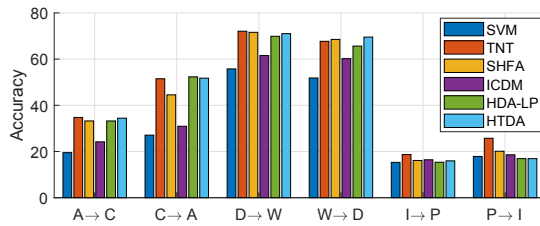
Furthermore, we explore the adaptation performance when transitioning from tensor-based source features to vector-based target features. Fig. 2 illustrates the performance across six

Table 2. Average classification accuracy (%) for different N_L^c on the *Office31* Dataset: Decaf6→VGG16.

Methods	A→D		D→A		A→W		W→A		D→W		W→D	
	$N_L^c=1$	$N_L^c=3$	$N_L^c=1$	$N_L^c=3$	$N_L^c=1$	$N_L^c=3$	$N_L^c=1$	$N_L^c=3$	$N_L^c=1$	$N_L^c=3$	$N_L^c=1$	$N_L^c=3$
SVM	31.82	56.33	28.14	43.67	27.23	55.09	28.18	44.05	26.10	53.87	31.56	57.30
CDLS	71.93	80.71	54.83	65.96	68.52	82.77	54.55	65.99	70.86	85.17	72.21	84.43
TNT	63.19	76.49	49.96	63.82	61.92	78.49	49.39	62.96	66.71	80.14	66.23	81.31
SHFA	64.69	77.80	46.37	62.06	58.47	76.15	46.69	62.07	61.06	78.57	64.50	80.21
ICDM	52.25	68.79	43.07	56.42	48.66	69.09	43.96	56.49	47.76	69.09	49.19	68.86
HDA-LP	54.80	68.50	51.21	63.29	56.64	69.14	52.49	64.14	60.37	72.20	57.77	70.83
HTDA	73.15	82.47	58.02	68.04	73.82	83.39	56.07	67.63	79.74	85.75	78.46	84.82

Table 3. Average classification accuracy (%) under different N_L^c on *ImageCLEF* Dataset: VGG16→ResNet50.

Methods	I→P		P→I		B→I		I→B		B→P		P→B	
	$N_L^c=1$	$N_L^c=3$	$N_L^c=1$	$N_L^c=3$	$N_L^c=1$	$N_L^c=3$	$N_L^c=1$	$N_L^c=3$	$N_L^c=1$	$N_L^c=3$	$N_L^c=1$	$N_L^c=3$
SVM	22.87	38.56	31.58	49.68	26.48	53.09	15.66	24.55	22.93	37.31	17.24	25.42
TNT	48.88	62.19	71.05	79.87	67.07	80.28	28.45	43.61	44.42	56.44	31.07	43.17
SHFA	45.90	56.38	51.12	68.95	51.82	67.76	27.28	39.80	40.61	52.51	26.77	35.50
ICDM	37.72	50.18	55.46	73.21	50.24	70.99	25.97	36.98	37.91	48.99	27.04	33.94
HDA-LP	54.90	64.26	72.91	83.99	69.80	82.77	29.47	46.48	47.38	60.35	33.52	45.27
HTDA	62.52	68.97	82.52	85.95	77.81	84.15	39.80	49.27	58.18	63.98	38.11	48.03

**Fig. 1.** Average classification accuracy versus N_L^c on *Office-Caltech*: SURF→VGG16.**Fig. 2.** Average classification accuracy under vector-based target features. Columns 1-2: *Office-Caltech*: VGG16→SURF; columns 3-4: *Office31*: VGG16→Decaf6; columns 5-6: *ImageCLEF*: ResNet50→SIFT.

domain tasks. It is evident that HTDA achieves comparable results with TNT and SHFA in most tasks. The findings demonstrate that our algorithm is particularly effective in improving performance when the target domain involves tensor data. Nevertheless, our algorithm provides a versatile solution, effective for both tensor and vector-based target features.

5. CONCLUSION

In this work, we investigated a novel domain adaptation challenge termed heterogeneous tensor domain adaptation (HTDA), where the features in both source and target domains are high-order tensors with distinct sizes and orders. We introduced a new HTDA solution that encompasses MPCA-based tensor projection, tensor class-wise distribution alignment, and tensor correlation-based label propagation. Additionally, we developed an algorithm tailored to the HTDA problem. Experimental findings demonstrate the superior performance of our proposed method compared to existing vector-based HDA algorithms when dealing with tensor features.

6. REFERENCES

- [1] Baochen Sun, Jiashi Feng, and Kate Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016, vol. 30.
- [2] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu, "Transfer feature learning with joint distribution adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2200–2207.
- [3] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, "Adversarial discriminative domain adaptation,"

in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.

- [4] Joey Tianyi Zhou, Ivor W Tsang, Sinno Jialin Pan, and Mingkui Tan, “Multi-class heterogeneous domain adaptation,” *Journal of Machine Learning Research*, 2019.
- [5] Feng Liu, Guangquan Zhang, and Jie Lu, “Heterogeneous domain adaptation: An unsupervised approach,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5588–5602, 2020.
- [6] Zhen Fang, Jie Lu, Feng Liu, and Guangquan Zhang, “Semi-supervised heterogeneous domain adaptation: Theory and algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1087–1105, 2022.
- [7] Chang Wang and Sridhar Mahadevan, “Heterogeneous domain adaptation using manifold alignment,” in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, 2011, vol. 22, p. 1541.
- [8] Wen Li, Lixin Duan, Dong Xu, and Ivor W Tsang, “Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1134–1148, 2013.
- [9] Wei-Yu Chen, Tzu-Ming Harry Hsu, Yao-Hung Hubert Tsai, Yu-Chiang Frank Wang, and Ming-Syan Chen, “Transfer neural trees for heterogeneous domain adaptation,” in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part V 14*. Springer, 2016, pp. 399–414.
- [10] Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang, “Learning cross-domain landmarks for heterogeneous domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5081–5090.
- [11] Jingjing Li, Ke Lu, Zi Huang, Lei Zhu, and Heng Tao Shen, “Heterogeneous domain adaptation through progressive alignment,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1381–1391, 2018.
- [12] Hanrui Wu, Hong Zhu, Yuguang Yan, Jiaju Wu, Yifan Zhang, and Michael K Ng, “Heterogeneous domain adaptation by information capturing and distribution matching,” *IEEE Transactions on Image Processing*, vol. 30, pp. 6364–6376, 2021.
- [13] Hao Lu, Lei Zhang, Zhiguo Cao, Wei Wei, Ke Xian, Chunhua Shen, and Anton van den Hengel, “When unsupervised domain adaptation meets tensor representations,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 599–608.
- [14] Yao Qin, Lorenzo Bruzzone, and Biao Li, “Tensor alignment based domain adaptation for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9290–9307, 2019.
- [15] Ali Braytee, Mohamad Naji, and Paul J Kennedy, “Unsupervised domain-adaptation-based tensor feature learning with structure preservation,” *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 3, pp. 370–380, 2022.
- [16] Wentao Mao, Zongtao Chen, Yanna Zhang, and Zhidan Zhong, “Harmony better than uniformity: A new pre-training anomaly detection method with tensor domain adaptation for early fault evaluation,” *Engineering Applications of Artificial Intelligence*, vol. 127, pp. 107427, 2024.
- [17] Ledyard R Tucker, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
- [18] Haiping Lu, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos, “Mpca: Multilinear principal component analysis of tensor objects,” *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.
- [19] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola, “A kernel method for the two-sample-problem,” *Advances in Neural Information Processing Systems*, vol. 19, 2006.
- [20] 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, 2010.
- [21] Zhengming Ding, Sheng Li, Ming Shao, and Yun Fu, “Graph adaptive knowledge transfer for unsupervised domain adaptation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 37–52.
- [22] Yun Fu and Thomas S Huang, “Image classification using correlation tensor analysis,” *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 226–234, 2008.
- [23] Handong Zhao, Zhengming Ding, and Yun Fu, “Multi-view clustering via deep matrix factorization,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, vol. 31.
- [24] Zaiwen Wen and Wotao Yin, “A feasible method for optimization with orthogonality constraints,” *Mathematical Programming*, vol. 142, no. 1–2, pp. 397–434, 2013.