Low-Rank Tensor Learning for Human Action Recognition

A Dissertation Presented
by

Chengcheng Jia

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Northeastern University
Boston, Massachusetts

April 2016
I dedicate my dissertation work to my family and many friends. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears.

I also dedicate this dissertation to my many friends who have supported me throughout the process. I will always appreciate all they have done, especially Ming Shao for helping me develop my technology skills, Hongfu Liu for the many hours of proofreading, and Yue Wu for helping me to set the academic environment.

I dedicate this work and give special thanks to my friend Haiyi Mao and Xiaoyu Wang for spiritual guidance. Both of you have been my best cheerleaders.
Contents

List of Figures v
List of Tables ix
Acknowledgments x
Abstract of the Dissertation xi

1 Introduction 1
1.1 Background ................................................................. 1
1.2 Related Work ............................................................... 2
1.2.1 Human Action Recognition ............................................. 2
1.2.2 Low-Rank tensor Learning ............................................. 3
1.2.3 Different Problems .................................................... 3
1.2.4 Organization ............................................................. 4

2 Low-Rank Tensor Learning 5
2.1 Abstract ................................................................. 5
2.2 Introduction .............................................................. 5
2.3 Preliminary ............................................................... 8
2.4 Low-rank Tensor Completion .......................................... 8
2.4.1 Discriminant analysis .................................................. 10
2.4.2 Objective function .................................................... 11
2.4.3 Optimization ........................................................ 11
2.4.4 Improvement .......................................................... 12
2.5 Experiment Results .................................................... 14
2.5.1 On the MSR hand gesture 3D database .......................... 14
2.5.2 On the MSR action 3D database ................................... 16
2.5.3 On CMU Multi-PIE face database ................................ 17
2.6 Conclusion ............................................................... 18
2.7 Low-Rank Tensor Analysis for Subspace Learning ............... 19
2.7.1 Overview of Our Framework ....................................... 20
2.7.2 Preliminary Knowledge ............................................... 20
2.7.3 Optimization .......................................................... 25
2.7.4 Discussion ................................................................. 26
2.8 Experiment ................................................................. 27
  2.8.1 Datasets ................................................................. 28
  2.8.2 Parameters Setting ..................................................... 28
  2.8.3 Comparison Methods .................................................. 28
  2.8.4 MSRpair3D Action Dataset .......................................... 29
  2.8.5 MSRpair Skeleton Dataset .......................................... 31
  2.8.6 MSRdaily3D Action Dataset ......................................... 34
  2.8.7 UTKinect-Action Dataset ........................................... 38
  2.8.8 Discussion .............................................................. 38
2.9 Conclusion ................................................................. 39

3 Low-Rank Tensor in Transfer Learning .................................. 40
  3.1 abstract ................................................................. 40
  3.2 Introduction .............................................................. 41
    3.2.1 Our Contributions ..................................................... 44
  3.3 Related work .......................................................... 44
    3.3.1 Action Recognition ................................................ 45
    3.3.2 Transfer Learning .................................................. 46
    3.3.3 Semi-Supervised Transfer Learning ............................. 47
    3.3.4 Tensor Representation .......................................... 47
  3.4 Latent Low-rank Tensor Transfer Learning ......................... 48
    3.4.1 Preliminary .......................................................... 48
    3.4.2 Motivation ........................................................... 49
    3.4.3 Latent Problem Formulation ..................................... 50
    3.4.4 Cross-Modality Regularizer ..................................... 52
    3.4.5 Optimization ........................................................ 53
    3.4.6 Semi-Supervised Low-Rank Transfer Learning ................ 54
    3.4.7 Complexity Analysis .............................................. 56
    3.4.8 Discussion .......................................................... 56
  3.5 Experiments ............................................................ 57
    3.5.1 Datasets ............................................................. 57
    3.5.2 Experimental Settings ............................................ 58
    3.5.3 Compared methods ............................................... 60
    3.5.4 Comparison results ............................................... 61
    3.5.5 Discussion .......................................................... 70
  3.6 Conclusion ............................................................. 71

4 Deep Non-Negative Tensor Factorization for Time Alignment .......... 73
  4.1 abstract ................................................................. 73
  4.2 Introduction ............................................................ 74
  4.3 Related Work .......................................................... 76
  4.4 Sparse Canonical Temporal Alignment ................................ 77
  4.5 Deep Non-Negative Tensor Factorization ............................ 79
    4.5.1 Non-Negative Tensor Factorization (NTF) ...................... 79
List of Figures

2.1 Framework of the proposed algorithm for action recognition. The tensor training set $\mathcal{X}$ is used for calculating the low-rank projection matrices, which are employed for subspace alignment of training and testing action videos $\mathcal{Y}$ and $\mathcal{Y}'$. .................................................. 7

2.2 Illustration of 2, 3-dimensional projection matrices respectively. Form left to right: $U_1$, $U_2$ and $U_3$. During learning, each projection matrix is calculated by the alternative direction method. .................................................. 9

2.3 Key frames of different actions/gestures of (1) MSR action 3D database and (2) MSR hand gesture database on the first and second rows, respectively. ................................. 14

2.4 Illustration of the projection matrices on the MSR Hand Gesture database. The left column are our result, while the right column shows Zhong and Cheriet’s. Form top to bottom: $U_1$, $U_2$, $U_3$. .................................................. 15

2.5 Left to right: Zhong and Cheriet’s and our learned projected gestures of the MSR hand gesture database. .................................................. 16

2.6 Accuracy of the proposed method under different parameter settings of $\lambda$ and $\beta$ on two used databases. .................................................. 17

2.7 Accuracy with different parameters set for the MSR action 3D database. Top to bottom: Test One, Test Two, and Cross Subject Test. Each test contains three sets AS1, AS2, AS3, respectively. .................................................. 18

2.8 PIE face database. Top row: the original faces; second row: the low-rank faces; third row: the errors. .................................................. 18

2.9 Schematic of our general model. In the right part, all the tensor samples are distributed in a manifold $\mathcal{M}$, and we construct a graph $\mathcal{L}$ with the intra-class neighbors. Low-rank matrix $Z$ contains local information by the constraint $\text{tr}(Z^TLZ)$ because of neighbor information in $\mathcal{L}$. Besides, $Z$ also contains the class information because of the intra-class neighbors in $\mathcal{L}$. The discriminant information is benefit for classification in the learned subspace. The left part aims to find the tensor subspace by low-rank learning of projection matrix $U_n$, which is calculated by the truncated tucker decomposition. The original tensor $\mathcal{X}$ is decomposed to be 3 projection matrices $U_n$ ($n = 1, 2, 3$) and the core tensor $G$. After learning $U_n$ and $Z$ alternatively, we can find the tensor subspace for RGB-D action recognition. .................................................. 22

2.10 Depth frames of various actions of different datasets. .................................................. 30

2.11 MSRpair3D dataset: accuracy under different dimensions of tensors, where the dimension is $k \times k \times k$ $(k = 1, \ldots, 16)$. .................................................. 32
2.12 MSRpair3D dataset: confusion matrix of our method. 32
2.13 MSRpair3D dataset: accuracy under different value of $\beta$ and $\eta$. 33
2.14 MSRpair3D dataset: the optimization on 3 modes of training tensors. Each curve of our algorithm LRTS which indicates the value of objective function of one mode converges to a stationary point within 20 iterations. 33
2.15 MSRpair3D dataset: reconstruct matrix $Z$ under 5, 7 and 9 iterations. We can see $Z$ contains the local structure, i.e., the class information of the training set. 33
2.16 MSRpair skeleton dataset: accuracy under different dimensions of tensors, where the dimension is $k \times k \times k$ ($k = 1, \ldots, 28$). 34
2.17 MSRdaily3D dataset: accuracy under different dimensions of tensors. 36
2.18 MSRdaily3D dataset: confusion matrix of our method. 36
2.19 MSRdaily3D dataset: low-rank $Z$ under different iterations. Left to right: $Z$ under 2, 5 and 8 iterations respectively. 37
2.20 Accuracy on Reduced Version of UTKinect-Action Dataset. 37

3.1 In our framework, the source domain $X_S$ is the newly generated RGB-D dataset, which contains both RGB and depth action samples. The target domain $X_T$ is the traditional RGB dataset, which only includes the RGB actions. In order to make use of the existing depth information to help recognize the traditional RGB data, we transfer the correlation of two modalities from the source domain to the target domain by aligning them in the formula $U_nX_S = U_nX_TZ + LU_nX_S + E$, to find the latent information $L$ which helps to recover "missing" modality in the target domain. To find the correlation, we construct a graph regularizer $\text{tr}(ZLZ^T) + \text{tr}(Z^TGGZ)$ containing the class information of the source domain, then transfer the correlation by low-rank learning of the alignment. 41
3.2 Illustration of tensor mode-$n$ unfolding. An action sequence is represented as a RGB-D tensor $X$, and we get mode-$n$ unfolding matrix $X^{(n)}$ along its $x, y, z$-axis. 49
3.3 UTKinect-Action dataset, (a) the RGB walking sequence and (b) the corresponding depth sequence. 59
3.4 MSR DailyActivity3D Dataset, (a) is the depth sequence of sitting, (b) is the HOG feature of RGB sequence. 62
3.5 Illustration of performance by $U_3$. (a) RGB samples; (b) HOG feature and eigenAction. 65
3.6 Illustration of mode-$n$ projection matrices. $U_1, U_2$ perform on the mode-1, 2 feature, while $U_3$ on the sequence dimension. 66
3.7 Accuracy (%) of Test 6 (RGB - RGB) on UTKinect-Action Dataset. 65
3.8 Accuracy (%) of Test 7 (Depth - Depth) on UTKinect-Action Dataset. 66
3.9 Accuracy (%) of Test 8 (RGB - Depth) on UTKinect-Action Dataset. 66
3.10 Convergence property of SLTTL $(\beta_2 = 0)$, sum of mode-$n$ error with different number of iterations on [Test 6: Case 5]. 67
3.11 Accuracy (%) of Test 9 (Depth - RGB) on UTKinect-Action Dataset. 68
3.12 Accuracy (%) of Test 10 (RGB - RGB-D) on UTKinect-Action Dataset. 68
3.13 The accuracy under different dimensions of Test 6 ~ 10. Different tests indicate different modality transferring, i.e., (1) RGB-RGB, (2) Depth-Depth, (3) RGB-Depth, (4) Depth-RGB and (5) RGB-(RGB-D). 72
4.1 Illustration of the proposed SCTA framework. Suppose $X$ is the video dataset which can be decomposed into two low-rank tensors: $W, H$, where $W$ is low-dimensional projection tensor, and $H$ is the corresponding coefficients. The sparse key frames selected by SCTA in the tensor space make sure that two intra-class video sequences are well-aligned without redundant information, which is able to further boost the action recognition performance in the tensor subspace sought out by our deep NTF mechanism. .................................................. 74

4.2 Schematic illustration of mode-1 deep non-negative tensor factorization (DNTF). $X$ is a third-order tensor in the first layer of DNTF. Two matrices $W_1$ and $H_1$ are obtained by mode-1 NTF, followed by TT decompositions in the second layer to obtain $W_{1\text{deep}}$ and $H_{1\text{deep}}$. .................................................. 80

4.3 Illustration of tensor-train (TT) decomposition. Note rectangles indicate tensor cores, while the circles indicate auxiliary indices. .................................................. 81

4.4 Synthetic data evaluations. Original triple sequences $X_i, i \in \{1, 2, 3\}$ are generated first, with additional Gaussian noises in the third dimension. Spatiotemporal warping functions are calculated by pDTW, pDDTW, pIMW, pCTW, GTW and SCTA, respectively. pCTW, GTW and SCTA are based on CCA to align the homogeneous resources, and rule out the noises from the third dimension. sub-figure on upper right shows different warping paths, while that on bottom right indicates mean alignment errors. .................................................. 87

4.5 Action samples from two datasets. Top: “drinking” from MSRDailyActivity3D dataset. Bottom: “lifting a box” from MSRActionPairs dataset. .................................................. 88

4.6 Accuracy of DNTF on MSRActionPairs dataset for multi-subject problem. .................................................. 91

4.7 Visualization of temporal alignment of key frames from two intra-class action sequences with 20 frames. The first two rows show the result for multi-subject challenge in “depth” modality, from which we can see that the key frames of “putting things on chair” actions are well aligned. The last two rows show the result for multi-modality challenge, from which we can see that the key frames of “putting things on floor” actions are aligned as well. .................................................. 92

4.8 Illustration of temporal key frames alignment on MSRDailyActivity3D dataset. Left: Sit action. Solid lines connect the key frames between two sequences. Middle: SSM of two action sequences by [1]. The red curve is the realistic aligned frames while the green dots are the aligned key frames. Right: SSM by our method after sampling. The green dots indicate the key frames while the red curve shows the aligned path. .................................................. 93

4.9 Mean accuracy of the proposed method with $\lambda_1 \in [0, 1], \lambda_2 = \lambda_3 = 1$ under different dimension settings on two datasets. Note $\lambda_1 = 0$ is baseline, and we can see the accuracy under $\lambda_1 > 0$ is higher than that of $\lambda_1 = 0$. .................................................. 94

4.10 First three sub-figures: mode-n error in different iterations on MSRActionPairs dataset. Fourth sub-figure: accuracy under mode-n ($n = 1, 2$) dimensions. We can see that mode-3 dimension $\text{Dim}(3) = 7$ achieve relatively good results. .................................................. 95

4.11 Accuracy with different penalty factors $\lambda_2$ and $\lambda_3$ for left: sub-action and right: multi-modality challenges. We can see that the accuracy under $\lambda_2, \lambda_3 > 0$ is better than that of $\lambda_2 = 0$. .................................................. 98
4.12 Objective function value (OFV) of our model on MSRActivityPairs dataset. Left: mode-1,2 OFV. Right: mode-3 OFV. The OFVs of all modes will not change after a few iterations.

4.13 RRSE under various noise levels (SNR). Left: dataset size is 96. Right: dataset size is 192.

4.14 Running time comparisons of TT and tucker decomposition given different numbers of training data.
# List of Tables

2.1 MSR gesture database. ................................................. 17  
2.2 MSRaction3D database. ................................................. 17  
2.3 Accuracy (%) of 3 sets on the MSR action database. ............... 19  
2.4 Accuracy (%) of MSRPair3D Dataset. ................................. 31  
2.5 Accuracy (%) of MSRpair Skeleton Dataset. .......................... 35  
2.6 Accuracy (%) of MSRdaily3D Dataset. ................................. 35  
2.7 Accuracy (%) of UTKinect-Action Dataset. ........................... 37  
2.8 Accuracy (%) of Skeleton-Based Methods on UTKinect-Action Dataset .......................................................... 38  
3.1 Training and testing settings for source and target domain. .......... 42  
3.2 Accuracy (%) of Test 1 on MSRDailyActivity3D and MSRActionPairs Datasets, with label information. ................................. 61  
3.3 Accuracy (%) of Test 2 on MSRDailyActivity3D and MSRActionPairs Datasets, with label information. ................................. 63  
3.4 Accuracy (%) of Test 3 on MSRDailyActivity3D and MSRActionPairs Datasets, with label information. ................................. 63  
3.5 Accuracy (%) of Test 4 on MSRDailyActivity3D and MSRActionPairs Datasets, with label information. ................................. 64  
3.6 Accuracy (%) of Test 5 on MSRDailyActivity3D and MSRActionPairs Datasets, with label information. ................................. 65  
3.7 Training time cost of methods. ........................................ 69  
4.1 Notations and Descriptions. ......................................... 78  
4.2 Accuracy of sub-action and multi-subject problems. .................. 90  
4.3 Accuracy of multi-modality problem, Train-Test: RGB-Depth & Depth-RGB. ......................................................... 91  
4.4 Accuracy (%) of MSRDailyActivity3D with setting [10, 10, 10]. ........... 96  
4.5 Accuracy (%) of MSRDailyActivity3D with setting [40, 40, 10]. ........... 96  
4.6 Accuracy (%) of MSRActionPairs with setting [20, 20, 20]. ............... 97  
4.7 Accuracy (%) of MSRActionPairs with setting [40, 40, 20]. ............... 97
Acknowledgments

I would first like to thank my PhD advisor Professor Yun Fu, who has been supervising my research for the past three years. He gives me guidance in every aspect, including research topic, logic writing and academic presentation. I appreciate his guidance and I have a considerable progress after I join in our group. I wish I could learn more from my professor, and it is really a great honor for me to work in SMILE group.

I would also like to thank my PhD committee members professor Yizhou Sun and Professor Stratis Ioannidis, who have been advising me and providing valuable suggestions ever since my PhD research proposal.

In addition, I would like to thank all the members of SMILE Lab, including: Dr. Yu Kong, Dr. Jun Li, Ming Shao, Sheng Li, Zhengming Ding, Shuyang Wang, Handong Zhao, Hongfu Liu, Shuhui Jiang, Haiyi Mao, Yue Wu, Zhiqiang Tao, Joseph Robinson. They are diligent and very generous for helping me on my research. Thank you!

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study. Thank you!

My research is supported in part by the NSF CNS award 1314484, Office of Naval Research award N00014-12-1-1028, U.S. Army Research Office grant W911NF-13-1-0160, ONR Young Investigator Award N00014-14-1-0484, U.S. Army Research Office Young Investigator Award W911NF-14-1-0218, NPS award N00244-15-1-0041.
Abstract of the Dissertation

Low-Rank Tensor Learning for Human Action Recognition

by

Chengcheng Jia

Doctor of Philosophy in Electrical and Computer Engineering
Northeastern University, April 2016
Dr. Advisor: Yun Fu, Adviser

In the era of social media, social security is an important topic such as public surveillance. Even tons of web cameras are widely used in public places, e.g., bank, square and airport, it still needs lots of time and labors to keep attention. Therefore, how to recognize human action in a specific circumstance with a complex background is a basic question for social security problem.

In this thesis, we focus on the social security problem, in particular human action recognition, and give the analytics in two lines, (1) machine learning algorithms for action recognition, (2) applying algorithms for novel problems in action recognition, e.g., missing-modality problem, dimensionality reduction. These two lines are detailed in following.

For machine learning algorithm, extracting features from high-dimensional action data is crucial in human action recognition. The usual approach is finding a subspace, i.e., projecting high-dimensional data into a low-dimensional subspace containing main pattern of original data and fewer variables, for classification. First of all, data representation is crucial for action video which contains spatiotemporal information. To this end, we propose high-order tensor to represent the action videos, and employ tensor decomposition methods for dimensionality reduction. Second, different problems in action recognition tasks are solved by machine learning algorithms, such as transfer learning, low-rank learning, manifold learning.

Low-rank tensor learning is appropriate for some problems in action recognition task such as missing-modality problem, intra-class diversity problem and manual dimensional setting problem. In missing-modality problem, training ans testing data are different modalities, which may not obtain good performance in the test phase. Intra-class diversity problem contains three situations, such as multi-subject, multi-modality and sub-action challenges, which will decrease the performance of recognition. Sub-action problems indicates some body poses are different even for the same action, e.g., drinking action by standing person and drinking by sitting person.
Chapter 1

Introduction

1.1 Background

Human actions arouse widely attention in social media, as there are huge amount of action videos in Youtube everyday. In reality, human action recognition suffer from distinctive feature extraction from the original high-dimensional videos with complex background. There are some common problems existing in the action recognition task, which are listed as following:

- For dimensionality setting problem, the usual way to extract features from action videos is to project the high-dimensional data to a low-dimensional subspace, whose dimension is pre-defined. However, larger or smaller subspace dimension would decrease the performance in the subspace.

- For missing-modality problem, given a human action dataset with more than one modalities, it is not easy to get high accuracy of recognizing new data, if the training and testing data are RGB and depth modalities respectively. We consider this situation is missing-modality problem in the training phase.

- For intra-class diversity problem, there are some diversities because of multi-subject, multi-modality or sub-action in the same class of human activity. We are interested in aligning two intra-class action sequences in a new subspace in the training phase.

These problems can be consolidated to a single question: how to represent the high-dimensional action videos, and find a new subspace by some machine learning algorithms to solve those problems.
CHAPTER 1. INTRODUCTION

Tensor representation is important for action videos, as its multi-direction structure to present spatiotemporal information in a video. Low-rank tensor decomposition could be used for subspace learning, including tucker decomposition, CP decomposition and tensor-train decomposition.

In the second chapter of this study, we proposed a novel framework for tensor subspace learning. We utilized the low-rank constraint to find the subspace dimension automatically. We calculated the mode-n projection matrix by low-rank learning, and the rank is set to be subspace dimension. There are two benefits of our model. First, the number of dimension would not cause redundancy or insufficiency for recognition in the subspace. Second, we integrated discriminant information in our model for recognition in the subspace.

In the third chapter of this study, we proposed a novel framework for missing-modality problem by transfer learning. Suppose in the target domain, there are RGB data for training but depth data for testing, which would not get good performance because of different modalities. We call it missing-modality problem. Given an auxiliary source domain containing both RGB and depth data, in the training phase we explored the correlation of RGB and depth modalities in the source domain, which is transferred to the target domain to help recover additional "missing" depth information. In this way, we aimed to improve accuracy of recognizing missing modality data in the target domain. Experiments demonstrated our method could improve accuracy compared with other transfer learning methods.

In the fourth chapter of this study, we proposed a novel framework for intra-class diversity problems, e.g., multi-subject, multi-modality and sub-action. We selected key frames of two intra-class action sequences, and aligned them by the key frames. This procedure aims to reduce the diversity within class. Then we found a tensor subspace by deep Non-Negative Tensor Factorization to extract discriminative feature for recognition. Experiments demonstrated our method could improve accuracy compared with other subspace alignment methods.

1.2 Related Work

1.2.1 Human Action Recognition

Human action recognition is a crucial topic in social security area, e.g., there are tons of action videos in Youtube everyday, and more and more academic researches have focused on action recognition. The previous captured action datasets contain simple human activities and single RGB modality, while recently, action datasets including complex activities and multi-modalities have
been captured by Kinect device, which is cheap and easy to use. Accordingly, new problems have been explored for the new data, including data representation, dimensionality reduction approaches, missing-modality problem and intra-class diversities problem.

1.2.2 Low-Rank tensor Learning

The traditional way to represent an action video is unfolding a 3D video to be a 1D high-dimensional vector, which may damage the spatiotemporal structure of a video. Recently, many works use high-order tensor to represent a video \([136][59]\). Particularly, an action image sequence can be seen as a 3-order tensor, of which mode-1 and mode-2 represent the action image information and mode-3 describes the temporal information \([2]\).

Low-rank tensor learning (LrTL) \([3][4][5]\) has been shown to be an effective way for subspace learning, which has been successfully applied to various recognition related problems, such as action recognition \([6][7][8]\), face recognition \([9][10][11]\), and gait recognition \([12][13][14]\). LrTL is essentially the extension of vector or matrix analysis to higher-order tensor analysis. In the tensor subspace, the discriminant projection matrix of each mode \([15]\) is calculated alternately by fixing the other modes.

1.2.3 Different Problems

The first problem is subspace dimensionality setting. Defining the subspace dimension manually would decrease the performance of recognition with redundant or insufficient information caused by larger or smaller dimensions respectively. We utilized the low-rank constraint to find the subspace dimension automatically. We calculated the mode-n projection matrix by low-rank learning, and the rank is set to be subspace dimension. There are two benefits of our model. First, the number of dimension would not cause redundancy or insufficiency for recognition in the subspace. Second, we integrated discriminant information in our model for recognition in the subspace.

The second problem is called missing-modality problem. Given an auxiliary source domain containing both RGB and depth data, in the training phase. We explored the correlation of RGB and depth modalities in the source domain, which is transferred to the target domain to help recover additional "missing" depth information. In this way, we aimed to improve accuracy of recognizing missing modality data in the target domain.

The third problem is intra-class diversities, e.g., multi-subject, multi-modality and sub-action. In order to reduce the diversities, we selected key frames of two intra-class action sequences,
CHAPTER 1. INTRODUCTION

and aligned them by the key frames. This procedure aims to reduce the diversity within class. Then we found a tensor subspace by deep Non-Negative Tensor Factorization to extract discriminative feature for recognition.

1.2.4 Organization

The whole thesis is organized as following:

In Chapter 2, we propose a novel framework for tensor subspace learning to find the dimension automatically by low-rank learning. We first introduce tensor representation of action videos, and describe low-rank tensor decomposition method for dimensionality reduction. We integrate discriminant information into our model to help classification in the new subspace. Additionally, we add local information in the improved model, to make the intra-class neighbors be closer in the new subspace.

In Chapter 3, we explore a missing-modality problem in action recognition solved by transfer learning. We describe the missing-modality problem in different settings, and explain how to recover the "missing" modality as latent information in the target domain. We show the derivation of our model in details and compare the accuracy under different parameters in the experiment. In the semi-supervised model, we integrate the predicted target labels into our model to help improve the performance.

In Chapter 4, we propose a novel framework for intra-class diversity problems, e.g., multi-subject, multi-modality and sub-action. In order to reduce the diversities, we select key frames of two intra-class action sequences by sparse learning, and align them in a new subspace by temporal alignment, finally extract features using Non-Negative Tensor Factorization (NTF). In the improved model, we use a deep NTF framework to extract more discriminative features with two hidden layers scheme.

Finally, we give a conclusion of this thesis, talking about three different problems in human action recognition task, and the corresponding solutions by machine learning algorithms. Future work is also discussed in the end.
Chapter 2

Low-Rank Tensor Learning

2.1  Abstract

Tensor completion is an important topic in the area of image processing and computer vision research, which is generally built on extraction of the intrinsic structure of the tensor data. Drawing on this fact, action classification, relying heavily on the extracted features of high-dimensional tensors, may indeed benefit from tensor completion techniques. In this paper, we propose a low-rank tensor completion method for action classification, as well as image recovery. Since there may exist distortion and corruption in the tensor representations of video sequences, we project the tensors into a subspace, which contains the invariant structure of the tensors. In order to integrate useful supervisory information for classification, we adopt a discriminant analysis criterion to learn the projection matrices. The resulting multi-variate optimization problem can be effectively solved using the augmented Lagrange multiplier (ALM) algorithm. Experiments demonstrate that our method results with better accuracy compared with some other state-of-the-art low-rank tensor representation learning approaches on the MSR Hand Gesture 3D database and the MSR Action 3D database. By denoising the Multi-PIE face database, our experimental setup testifies the proposed method can also be employed to recover images.

2.2  Introduction

Images and video sequences can be naturally represented as high-dimensional tensors. However, the real tensor representations of images and videos are usually incomplete, due to missing elements or the presence of noise. This issue impels great research interest for recovering the
original tensors these past recent years. Many tensor representation learning approaches have been proposed \cite{16, 17, 18, 19, 20, 21, 22, 23, 24}. Many of these previous approaches aim to learn the low-dimensional representations of tensors, while mainly using the high-order singular value decomposition (HOSVD). Regardless, some tensor approximation approaches have been proposed as well, which, in general, estimate a rank-one tensor via vector outer product \cite{25, 26, 27, 28, 29, 30, 31}.

As of recent, several low-rank tensor representation learning approaches have been proposed for computer vision applications, such as image reflection and alignment \cite{32}, target tracking \cite{31}, face and object recognition \cite{33, 34}. These methods aim to learn the invariant structure of the tensor data. However, the formulation and optimization of these approaches are quite different. For concreteness, Zhang et al. performed the low-rank tensor representation learning on the original images, in parallel to eliminate noise and recover missing pixels \cite{32}; Shi et al. employed rank-one tensors for multi-target tracking \cite{31}; Ding et al. used rank-one tensors to reduce tensor dimensionality for applications such as video compression and face classification \cite{33}; Zhong and Cheriet proposed a manifold-based tensor representation learning model for face and object recognition \cite{34}. Note, although these low-rank tensor representation learning approaches have been successful when applied to different visual classification scenarios, they are rarely integrated in the supervisory information for maximizing class discrimination \cite{35, 36, 37}, which may dramatically improve the visual classification accuracy.

Some low-rank matrix learning approaches based on the discriminant analysis criterion have been addressed. For example, Zheng et al. used intra-class and inter-class information for face recognition \cite{38}, also, Cai et al. employed the discriminant analysis criterion with low-rank matrix learning for face and digits recognition \cite{39}. These discriminative low-rank matrix learning approaches have shown that the label information of data is typically beneficial for visual classification. Rare previous work was integrated the discriminant analysis criterion into a low-rank tensor completion model, to the best of our knowledge. This can be directly applied to visual classification applications, such as action classification.

In this paper, we present a supervised low-rank tensor completion method for dimensional reduction, to learn an optimal subspace for action video recognition. Our model automatically learns the low dimensionality of tensor, opposed to manually pre-defined, as other dimensional reduction methods. Considering the underlying structure information of the whole high-dimensional dataset, it can use the low-rank learning to extract the structure for image recovery, while integrating with the discriminant analysis criterion. Figure 2.1 shows the framework of our method applied to the
Figure 2.1: Framework of the proposed algorithm for action recognition. The tensor training set $\mathcal{X}$ is used for calculating the low-rank projection matrices, which are employed for subspace alignment of training and testing action videos $\mathcal{Y}$ and $\mathcal{Y}'$.

We first select a training set from an action video database to learn the low-rank projection matrices, which are then used to calculate a tensor subspace for the action classification. When calculating the low-rank projection matrices, we adopt a discriminant analysis criterion as a regularizer to avoid over-fitting. Meanwhile, with this discriminant analysis criterion, supervisory information is seamlessly integrated in the low-rank tensor completion model. After projecting the original training and testing sets to the learned tensor subspace, we predict the labels of the test video sequences with a K-nearest neighbor (KNN) classifier. We add the sample information to recovery some face images by removing different illuminations.

The contributions of this paper are as follows:

1. We proposed a new discriminative method for low-rank tensor completion, which automatically learns the low dimensionality of the tensor subspace for feature extraction.

2. We integrated the discriminant analysis criterion in the low-rank tensor completion model based on the given supervisory information.

3. The proposed model extracts the underlying structure of the original tensor data by low-rank learning, which reconstructs the data from the learned tensor subspace, for high-dimensional image recovery.
CHAPTER 2. LOW-RANK TENSOR LEARNING

2.3 Preliminary

A N-dimensional array is called a tensor, which is represented as $A \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_N}$, where $I_n$ is the mode-$n$ dimension ($1 \leq n \leq N$). A metadata of $A$ is presented as $A_{i_1, i_2, \ldots, i_n, \ldots, i_N}$, where $i_n$ is the index of mode-$n$ ($1 \leq i_n \leq I_n$). The mode-$n$ vectors of $A$ are the vectors in $\mathbb{R}^{I_n}$, by keeping the vectors of other modes fixed \[40\].

**Definition 1** (Mode-$n$ unfolding) The mode-$n$ unfolding of $A$ is denoted by matrix $A_{(n)} \in \mathbb{R}^{I_n \times (I_1 \cdot I_2 \cdots I_{n-1} \cdot I_{n+1} \cdots I_N)}$, with the column vectors that are the mode-$n$ vectors of $A$.

**Definition 2** (Core tensor) A tensor $A \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is decomposed by $U_n \in \mathbb{R}^{I_n \times J_n}$ ($1 \leq n \leq N$) as

$$S = A \times_1 U_1 \times_2 U_2 \cdots \times_n U_n \cdots \times_N U_N, \quad (2.1)$$

where $A = S \times_1 U_1^T \times_2 U_2^T \cdots \times_n U_n^T \cdots \times_N U_N^T$, $\times_n$ indicates mode-$n$ product. The transformed tensor $S \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ is called the core tensor. Its mode-$n$ unfolding matrix is represented as $S_{(n)} = (U_N \ldots U_n \ldots U_1 A)_{(n)}$.

**Definition 3** (Tensor Frobenius Norm) The tensor Frobenius Norm (F-norm) can be calculated by

$$\|A\|_F = \sqrt{\sum_{i_1} \cdots \sum_{i_N} A_{i_1 i_2 \ldots i_N}^2}. \quad (2.2)$$

2.4 Low-rank Tensor Completion

Here we introduce the proposed method, along with an indepth look at its formulation and optimization.

Given a set of N-order tensors $\mathcal{X}' = \{\mathcal{X}_i \in \mathbb{R}^{I_1 \ldots I_N} \mid i = 1, \ldots, M\}$, the corresponding labels $\{l_1, \ldots, l_M\}$, and suppose the projection matrices are $U_n \in \mathbb{R}^{I_n \times J_n}$. Then tensors after projection can be calculated as

$$Y = \mathcal{X} \times_1 U_1 \cdots \times_n U_n \cdots \times_N U_N. \quad (2.3)$$

Previous low-rank tensor completion and approximation methods \[41, 39, 42, 43\] are widely used for image denoising and recovering an alignment. The usual way to obtain the intrinsic structure of the tensors is to calculate the trace norm of the N-order tensor as following:
CHAPTER 2. LOW-RANK TENSOR LEARNING

Alternative direction method

\[
\begin{align*}
U_1 & \quad U_2 & \quad U_3
\end{align*}
\]

Figure 2.2: Illustration of 2, 3-dimensional projection matrices respectively. Form left to right: \(U_1\), \(U_2\) and \(U_3\). During learning, each projection matrix is calculated by the alternative direction method.

\[
\min_{X(1), \ldots, X(N)} \sum_{n=1}^{N} \|X(n)\|_* + \lambda \|E(n)\|_1, \tag{2.4}
\]

where \(X(n)\) is the mode-\(n\) unfolding matrix, and \(E(n)\) is the mode-\(n\) error tensor, \(l \in \{*, 1\}\). This means the error item can be calculated by trace norm or sparse learning.

To learn an effective subspace of the tensors for action classification, we alternatively optimize each projection matrices. We denote

\[
X(n) = U_n D(n), \tag{2.5}
\]

where \(X(n)\) is the mode-\(n\) unfolding of tensor datum, \(U_n\) is the projection matrix, \(D(n) = (U_N \ldots U_{n+1} U_{n-1} \ldots U_1 X)_n\). During learning the projection matrices \(U_n\), \(D(n)\) is taken as a constant matrix. Hence, Problem (2.23) can be transformed to minimizing the trace norm of \(U_n\) according to

\[
\min_{U_n} \sum_{n=1}^{N} \|U_n\|_* \tag{2.6}
\]

with some conditions imposed. Meanwhile, the low dimensional structure of \(U_n\) can be automatically captured by the low-rank learning, which is useful for tensorial subspace learning and dimensional reduction.

The matrices \(U_n\) can indicate rotation properties of tensors in the subspace, such as row space, column space. It can also reflect the degree of movement in the frame space. Figure 2.2 shows how the \(U_n\) works. Here \(X \in \mathbb{R}^{I_1 \times I_2 \times I_3}\) is a 3-order tensor with \(1 \leq n \leq 3\). From left to right, it shows the learned projection matrices \(U_1\), \(U_2\) and \(U_3\), which correspond to the transformation in the row, column, and frame space, respectively. Compared with the convectional vector-based method, the matrices can reflect different variances in the row, column of an image, and show the sample
CHAPTER 2. LOW-RANK TENSOR LEARNING

information as well. The first and second rows illustrate the 2, 3-dimensional projection matrices, respectively. $U_1$ and $U_2$ reflect the location of the movement in the row space and the column space of the database. This is different from the vector-based low-rank method [44], which cannot reflect the variance in the row and column of an image. As for $U_3$, each small block stands for a frame reflecting the significance of the frames – if it is a full-rank matrix, each frame plays an important role in the video sequence. Here, the color bar in the second row means different values of $U_n$ $(1 \leq n \leq 3)$. During learning the projection matrices are calculated in an iterative process performed by the alternative direction method.

2.4.1 Discriminant analysis

In order to integrate supervisory information into the low-rank tensor completion model, the discriminant analysis criterion is adopted as a regularizer. For simplicity, let $A = X(n)$. The inter-class and intra-class scatter matrices as follows:

$$B_n = \sum_{i=1}^{C} m_i (\overline{A}_i - \overline{A})(\overline{A}_i - \overline{A})^T,$$

(2.7)

$$W_n = \sum_{i=1}^{C} \sum_{j=1}^{C_i} (A_{ij} - \overline{A}_i)(A_{ij} - \overline{A}_i)^T,$$

(2.8)

where $B_n$, $W_n$ are the mode-$n$ inter-class and intra-class matrices respectively. $\overline{A}_i$, $\overline{A}$ are the mean samples of the $i$-th class and the total number of samples, respectively. $A_{ij}$ is the $j$-th sample of the $i$-th class. $m_i$ denotes the number of $i$-th class.

The corresponding discriminant regularizer is given as

$$\lambda \| U_n^T (W_n - \alpha B_n) U_n \|_F^2,$$

(2.9)

where $\| \cdot \|_F^2$ is the Frobenius Norm [40]. $\alpha$ is the tuning parameter to control the value of the regularization, and $\lambda$ is the parameter to balance the low-rank item and the discriminant item. According to the regularization constraint the low-rank tensor completion model can be expressed as follows:

$$\min_{U_1, \ldots, U_N} \sum_{n=1}^{N} \|U_n\|_s + \lambda \| U_n^T (W_n - \alpha B_n) U_n \|_F^2.$$

(2.10)

With this, the discriminant regularizer not only avoids over-fitting, but also seamlessly integrates the intra-class and inter-class information into the proposed model.
CHAPTER 2. LOW-RANK TENSOR LEARNING

2.4.2 Objective function

Provided an N-order tensor \( X \in \mathbb{R}^{I_1 \times \ldots \times I_N} \). For its real data there always exists some noise or corruption \( E \), which satisfies the following conditions: (1) there is only small fragment or missing part; (2) the location of the error is unknown. Hence, the original tensor can be represented as

\[
X = Y + E,
\]

where \( Y = U_N \ldots U_1 X \) is the low-rank tensor, and \( E \) is the error. We next employed the error item as a constraint, defined by mode-\( n \) unfolding as follows:

\[
\|Y(n) - X(n)\|_F^2 \leq \epsilon,
\]

where \( \epsilon \) is the bias.

Considering the discriminant regularizer and the error item, The low-rank tensor completion model is rewritten as

\[
\min_{U_n} \sum_{n=1}^{N} \|J_n\|_* + \lambda \|U_n^T(W_n - \alpha B_n)U_n\|_F^2 + \beta \|M(n) - X(n)\|_F^2
\]

s.t. \( Y(n) = (U_N \ldots U_1 X)(n), Y(n) = M(n) \),

This model is intractable, because the error item is not convex with respect to the variables. In order to solve this problem, we employ the augmented Lagrange multiplier (ALM) algorithm to optimize Problem (2.13).

2.4.3 Optimization

Due to the difficulty of solving Eq. (2.13), we introduce two auxiliary matrices \( J_n \) and \( M(n) \) to the objection function. The regularization \( \|M(n) - X(n)\|_F^2 \leq \epsilon \) is set as an error term in Eq. (2.13), allowing the objective function to be integrated and rewritten as

\[
\min_{U_1, \ldots, U_N} \sum_{n=1}^{N} \|J_n\|_* + \lambda \|J_n^T(W_n - \alpha B_n)J_n\|_F^2 + \beta \|M(n) - X(n)\|_F^2
\]

s.t. \( U_n = J_n, Y(n) = (U_N \ldots U_1 X)(n), Y(n) = M(n) \),

where \( \beta \) is the parameter of the error item. We use the ALM algorithm to solve the following unconstrained multi-variate optimization problem. The Lagrange function is defined as
CHAPTER 2. LOW-RANK TENSOR LEARNING

\[ L_n = \arg \min_{J_n, U_n, Y_n, M_n, V_1, V_2, V_3} \sum_{n=1}^{N} \| J_n \|_* + \lambda \| J_n^T (W_n - \alpha B_n) J_n \|_F^2 + \beta \| M_n - X_n \|_F^2 + \text{tr} \left[ V_1^T \left( Y_n - (U_N \ldots U_1 X)_n \right) \right] \]

\[ + \text{tr} \left[ V_2^T (U_n - J_n) \right] + \text{tr} \left[ V_3^T (Y_n - M_n) \right] + \frac{\mu}{2} \| (Y_n - (U_N \ldots U_1 X)_n) \|_F^2 + \| U_n - J_n \|_F^2 + \| Y_n - M_n \|_F^2, \]

where \( V_1, V_2, V_3 \) are the Lagrange multipliers, \( \mu > 0 \) is the penalty operator, \( \text{tr}(\cdot) \) is the trace of a matrix. All the variables in the Lagrange function are solvable as following:

\[
\begin{align*}
J_n &= \arg \min_{J_n} \frac{1}{\mu} \sum \| J_n \|_* + \frac{1}{2} \| J_n - (I + \frac{2\lambda}{\mu} (W_n - \alpha B_n))^{-1} (U_n + \frac{V_2}{\mu}) \|_F^2, \\
U_n &= \left( Y_n D_{(n)}^T + J_n + \frac{1}{\mu} (V_1 D_{(n)}^T - V_2) \right) \cdot \left( D_{(n)} D_{(n)}^T + I \right)^{-1}, \\
Y_n &= \frac{1}{2} (M_n + U_n D_{(n)} - \frac{1}{\mu} (V_1 + V_3)) = 0, \\
M_n &= \frac{1}{2\beta + \mu} (2\beta X_n + \mu Y_n + V_3), \\
V_1 &= V_1 + \mu \left( Y_n - (U_N \ldots U_1 X)_n \right), \\
V_2 &= V_2 + \mu (U_n - J_n), \\
V_3 &= V_3 + \mu (Y_n - M_n).
\end{align*}
\]

The convergence conditions are \( \| U_n - J_n \|_F < \varepsilon, \| Y_n - (U_N \ldots U_1 X)_n \|_F < \varepsilon, \) and \( \| Y_n - M_n \|_F < \varepsilon. \) The whole iterative procedure is shown in Algorithm 1.

2.4.4 Improvement

In all actuality, there are many of high-dimensional images with noise or small corruption. Here, we improve the low-rank tensor model in order to complete such images. Motivated by the 2-dimensional image recovery method \( X = AZ + E \), where \( X \) is the original image with noise, \( A \) is the low-rank image and \( E \) is the error, we proposed a 3-dimensional image recovery method by learning the low-rank structure of the sample space \( Z \). Given an image set with \( M \) 3-order samples \( \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times M} \), which is transformed by \( \mathcal{X} = U_3 U_2 U_1 \mathcal{X} Z + \mathcal{E} \), where \( U_1, U_2 \) and \( U_3 \) are the
CHAPTER 2. LOW-RANK TENSOR LEARNING

Algorithm 1 Low-rank tensor discriminant analysis (LRTD)

**INPUT:** M labeled N-order training tensors $\mathbf{\Gamma} = \{\mathbf{X}_i\}$, associated labels $\{l_1, \ldots, l_M\} \in \{1, \ldots, C\}$, the tuning parameter $\alpha$, $\lambda$, $\beta$, and the maximum number of training iterations $t_{\text{max}}$.

**OUTPUT:** Updated $U_n^{(t)}$ ($1 \leq n \leq N$).

1: Initialize $U_n$ by eigen-decomposition of dataset $\mathbf{\Gamma}$. $J_n = 0, V_1 = V_2 = V_3 = 0, \mu = 10^{-6}$, $\mu_{\text{max}} = 10^6, \rho = 1.1,$ and $\epsilon = 10^{-8}$.

2: for $t = 1$ to $t_{\text{max}}$ do

3: for $n = 1$ to $N$ do

4: $\mathbf{X}_i \leftarrow \mathbf{X}_i \times_1 (U_1^{(t-1)})^T \cdots \times_{n-1} (U_{n-1}^{(t-1)})^T \times_{n+1} (U_{n+1}^{(t-1)})^T \cdots \times_N (U_N^{(t-1)})^T$.

5: while $t' < t'_{\text{max}}$ do

6: 1) Update $B_n, W_n$ by Eqs. (2.7) ~ (2.8).

7: 2) Update $J_n, U_n, Y_n, M_n$ and multipliers $V_1, V_2, V_3$ via fixing others in equation set (2.16).

8: 3) Update $\mu$ by $\mu = \min(\rho \mu, \mu_{\text{max}})$.

9: end while

10: $U_n^{(t-1)} = U_n^{(t)}$.

11: end for

12: end for

projection matrices of one sample, $Z$ reflects the low-rank structure of all the samples, and $E$ is the error. The original dataset $\mathbf{X}$ can be reconstructed from the discriminant subspace by low-rank learning of $Z$, therefore, the pure images without noise or illumination interference can be obtained. The model is given as follows:

$$
\min_{U_n} \sum_{n=1}^{3} \|U_n\|_F + \|Z\|_F + \lambda \|U_n^T(W_n - \alpha B_n)U_n\|_F^2 \\
\text{s.t.} \quad \|Y_n - X_nZ\|_F^2 \leq \epsilon, Y_n = (U_3U_2U_1\mathbf{X}Z)_{(n)},$$

(2.17)

where $Z = \left(Y'D' + J' + \frac{V_1D'^T - V_2}{\mu} \left(D'D^T + I\right)^{-1} \right)^{-1}$, $Y'$, $D'$ and $J'$ are the mode-4 variates, which reflects the sample information.
CHAPTER 2. LOW-RANK TENSOR LEARNING

Figure 2.3: Key frames of different actions/gestures of (1) MSR action 3D database and (2) MSR hand gesture database on the first and second rows, respectively.

2.5 Experiment Results

In this part, we use two databases to verify our algorithm and to compare it with other state-of-the-art low-rank tensor representation learning methods used for the action classification (see Figure 2.3).

2.5.1 On the MSR hand gesture 3D database

The MSR hand gesture 3D database \[46\] \[47\] contains 12 classes of hand gestures: letter "Z", "J", "Where", "Store", "Pig", "Past", "Hungary", "Green", "Finish", "Blue", "Bathroom", and "Milk". These are performed by 10 subjects, with each subject performs 2-3 times. There are total of 333 samples, each is an action video consisting of a depth image sequence. We use the same experimental set-up as \[46\] \[47\] in this experiment. All the subjects are independent, and each video sequence is sub-sampled to be the size of \(80 \times 80 \times 18\). The image dimension is sufficient to represent the gesture, and the third dimension is due to the least number of the video sequence.

The optimized low-rank projection matrices of each mode \(U_1, U_2, U_3\) are illustrated in Figure 2.4. The x, y, z-axis indicate the dimension of row, column, and frame, respectively. The color bar represents the value of the matrices. The left column shows our matrices with regular color distribution, specifically, \(U_1, U_2\) indicates the number of variations of the row space and column space, respectively. \(U_3\) indicates the significant frames in the video sequence. It is similar with the full-rank matrix, that is, each frame in the sequence plays an important role in the action. In Zhong and Cheriet’s method \[34\], the matrices do not have the obvious structure in the row, column, and frame space. The differences between Zhong’s method and ours are twofold: (1) Zhong used the k-neighbors to construct local graph, while our method considers the global discriminant information,
and it is sufficient for describing the whole dataset; (2) Zhong used gradient decrease method to update only one variable $W_n = U_n^T \times U_n$ to solve their problem, while we use augmented Lagrange method (ALM) to update all the variables iteratively. In conjunction with this, the matrix $U_n$ has two properties: (1) it is a low-rank structure; (2) contains the structure of the action videos in row, column, and frame space, respectively. The corresponding subspace obtained by the low-rank projection is shown in Figure 2.5. By reference of this, our method portrays the projected gestures of the action video with the details containing more energy (e.g. moving fingers) compared with Zhong and Cheriet’s method. This situation indicates that the projection matrix we obtained contains effective information that ensures a more reliable subspace for the classification task. Table 2.1 shows the accuracy of different methods. It should be evident that, the proposed method performs better than the state-of-the-art low-rank tensor representation learning methods. HON4D+D_{disc} [46] is the latest work on the gesture database using normal orientation histogram. Zhang et al.’s work [32] proposed to rectify align images with distortion and partial missing, which used image sequence after low-rank learning in this experiment. It had lower accuracy than our method, as it relies on the original images and can deal with the trivial changing, such as sparse noise, small fragment, and distortion; while it is not suitable for the large scale of movement, distortion or rotation in the gesture classification task. Zhong and Cheriet’s method is less effective when compared with ours. Figure 2.6(a) shows the accuracy under different parameters $\beta$ and $\lambda$. We can see the proposed method is robust across different parameter settings.
CHAPTER 2. LOW-RANK TENSOR LEARNING

Figure 2.5: Left to right: Zhong and Cheriet’s and our learned projected gestures of the MSR hand gesture database.

2.5.2 On the MSR action 3D database

The MSR action 3D database contains 20 classes of actions. This includes "arm waving", "horizontal waving", "hammer", "hand catching", "punching", "throwing", "drawing x", "drawing circle", "clapping", "two hands waving", "sideboxing", "bending", "forward kicking," "side kicking", "jogging", "tennis swing," "golf swing," "picking up and throwing". Each action is performed by 10 subjects, each performing 2-3 times. There are 567 samples in total. The action video is represented as a high-dimensional tensor in this experiment. In the following, we report two sets of results performed under different experimental settings.

2.5.2.1 Experiment setting 1

Here uses the same conditions as [47,46]. The first 5 subjects are chosen for training, while the rest are for testing. Considering the 0 value pixel as non-informative in the depth image, we first cropped the images using a bounding box to resize each image to $80 \times 80$. Next, we sub-sampled each tensor to $80 \times 80 \times 10$. Figure 2.6(b) shows the accuracy with different value of parameters on the MSR action 3D database. This shows our method outperforms the state-of-the-art low-rank tensor representation learning methods. Table 2.2 shows the accuracy for different parameter settings of $\beta$ and $\lambda$. The time for training of our method is approximately 190 seconds, while it takes 160 seconds in the recognition phase.

2.5.2.2 Experiment setting 2

Here we use the same conditions as Chen et al. [48]. We split the MSR Action 3D database into 3 different sets. In the Test One (Two) set, we take the first (second) action video of each subject for training and the rest for testing. In the Cross Subject set we took the 1, 3, 5, 7, 9 subjects as training using the rest for testing. We performed three different tests on each action set. The results
CHAPTER 2. LOW-RANK TENSOR LEARNING

Figure 2.6: Accuracy of the proposed method under different parameter settings of $\lambda$ and $\beta$ on two used databases.

are shown in Figure 2.7. From top to bottom is Test One, Test Two, and Cross Subject test, each with three training, and testing sets $AS_1$, $AS_2$, $AS_3$ with different parameters. The best result in each test experiment is obtained with the parameter set to 0.01 and scaled as $[0, 1]$. The results compared with the state-of-the-art methods are shown in Table 2.3. In the Test One and Cross Subject sets our method performs best. In the Test Two set, we have an accuracy just 2% lower than Chen et al.’s method. For Zhang et al.’s [32] work, we used entire images in the database, i.e., $10 \times 567 = 5670$ images. Still, it was able to deal with the trivial sparse noise or distortion, such as the digit ‘3’ in their test experiment [32]. However, the action video containing large scale movements in the arms or legs, making it not suitable for this application.

![Accuracy of the proposed method under different parameter settings of $\lambda$ and $\beta$ on two used databases.](image)

(a) MSR Gesture database
(b) MSR Action database

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>HON4D + $D_{disc}$</td>
<td>92.45</td>
</tr>
<tr>
<td>HON4D</td>
<td>87.29</td>
</tr>
<tr>
<td>Zhang et al.</td>
<td>89.93</td>
</tr>
<tr>
<td>Zhong et al.</td>
<td>69.44</td>
</tr>
<tr>
<td>LRTD</td>
<td><strong>99.09</strong></td>
</tr>
</tbody>
</table>

Table 2.1: MSR gesture database.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>HON4D + $D_{disc}$</td>
<td>88.89</td>
</tr>
<tr>
<td>HON4D</td>
<td>85.85</td>
</tr>
<tr>
<td>Zhang et al.</td>
<td>95.96</td>
</tr>
<tr>
<td>Zhong et al.</td>
<td>92.88</td>
</tr>
<tr>
<td>LRTD</td>
<td><strong>98.50</strong></td>
</tr>
</tbody>
</table>

Table 2.2: MSRAction3D database.

2.5.3 On CMU Multi-PIE face database

The CMU Multi-PIE face database [49] includes about 750,000 face images of 337 subjects, involving 15 various views, in 19 changes to illuminations, and 4 expressions. In this experiment, we use 67 subjects with total of 469 samples, half for training and half for testing. The discriminant information was used in this experiment. Here we selected 10 faces from one subject to show our method’s performance when recovering images. The original face set $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times M}$, the corresponding low-rank faces $U_3U_2U_1\mathcal{X}Z$ and the errors $E$ are shown in Figure 2.8 where $M$ is the
CHAPTER 2. LOW-RANK TENSOR LEARNING

![Figure 2.7: Accuracy with different parameters set for the MSR action 3D database. Top to bottom: Test One, Test Two, and Cross Subject Test. Each test contains three sets AS1, AS2, AS3, respectively.](image)

![Figure 2.8: PIE face database. Top row: the original faces; second row: the low-rank faces; third row: the errors.](image)

It shows that the illumination effect is well eliminated by the low-rank learning.

2.6 Conclusion

We proposed a low-rank tensor completion method with discriminant learning for action classification and image recovery. We employed the alternative direction method to calculate each projection matrix, by having the others fixed. In order to integrate the label information of the database, we use the discriminant analysis criterion in the low-rank tensor completion model as a regularizer. To obtain the optimized projection matrices, the augmented Lagrange method was used to solve the multi-variate optimization problem. The property of the projected matrices is explained
Table 2.3: Accuracy (%) of 3 sets on the MSR action database.

<table>
<thead>
<tr>
<th></th>
<th>Chen</th>
<th>Zhang</th>
<th>Zhong</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test One</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS1</td>
<td>97.3</td>
<td>46.67</td>
<td>92.76</td>
<td><strong>99.34</strong></td>
</tr>
<tr>
<td>AS2</td>
<td>96.1</td>
<td>47.71</td>
<td>98.08</td>
<td><strong>99.36</strong></td>
</tr>
<tr>
<td>AS3</td>
<td>98.7</td>
<td>11.33</td>
<td>80.26</td>
<td><strong>99.34</strong></td>
</tr>
<tr>
<td>Average</td>
<td>97.4</td>
<td>35.24</td>
<td>90.37</td>
<td><strong>99.35</strong></td>
</tr>
<tr>
<td><strong>Test Two</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS1</td>
<td>98.6</td>
<td>45.95</td>
<td>77.63</td>
<td><strong>98.68</strong></td>
</tr>
<tr>
<td>AS2</td>
<td><strong>98.7</strong></td>
<td>47.24</td>
<td>91.03</td>
<td>97.44</td>
</tr>
<tr>
<td>AS3</td>
<td><strong>100</strong></td>
<td>10.81</td>
<td>90.79</td>
<td>96.05</td>
</tr>
<tr>
<td>Average</td>
<td><strong>99.1</strong></td>
<td>34.67</td>
<td>86.48</td>
<td>97.39</td>
</tr>
<tr>
<td><strong>Cross Subject</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS1</td>
<td>96.2</td>
<td>44.35</td>
<td>91.67</td>
<td><strong>98.33</strong></td>
</tr>
<tr>
<td>AS2</td>
<td>83.2</td>
<td>46.16</td>
<td>85.83</td>
<td><strong>97.50</strong></td>
</tr>
<tr>
<td>AS3</td>
<td>92.0</td>
<td>10.81</td>
<td>85.83</td>
<td><strong>99.17</strong></td>
</tr>
<tr>
<td>Average</td>
<td>90.5</td>
<td>33.78</td>
<td>87.78</td>
<td><strong>98.33</strong></td>
</tr>
</tbody>
</table>

in detail, *i.e.*, the matrices can reflect the low-rank structure in the row, column and frame space, respectively. In order to recover the high-dimensional images with noise or different illumination, we proposed an improved version that learns the low-rank structure of the sample space, and obtains good performance. Results on the MSR hand gesture 3D database and the MSR action 3D database have shown that our method performs better than the state-of-the-art low-rank tensor representation learning methods. Experiments on the Multi-PIE face database reveals the good recovery results of the faces under different illuminations.

### 2.7 Low-Rank Tensor Analysis for Subspace Learning

In our previous low-rank tensor learning work, we integrate discriminant information in our model for classification in new subspace. As we know, the local information of original dataset recovers the distribution of intra-class and inter-class structures, therefore we explore the local information in our new model in order to improve the performance of classification in the new tensor subspace.
CHAPTER 2. LOW-RANK TENSOR LEARNING

In this section, we first give the overview of our framework, then introduce the preliminary knowledge including tensor formula, low-rank subspace learning and reconstruction matrix. Next, the proposed objective function and solution are displayed, finally the time complexity is analyzed.

2.7.1 Overview of Our Framework

The proposed method aims to solve two main problem we mentioned before. First, most subspace learning methods fix the dimension manually, which may not get higher accuracy of recognition because of the random setting. Second, the local information is usually preserved by calculating inter-class and intra-class neighbors, which is high time-consuming.

To overcome this two problems, our goal is to learn the subspace dimension automatically, to get the proper dimension of the subspace. Meanwhile, the local information is preserved in a graph constraint, which is faster than calculating the neighbors in the training. The details are introduced as follows.

Our framework is shown in Fig. 4.1. In the training phase, the original action videos are taken as input. In order to preserve the spatio-temporal information of the actions, 4D normal vectors are extracted as features (HON4D) from a video, which is represented to be a third-order tensor $X$ to avoid the curse of dimensionality. To learn the tensor subspace, the whole training set is decomposed by HOSVD to obtain low-rank matrices, each of which is performed by nuclear norm and finally is used as projection matrix $U_n$. In the testing phase, we first extract the 4D normal vectors from the test videos, each is represented as a third-order tensor $Y$. Finally, the training and testing sets are projected to a common subspace by $U_n$, then they are matched by a SVM classifier.

Moreover, we construct a graph $\mathcal{L}$ integrated with discriminant information, which guides to learn a reconstruction matrix $Z$ containing low-rank structure and discriminant knowledge. Since $Z$ indicates the number of training samples, it could help to learn $U_n$ with discriminant information in the iterative procedure which is useful for classification.

2.7.2 Preliminary Knowledge

2.7.2.1 Tensor Formula

An $N$-order tensor is a multi-dimensional array with $N$ directions, which is represented as $\mathcal{X}^{I_1 \times I_2 \times \ldots \times I_N}$, where $I_n$ is the dimension of $n$-th direction.
CHAPTER 2. LOW-RANK TENSOR LEARNING

Definition 4 Mode-n unfolding

An N-order tensor $\mathcal{X}$ can be stretched to be a matrix $X_{(n)} \in \mathbb{R}^{I_n \times (I_1 \cdot I_2 \cdot \ldots \cdot I_{n-1} \cdot I_{n+1} \ldots I_N)}$ by fixing one direction, where $\cdot$ is used for scalar product.

All the columns of the unfolding matrix are actually the fibers which along the $n$-th direction of the tensor. And each element of $\mathcal{X}$ is represented as

$$x_{i_1 \ldots i_N} = \sum_{r_1=1}^{I_1} \ldots \sum_{r_N=1}^{I_N} g_{r_1 \ldots r_N} u_1^{(1)} \ldots u_{N}^{(N)},$$

where $i_n = 1, \ldots, I_n, n = 1, \ldots, N$.

Definition 5 Tucker decomposition

The tucker model of a tensor $\mathcal{X}$ is generalized as

$$\mathcal{X} = \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 U_3,$$

where $U_n (n = 1, 2, 3)$ is the projection matrix for dimensional reduction, and $\mathcal{G}$ is called core tensor.

The Tucker decomposition can be rewritten as follows:

$$\arg \min \| \mathcal{X} - \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 U_3 \|^2_F.$$  \hspace{1cm} \text{(2.20)}

Definition 6 $n$-Rank

The $n$-rank of tensor $\mathcal{X}$, which is denoted as $(r_1, \ldots, r_n, \ldots, r_N)$, is obtained from a vector space spanned by mode-$n$ fibers [40].

The $n$-rank can be obtained by the tucker decomposition of tensor, and $r_n = \text{rank}_n(\mathcal{X})$. However, if we compute the rank $(r_1, \ldots, r_n, \ldots, r_N)$ of the tensor, with $r_n < \text{rank}_n(\mathcal{X})$ for several modes, we could use Tucker decomposition to find the proper rank, as left part of Fig. 2.9 shows. The original tensor $\mathcal{X}$ is decomposed to be a core tensor $\mathcal{G}$ and three projection matrix $U_n$. We can see that $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ and $U_n \in \mathbb{R}^{I_n \times r_n}$ in the truncation.

When the rank of the vector space $r_n$ is smaller than rank of $\mathcal{X}$, we can use truncation HOSVD method to find the proper rank. Furthermore, it is useful to find the best fit rank by alternating least squares (ALS) method [50] with truncation.

In the normal case, the tensor is unfolded to be a matrix for execution. Accordingly, the matrix version of Tucker decomposition in Eq. (2.19) is
CHAPTER 2. LOW-RANK TENSOR LEARNING

Truncated tucker decomposition

Low-rank \( Z \approx U_1 U_2 U_3 \)

Core tensor

Projection matrix

Original tensor

Manifold \( \mathcal{M} \)

Tensor neighbors of class 1

Low-rank reconstruction matrix \( Z \) with discriminant information, \( \mathcal{L} \) is a graph.

Low-rank \( Z \)

Tensor neighbors of class 2

Figure 2.9: Schematic of our general model. In the right part, all the tensor samples are distributed in a manifold \( \mathcal{M} \), and we construct a graph \( \mathcal{L} \) with the intra-class neighbors. Low-rank matrix \( Z \) contains local information by the constraint \( \text{tr}(Z^T \mathcal{L} Z) \) because of neighbor information in \( \mathcal{L} \). Besides, \( Z \) also contains the class information because of the intra-class neighbors in \( \mathcal{L} \). The discriminant information is benefit for classification in the learned subspace. The left part aims to find the tensor subspace by low-rank learning of projection matrix \( U_n \), which is calculated by the truncated tucker decomposition. The original tensor \( \mathcal{X} \) is decomposed to be 3 projection matrices \( U_n \) \((n = 1, 2, 3)\) and the core tensor \( \mathcal{G} \). After learning \( U_n \) and \( Z \) alternatively, we can find the tensor subspace for RGB-D action recognition.

\[
X_{(n)} = U^{(n)} G_{(n)} \left( U^{(N)} \otimes \ldots \otimes U^{(n+1)} \right)^T \quad U^{(n-1)} \otimes \ldots \otimes U^{(1)} \]

(2.21)

where \( G_{(n)} \) is the matrix version of core tensor, and \( U^{(n)} \) is the mode-n projection matrix, and \( \otimes \) is the Kronecker product notation. This equation is usually used for mode-n dimensional reduction.

2.7.2.2 Low-Rank Subspace Learning

Here we introduce the proposed method with formulations and its optimization.

Given a set of N-order tensors \( \mathcal{X} = \{ \mathcal{X}_i \in \mathbb{R}^{l_1 \ldots l_N} \mid i = 1, \ldots, M \} \), the corresponding labels \( \{ l_1, \ldots, l_M \} \), and suppose the projection matrices are \( U_n \in \mathbb{R}^{l_n \times J_n} \). Then tensors after projection can be calculated as

\[
Y = \mathcal{X} \times_1 U_1 \ldots \times_n U_n \ldots \times_N U_N.
\]

(2.22)

Previous low-rank tensor completion and approximation methods \([41, 39, 42, 43]\) are widely used for image denoising and recovering an alignment. The usual way to obtain the intrinsic
CHAPTER 2. LOW-RANK TENSOR LEARNING

structure of the tensors is to calculate the trace norm of the N-order tensor as following:

$$\min_{X^{(1)},\ldots,X^{(N)}} \sum_{n=1}^{N} \|X^{(n)}\|_* + \lambda \|E^{(n)}\|_l,$$

(2.23)

where $X^{(n)}$, $E^{(n)}$ are the mode-$n$ unfolding matrix and error respectively, $l \in \{*, 1\}$, which means the error item can be calculated by nuclear norm ($*$) or $L_1$ norm.

To learn an effective subspace of the tensors for action classification, we denote the tensor $X$ by mode-$n$ projection matrices:

$$X^{(n)} = U_n D^{(n)},$$

(2.24)

where $X^{(n)}$ is the mode-$n$ unfolding of tensor datum, $U_n$ is the projection matrix, $D^{(n)} = (U_N \ldots U_{n+1} U_{n-1} \ldots U_1 X)^{(n)}$ is a mode-$n$ unfolding tensor. During learning the projection matrices $U_n$, $D^{(n)}$ is taken as a constant matrix. Hence, Problem (2.23) can be transformed to minimizing the trace norm of $U_n$ according to

$$\min_{U_n} \sum_{n=1}^{N} \|U_n\|_*,$$

(2.25)

with some conditions imposed. Meanwhile, the low dimensional structure of $U_n$ can be automatically captured by the low-rank learning, which is useful for tensorial subspace learning and dimensional reduction. The matrices $U_n$ can indicate rotation properties of tensors in the subspace, such as row space, column space. It can also reflect the degree of movement in the frame space.

Fig. 2.9 shows our schematic general model. The left part shows how to learn the projection matrices $U_n$ by truncated tucker decomposition iteratively, i.e., the original tensor $\mathcal{X}$ is decomposed to be 3 projection matrices $U_n$ ($n = 1, 2, 3$) and the core tensor $\mathcal{G}$ by HOSVD. The right part illustrates how to construct the graph constraint from the manifold $\mathcal{M}$. Particularly, suppose that all the tensor samples are distributed on $\mathcal{M}$, and the graph $\mathcal{L}$ is built with inter-class and intra-class neighbors. In the constraint $\text{tr}(Z^T \mathcal{L} Z)$, $\mathcal{L}$ is used to guide to learn $Z$ with local information integrated with class information by low-rank learning. $Z$ is used to reconstruct $\mathcal{X}$, therefore the decomposed $U_n$ could contain the discriminant knowledge. In our model, both left and right parts are performed together to learn all the variables iteratively. Finally, $U_n$ ($n = 1, 2, 3$) as projection matrices are used for action recognition in the common subspace.

2.7.2.3 Low-Rank Reconstruction Matrix

In the previous low-rank learning method [51] for image recovery, the low-rank sample $X$ is obtained by minimizing the nuclear norm of $\|X\|_*$ with the constraint $\hat{X} = X + E$, where $E$ is the
error. It can get good result by this performance, however, low-rank learning of $X$ does not contain the class information which is beneficial for classification. Considering this, Liu et al. propose a low-rank framework to find the class information by the constraint $X = AZ + E$, where $Z$ can reflect the clustering information from the number of samples $52$. $A$ is the “dictionary” which is the spanned set of data space and $E$ is the error. $Z$ is used to reconstruct $A$, which is also a low-rank matrix calculated by

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_l, \quad \text{s.t.} \quad X = AZ + E,$$

(2.26)

As $\text{rank}(AZ) \leq \text{rank}(Z)$, $AZ$ is taken as a low-rank representation of sample $X$ [52].

In our case of tensor subspace learning, we aim to obtain both the low-rank projection matrix $U_n$ and the reconstruction matrix $Z$, in order to find the subspace while recovering the class information for classification. Considering this, we show our procedure for the subspace learning as follows.

2.7.2.4 Objective Function

Given an image set with $M$ 3-order samples $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times M}$, which is transformed by $\mathcal{X} = U_3 U_2 U_1 \mathcal{X} Z + \mathcal{E}$, where $U_1$, $U_2$ and $U_3$ are the projection matrices of one sample, $Z$ reflects the low-rank structure of all the samples, and $\mathcal{E}$ is the error. The original dataset $\mathcal{X}$ can be reconstructed from the discriminant subspace by low-rank learning of $Z$, therefore, the images without noise or illumination interference can be obtained. The model is given as follows:

$$\min_{U_n} \sum_{n=1}^{3} \|U_n\|_* + \|Z\|_*, \quad \text{s.t.} \quad \|Y - Y \times_4 Z\|_F^2 \leq \epsilon,$$

(2.27)

where $Y = \mathcal{X} \times_1 U_1 \times_2 U_2 \times_3 U_3$, which is projected tensor in the subspace.

Our goal is to learn a tensor subspace for tensor action recognition, via dimensional reduction with each mode of projection matrix $U_n$ ($n = 1, 2, 3$) and reconstruction matrix $Z$. What’s more, we integrate the intra-class and inter-class information in our model for classification. Considering the discriminant information can only reflect the global structure, we intend to find the local information of the dataset to preserve the original structure in the subspace.

In this section, we introduce a graph regularizer to capture the local information. The learned $Z$ can be treated as a new low-rank representation of the projected data $U_n X$. Our goal is to couple the samples of the same class by the similar low-rank coefficients. We define $z_i$ as the $i$-th column of $Z$, which correlates with the $i$-th sample $x_i$ of $X$. According to this property, we introduce a novel regularization term.
CHAPTER 2. LOW-RANK TENSOR LEARNING

\[
\min_{z_i, z_j \in \mathcal{W}} \sum_{i=1}^{I_S} \sum_{j=1}^{I_S} (z_i - z_j)^2 w_{ij}, \forall i, j, w_{ij} \in \mathcal{W},
\]

(2.28)

where \( w_{ij} = 1 \), if \( x_i \) and \( x_j \) have the same label; \( w_{ij} = 0 \) otherwise. From this formulation, we saw that Eq. (3.9) enforces \( z_i \) and \( z_j \) to be similar if \( w_{ij} \) is 1. Mathematically, Eq. (3.9) can be rewritten as \( \text{tr}(Z^T\mathcal{L}Z) \), where \( \mathcal{L} = \mathcal{D} - \mathcal{W} \) and \( \mathcal{D} \) is a diagonal matrix with the rows sum of \( \mathcal{W} \) as the element.

It can be seen that \( \mathcal{L} \) is very similar to graph Laplacian, which has been extensively used in spectral clustering \[53\] and graph embedding \[54\]. However, different from them, the proposed term \( \mathcal{W} \), as it carries discriminative label information between modalities for source data, where RGB and depth data with the same label are coupled.

By adding the regularization term \( \text{tr}(Z^T\mathcal{L}Z) \) to Eq. (2.25), our final objective function is as follows:

\[
\arg\min_{Z, U_n} \|Z\|_* + \sum_{n=1}^{N} \|U_n\|_* + \frac{\eta}{2} \text{tr}(Z^T\mathcal{L}Z),
\]

(2.29)

\[
\text{s.t. } \|\mathcal{Y} - \mathcal{Y} \times_4 Z\|_F \leq \epsilon, \quad n = 1, 2, 3,
\]

where \( \mathcal{Y} = \mathcal{X} \times_1 U_1 \times_2 U_2 \times_3 U_3, U_n \neq 0 \) is employed to avoid trivial solution, and \( \eta \) controls the class information.

In this objective function, the tensor subspace is learned by the low-rank matrix \( U_n \), while reconstruction matrix \( Z \) can find the category distribution of the whole dataset. What’s more, the global discriminant constraint and local constraint integrated can better reflect the structure of the dataset. In a word, our model can automatically find the low-rank subspace with the global and local information of the original dataset.

2.7.3 Optimization

In the tensor subspace, we find a close-form solution for all these variables. Considering multi-variates existing in the model, we intend to use the augmented Lagrange method (ALM) \[52\] to solve the function, which is rewritten as:

\[
\arg\min_{K, Z, J_n, U_n} \|K\|_* + \|J_n\|_* + \frac{\eta}{2} \text{tr}(Z^T\mathcal{L}Z)
\]

\[
+ \beta \|U_n D_n - U_n D_n Z\|_F^2,
\]

(2.30)

\[
\text{s.t. } U_n = J_n, Z = K, \quad n = 1, 2, 3,
\]
CHAPTER 2. LOW-RANK TENSOR LEARNING

where $D_n = \mathcal{X} \times_{n-1} U_{n-1} \times_{n+1} U_{n+1} \cdots \times_N U_N$, and $\beta$ controls the error of dataset. To minimize the rewritten function, the Lagrange function is given as:

$$L_n = \arg \min_{K,Z,J_n,U_n} \|K\|_* + \|J_n\|_* + \frac{\eta}{2} \text{tr}(Z^T L Z) + \frac{\beta}{2} \|U_n D_n - U_n D_n Z\|_F^2$$

$$+ \text{tr} \left[ V_1^T (U_n - J_n) \right] + \text{tr} \left[ V_2^T (Z - K) \right] + \frac{\mu}{2} \left[\|U_n - J_n\|_F^2 + \|Z - K\|_F^2\right],$$

The next work is to calculate the partial derivation of each variate via ALM. $J_n$ and $K$ are calculated by truncated SVD:

$$J_n = \arg \min_{J_n} \frac{1}{\mu} \|J_n\|_F^2 + \frac{1}{2} \||U_n + \frac{V_1}{\mu}\|_F^2,$$  \tag{2.32}

$$K = \arg \min_{K} \frac{1}{\mu} \|K\|_F^2 + \frac{1}{2} \|K - \frac{V_2}{\mu} + Z\|_F^2,$$  \tag{2.33}

For $U_n$:

$$U_n = \left( J_n - \frac{V_1}{\mu} \right) \left( \frac{\mu}{2} D_n \text{Gram}(Z') D_n^T + \frac{V_2}{\mu} \right)^{-1},$$  \tag{2.34}

where $\text{Gram}(Z') = (Z - I)(Z - I)^T$.

For $Z$:

$$\left( \frac{\eta}{2} \mu C + I \right) Z + Z \left( \frac{\beta}{2} U_n D_n D_n^T U_n^T \right) + C = 0.$$  \tag{2.35}

where $C = \frac{V_2}{\mu} - K - \frac{\beta}{2} U_n D_n D_n^T U_n^T$.

The main procedure is detailed in the Algorithm which is different from our previous method and [56].

2.7.4 Discussion

2.7.4.1 Distinct from Previous Work

In [55], we use Fisher discriminant criteria to integrate the global discriminant information in the model. While here we consider that the tensor samples are distributed in a manifold, which need to preserve the local information in the subspace. In [56], we aim to align different datasets by performing nuclear norm on reconstruction matrix $Z$ and latent part $L$, in order to discover the latent information in the target domain. A graph constraint is employed for coupling the RGB and depth modalities in source domain. Different from the previous purpose, here we construct a graph to keep the local structure from the manifold to reduce the time costing for calculating the neighbors.
CHAPTER 2. LOW-RANK TENSOR LEARNING

Algorithm 2 Low-rank tensor subspace learning (LRTS)

INPUT: M labeled N-order training tensors $\Gamma = \{X_i\}$, associated labels $\{l_1, \ldots, l_M\} \in \{1, \ldots, C\}$, the tuning parameter $\alpha$, $\lambda$, $\beta$, $\eta$ and the maximum number of training iterations $t_{\text{max}}$.

OUTPUT: Updated $U_n (t)$ ($1 \leq n \leq N$).

1: Initialize $U_n$ as identity matrix. $J_n = 0$, $V_1 = V_2 = 0$, $\mu = 10^{-6}$, $\mu_{\text{max}} = 10^6$, $\rho = 1.1$, and $\epsilon = 10^{-8}$.

2: for $t = 1$ to $t_{\text{max}}$ do

3: for $n = 1$ to $N$ do

4: $X_i \leftarrow X_i \times_1 (U_1^{(t-1)})^T \times_2 \cdots \times_{n-1} (U_{n-1}^{(t-1)})^T$

5: while $t' < t_{\text{max}}$ do

6: 1) Update $J_n$, $U_n$, $K$, $Z$ and multipliers $V_1$, $V_2$ via fixing others by Eqs. (2.32-2.35).

7: 2) Update $\mu$ by $\mu = \min(\rho \mu, \mu_{\text{max}})$.

8: end while

9: $U_n^{(t-1)} = U_n^{(t)}$.

10: end for

11: end for

2.7.4.2 Time Complexity

In this section, we mainly compare the time-efficiency of Zhong et al. and our method, which are both the tensor-based method. Given a dataset $\{X_1, \ldots, X_S\}$ with $S$ samples, each of which is an $N$-order tensor. The number of neighbors of each sample is set to be $G$.

In Zhong et al.’s method, the main step is to find the intra-class and inter-class neighbors of each sample, meanwhile perform truncated tucker decomposition. Therefore, the time complexity is $O(NSG^2I^3)$ (here $I_n$ is simplified to be $I$ as mode-$n$ dimension). In our method, we set up a graph with class information previously, so we do not have to decompose the tensor with each pair of neighbors, and ours is $O(NI^3)$.

2.8 Experiment

In this section, we first introduce four RGB-D action datasets, parameters setting and compared methods. Then the results are shown and the performance are analyzed.
CHAPTER 2. LOW-RANK TENSOR LEARNING

2.8.1 Datasets

Four RGB-D action databases are employed to evaluate the proposed method in this section: MSRpair3D action dataset, MSRpair skeleton dataset, MSRdaily3D action dataset[^1] and UTKinect-Action dataset[^2].

MSRpair3D dataset contains six pairs of 360 depth action samples, which are performed by 10 subjects with 3 trials each. The first five subjects are used for testing and the rest for training. Fig. 2.10(a) shows the depth frames of different actions of this dataset. MSRpair skeleton data with the same categories is employed also to testify the performance on skeleton modality.

MSRdaily3D dataset contains 16 categories of 320 depth action sequences, which are performed by 10 subjects with twice trials respectively. The odd subjects are used for training and the rest for testing. Fig. 2.10(b) shows the some depth frames of different actions.

In the UTKinect-Action dataset, there are 10 categories of actions: walking, sitting down, standing up, picking up, carrying, throwing, pushing, pulling, waving hands, clapping hands. There are 10 subjects, each performs all the categories twice. So there are total 200 action sequences, each contains about 5 ~ 120 frames. In this experiment, we use 10-fold leave one person out cross-validation (CV) and 2-fold leave one trial out CV for the performances of accuracy.

2.8.2 Parameters Setting

There are three parameters in our model, each of which controls the weight of different factors. The details are listed as follows:

- $\beta$: controls the weight of error of projected samples whether being reconstructed by $Z$.
- $\eta$: controls the weight of local structure of the dataset, which is derived by a pre-defined graph.
- $\alpha$: is a tuning parameter of the discriminant constraint.

In the following experiments, we follow the setting of [46] for comparison.

2.8.3 Comparison Methods

- **HON4D**[^46] Histogram of 4D normal vectors are extracted from an action sequence, which contains both spatial and temporal information. **HON4D + $D_{dist}$**[^46] is the discriminant version of HON4D, which is a modified SVM by adding a support vector.

CHAPTER 2. LOW-RANK TENSOR LEARNING

- **Jiang et al.** [57] Jiang et al. describe the selected discriminant samples from the action sequence via local occupancy pattern (LOP) descriptor.

- **Yang et al.** [11] Yang et al. accumulate the variance of motion between the neighboring frames, which could reflect the intensity of the global activities in a video.


- **Skeleton+LOP** [47] Wang et al. extract the skeleton from the action sequence, then propose a LOP feature for describing the interaction of body parts. **Skeleton+LOP+Pyramid** [47] employs a Fourier temporal pyramid as a low frequency filter, combined with the LOP feature on the skeleton.

- **Zhong et al.** [59] Zhong et al. learn a low-rank subspace automatically by performing nuclear norm on the projection matrix. Meanwhile, they integrate the discriminant information by calculating the intra-class and inter-class neighbors.

- **Jia et al.** [56] Jia et al. propose a low-rank tensor framework for transfer learning. In order to recover the “missing” depth information, they align the source and target domains by performing nuclear norm on the reconstruction matrix \( Z \) and latent information \( L \).

- **LATER** [60] They propose a discriminant tensor-based method, which integrating the Fisher criteria for classification.

- **TLPP** [61] They propose a tensor-based method on the manifold, to find a common subspace for classification. Here the supervised version is employed for comparison.

2.8.4 MSRpair3D Action Dataset

Table 2.4 shows the results of different methods on the MSRpair3D action dataset. For the vector-based methods, we can see that HON4D outperforms the other features because of capturing both spatial and temporal information of an action video. The skeleton + LOP method in [47] only considers the local interaction between two body parts, which may cause some meaningful knowledge missing. While Skeleton+LOP (2) add a pyramid structure compared with the previous model. Yang et al. extract HOG feature from action sequence in front, side and top of view angles respectively. The main issue is that the HOG from different views are combined together directly, so spatial or temporal information may be missing if some views are occluded. For the tensor-based methods, the
proposed method LRTS considers the local information from the manifold. Our previous method LRTD is competitive to LRTS. LTTL \cite{56} here does not learn low-rank projection matrices $U_n$, which causes lower accuracy than LRTS. LTTL also calculates the low-rank reconstruct matrix $Z$ which contains the local structure by employing the neighbor information, but it does not learn the low-rank $U_n$, which is used for align the source and target domain in the model. Meanwhile, there is no global discriminant constraint in the model. Zhong \textit{et al.} learns the dimension of projection matrices automatically either, where the result shows that their framework does not work well on HON4D feature.

Fig. 2.11 shows results of different methods along different dimensions, which indicates $k \times k \times k$ ($k = 1, \ldots, 16$) respectively. Here we turn the 9216 dimension of HON4D (obtained in \cite{46}) to be $24 \times 24 \times 16$, and we only consider the tensor-based methods which have dimensional selection. We can see that the accuracy of LRTS achieves higher along the increasing dimensions, and gets stable from some points. Our previous method LTRD is close to our work in this paper, while LATER/TLPP is also very competitive with ours at the large dimension. While LTTL also gets higher accuracy along the increasing dimension, but not overcomes others. The reason of worse performance of LTTL is mainly because that LTTL performs low-rank learning on $Z$ without learning the dimension of $U_n$ automatically. From this result we can see that our method could get higher accuracy with smaller dimension compared with others. The confusion matrix shown in Fig. 2.12 displays the classification
Table 2.4: Accuracy (%) of MSRPair3D Dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy %</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HON4D</td>
<td>HOG</td>
</tr>
<tr>
<td>Ours (LRTS)</td>
<td>97.78</td>
<td>85.00</td>
</tr>
<tr>
<td>LRTD [55] (AAAI14, Ours)</td>
<td>97.23</td>
<td>86.11</td>
</tr>
<tr>
<td>LTTL [56] (ACMMM14, Ours)</td>
<td>88.89</td>
<td>85.56</td>
</tr>
<tr>
<td>Zhong et al. [59]</td>
<td>88.24</td>
<td>85.56</td>
</tr>
<tr>
<td>TLPP [61]</td>
<td>95.00</td>
<td>86.67</td>
</tr>
<tr>
<td>LATER [60]</td>
<td>97.23</td>
<td>87.22</td>
</tr>
</tbody>
</table>

result of each category. Fig. 2.13 shows the accuracy along different parameters $\beta$ and $\eta$ under a fixed setting of dimension. $\eta$ is set to be $[0, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3]$, while $\beta$ is the same setting without $\{0\}$. We can see that the accuracy under $\eta > 0$ is higher than $\eta = 0$, which indicates that the constraint $\text{tr}(Z^T\mathcal{L}Z)$ plays a trivial role in this dataset.

Fig. 2.14 shows the value of objective function of 3 modes under different number of iterations on the pair dataset. We can see that the value converges to a stationary point within 20 iterations. Fig. 2.15 shows the reconstruction matrix $Z$ under different iterations, we can see that $Z$ contains the local structure by the low-rank learning. Particularly, in the constraint $\text{tr}(Z^T\mathcal{L}Z)$, $\mathcal{L}$ is a kind of Laplace matrix of a graph, which leads $Z$ to contain the local structure of the dataset.

In this paper, we initial the graph as $G$, whose element is 1 if the samples belong to the same class, otherwise 0. Therefore, $Z$ shows the $C$ diagonal blocks, which indicate $C$ categories.

### 2.8.5 MSRPair Skeleton Dataset

Table 2.5 shows the best results of different methods on the MSRPair skeleton dataset. All of these methods use the HON4D feature extracted from the pair skeleton data.

LRTD performs low-rank of $U_n$ which is integrated with discriminant information through Fisher criterion. While our LRTS improves results than LRTD about 2%, because we preserve the local information of the dataset in the subspace. LTTL does not perform low-rank learning on $U_n$, which is crucial to dimensional reduction for classification. Zhong et al. perform competitive with LTTL here.

Fig. 2.16 shows the accuracy of different tensor-based methods under different dimensions.
CHAPTER 2. LOW-RANK TENSOR LEARNING

Figure 2.11: MSRpair3D dataset: accuracy under different dimensions of tensors, where the dimension is $k \times k \times k$ ($k = 1, \ldots, 16$).

Figure 2.12: MSRpair3D dataset: confusion matrix of our method.

The $x$-axis indicates the tensorial dimension $k \times k \times k$ ($k = 1, \ldots, 28$), which is smaller than the dimension of HON4D (the magnitude is $10^4$). As the tensorial dimension is $20 \times 20 \times 28$ from the HON4D dimension 11200 (obtained in [46]), here we set 3D dimension in $[2, 20]$. We can see from
Figure 2.13: MSRpair3D dataset: accuracy under different value of $\beta$ and $\eta$.

Figure 2.14: MSRpair3D dataset: the optimization on 3 modes of training tensors. Each curve of our algorithm LRTS which indicates the value of objective function of one mode converges to a stationary point within 20 iterations.

Figure 2.15: MSRpair3D dataset: reconstruct matrix $Z$ under 5, 7 and 9 iterations. We can see $Z$ contains the local structure, i.e., the class information of the training set.

the figure that our LTRS performs better than LRTD along the increasing 3D dimension, and both of them get higher accuracy than LTTL at the larger dimension. We can see LATER/TLPP performs worse in this dataset, which means they are not suitable for skeleton modality.
CHAPTER 2. LOW-RANK TENSOR LEARNING

Figure 2.16: MSRPair skeleton dataset: accuracy under different dimensions of tensors, where the dimension is $k \times k \times k$ ($k = 1, \ldots, 28$).

2.8.6 MSRDaily3D Action Dataset

Table 2.6 shows the best results of different methods on the MSRDaily3D action dataset. We can see that this work and previous work LRTD performs better than Zhong et al., which are not suitable for the data with complex background. Yang et al. does not perform well in this dataset, where the HOG feature could only be extracted from the front view of an action sequence.

Fig. 2.17 shows the accuracy of our method LRTS and LTTL along different dimensions, which indicates $k \times k \times k$ ($k = 1, \ldots, 50$) respectively. Here we turn each vector of dimension 151200 to be $50 \times 50 \times 60$, and we only illustrate the first 10 3D dimension, from which we can see our performance is better than others even encountering low dimensions. LTTL obtains competitive accuracy with LRTD, and LATER/TLPP gets higher accuracy with increasing dimensions. Fig. 2.18 shows the confusion matrix of different classes, from which we can see that the distinction of read and write actions is not obvious because they are similar to each other, while the same situation occurs between drink and call. Fig. 2.19 shows the reconstruction matrix $Z$ under different iterations, we can see that $Z$ contains the local structure by the low-rank learning.
### Table 2.5: Accuracy (%) of MSRpair Skeleton Dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy %</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tensor-based</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours (LRTS)</td>
<td>81.67</td>
<td>-</td>
</tr>
<tr>
<td>LRTD [55] (AAAI 14, Ours)</td>
<td>79.44</td>
<td>2014</td>
</tr>
<tr>
<td>LTTL [56] (ACM MM 14, Ours)</td>
<td>78.89</td>
<td>2014</td>
</tr>
<tr>
<td>Zhong et al. [59]</td>
<td>76.88</td>
<td>2014</td>
</tr>
<tr>
<td>TLPP [61]</td>
<td>72.23</td>
<td>2005</td>
</tr>
<tr>
<td>LATER [60]</td>
<td>78.34</td>
<td>2005</td>
</tr>
<tr>
<td>Vector-based</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HON4D + $D_{dist}$ [46]</td>
<td>77.78</td>
<td>2014</td>
</tr>
<tr>
<td>HON4D [46]</td>
<td>64.44</td>
<td>2014</td>
</tr>
</tbody>
</table>

### Table 2.6: Accuracy (%) of MSRdaily3D Dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy %</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tensor-based</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours (LRTS)</td>
<td>80.63</td>
<td>-</td>
</tr>
<tr>
<td>LRTD [55] (AAAI 14, Ours)</td>
<td>81.25</td>
<td>2014</td>
</tr>
<tr>
<td>LTTL [56] (ACM MM 14, Ours)</td>
<td>80.63</td>
<td>2014</td>
</tr>
<tr>
<td>Zhong et al. [59]</td>
<td>76.88</td>
<td>2014</td>
</tr>
<tr>
<td>TLPP [61]</td>
<td>80.63</td>
<td>2005</td>
</tr>
<tr>
<td>LATER [60]</td>
<td>80.63</td>
<td>2005</td>
</tr>
<tr>
<td>Vector-based</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HON4D + $D_{dist}$ [46]</td>
<td>80.00</td>
<td>2014</td>
</tr>
<tr>
<td>HON4D [46]</td>
<td>57.50</td>
<td>2014</td>
</tr>
<tr>
<td>Skeleton + LOP (2) [47]</td>
<td>78.00</td>
<td>2013</td>
</tr>
<tr>
<td>Skeleton + LOP [47]</td>
<td>42.50</td>
<td>2013</td>
</tr>
</tbody>
</table>
Figure 2.17: MSRdaily3D dataset: accuracy under different dimensions of tensors.

Figure 2.18: MSRdaily3D dataset: confusion matrix of our method.
CHAPTER 2. LOW-RANK TENSOR LEARNING

Figure 2.19: MSRdaily3D dataset: low-rank $Z$ under different iterations. Left to right: $Z$ under 2, 5 and 8 iterations respectively.

Table 2.7: Accuracy (%) of UTKinect-Action Dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy %</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10-fold</td>
<td>2-fold</td>
</tr>
<tr>
<td>Ours (LRTS)</td>
<td>35.00</td>
<td>48.00</td>
</tr>
<tr>
<td>LRTD [55] (AAAI14, Ours)</td>
<td>28.00</td>
<td>35.50</td>
</tr>
<tr>
<td>LTTL [56] (ACMMM14, Ours)</td>
<td>34.00</td>
<td>38.50</td>
</tr>
<tr>
<td>Zhong et al. [59]</td>
<td>28.00</td>
<td>34.00</td>
</tr>
<tr>
<td>TLPP [61]</td>
<td>25.00</td>
<td>47.50</td>
</tr>
<tr>
<td>LATER [60]</td>
<td>32.50</td>
<td>34.00</td>
</tr>
<tr>
<td>HOJ3D [62]</td>
<td>31.00</td>
<td>-</td>
</tr>
</tbody>
</table>

Vector-based                      | Baseline   | 28.00  | 34.00 | 2014 |

Figure 2.20: Accuracy on Reduced Version of UTKinect-Action Dataset.
CHAPTER 2. LOW-RANK TENSOR LEARNING

Table 2.8: Accuracy (%) of Skeleton-Based Methods on UTKinect-Action Dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy %</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours (LRTS)</td>
<td>92.01</td>
<td>-</td>
</tr>
<tr>
<td>HOJ3D [62]</td>
<td>90.90</td>
<td>2012</td>
</tr>
<tr>
<td>Devanne et al. [63]</td>
<td>91.50</td>
<td>2013</td>
</tr>
<tr>
<td>Yang et al. [64]</td>
<td>92.30</td>
<td>2014</td>
</tr>
<tr>
<td>Jiang et al. [65]</td>
<td>91.90</td>
<td>2015</td>
</tr>
</tbody>
</table>

2.8.7 UTKinect-Action Dataset

2.8.8 Discussion

In this experimental section, we employ 3 common RGB-D action datasets: MSRpair3D action dataset, MSRpai skeleton dataset and MSRdaily3D action dataset, which could testify the performance under different modalities and complex background. In order to better describe an action video, we extract the normal vectors of each pixels. Then we use the histogram of 4D normals (HON4D) as the feature which reflects the tend of body pose of an action sequence. Besides, we tune the different parameters of our model to get higher accuracy which are proper to different datasets. We introduce some state-of-the-art methods for action recognition, including different features and different low-rank tensor framework.

For the MSRpai3D action dataset, we have discussed the accuracy under different dimensions and parameters. We also illustrate the convergence of each mode under different number of iterations. We illustrate the low-rank reconstruction matrix $Z$, which contains local information within several iterations. For the MSRpai skeleton dataset, we compare the accuracy of different methods, show the variation curves of accuracy under different dimensions and illustrate the low-rank $Z$ within several iterations. For the MSRdaily3D action dataset, we show the effect of different dimensions on the accuracy, low-rank $Z$ and confusion matrix. Moreover, we analyze the time complexity of our method with other low-rank tensor method, which indicates ours is more cost-efficient.

In our model, low-rank projection matrix $U_n$ is used to find the tensor subspace, while low-rank reconstruction matrix $Z$ contains local information, which indicates the discriminant information in the training phase. Different from Zhong et al.’s model calculating the inter-class and intra-class neighbors in the iterations, the proposed method integrates local information in a graph.
constraint, therefore it has low time complexity.

2.9 Conclusion

In this paper, we proposed a low-rank tensor subspace learning method for RGB-D action classification. Different from the traditional way to pre-define dimension manually, here the subspace dimension was learned automatically by low-rank learning. Considering the high time complexity for the class information of existing methods, we constructed a graph containing the local class information, which was faster than the previous subspace learning method. In order to describe the RGB-D action data, we extracted the 4D normal vectors from an action video, which could reflect the spatio-temporal information. Then we used the histogram of oriented normal vectors (HON4D) as the feature, which was represented as a tensor to avoid the curse of dimensionality. We employed 3 common RGB-D datasets, which were MSRpairs3D action dataset, MSRpairs skeleton action dataset and MSRdaily3D action dataset, to testify the performance of our method. The results showed that our method obtained higher accuracy than the state-of-the-art approaches based on different features, also higher than the recent tensor based methods for recognition task.
Chapter 3

Low-Rank Tensor in Transfer Learning

3.1 abstract

Kinect sensors are receiving increasing interests since they are cost-effective and can provide both visual and depth modality data at the same time. Unfortunately, depth or RGB modalities are unavailable in training or testing procedures in some realistic scenarios. Therefore, we explore a new problem focusing on arbitrary absence of modality, which is completely different from conventional action recognition. The new problem in action recognition aims to deal with cross modality data (e.g., RGB for training and depth for testing), “missing” modality data (e.g., RGB for training and RGB-D for testing) and single modality data (e.g., RGB/depth in both phases). Accordingly, our method aims to borrow some information from the well-established RGB-D dataset to the existing dataset to help improve the performance, by transferring the correlation between the two modalities. Particularly, we construct a cross-modality regularizer to capture and preserve as much correlation of RGB and depth data as possible. The “missing” knowledge is considered as latent information, which is recovered by low-rank learning in our model. In real world, the target data are usually sparse labeled or completely unlabeled, however, we could exploit the pseudo labels of the target as the prior knowledge to facilitate the target learning. Accordingly, we propose a semi-supervised model for transfer learning. Experiments on three public RGB-D action datasets show that the performance of our method is better than that of the state-of-art transfer learning methods in most cases in terms of accuracy and time efficiency.
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

Correlation ($\rho$) graph of RGB and depth data with class information from source domain

\[
U_nX_S = U_nX_TZ + LU_nX_S + E
\]

Reconstruct matrix

Latent information

Figure 3.1: In our framework, the source domain $X_S$ is the newly generated RGB-D dataset, which contains both RGB and depth action samples. The target domain $X_T$ is the traditional RGB dataset, which only includes the RGB actions. In order to make use of the existing depth information to help recognize the traditional RGB data, we transfer the correlation of two modalities from the source domain to the target domain by aligning them in the formula $U_nX_S = U_nX_TZ + LU_nX_S + E$, to find the latent information $L$ which helps to recover “missing” modality in the target domain. To find the correlation, we construct a graph regularizer $\text{tr}(ZZ^T) + \text{tr}(Z^TGZ)$ containing the class information of the source domain, then transfer the correlation by low-rank learning of the alignment.

3.2 Introduction

Recently action video processing is a fascinating topic in social media field, which receives increasing attention for action recognition \[66\]. There are millions of action videos with tons of clicks on Youtube every year \[67\], especially in public surveillance or entertainment filed. In recent years, RGB-D video data for these applications are used more frequently and have come into the public sight more widely, because it can help boost action recognition performance together with RGB data \[68\]. Depth data contain both geometric structure and depth information, which can be considered as an RGB-D structure and has several advantages: 1) it can eliminate some irrelevant factors such as illumination and occlusion; 2) it can also distinguish similar silhouettes belonging to different actions such as walking and jogging; 3) it is easy to be captured using a low price Kinect. Due to the special properties of RGB-D structure, many action recognition methods \[69\], \[70\] are designed to make better use of the depth data. These methods show that the extra depth information can help improve the accuracy in the recent RGB-D datasets (e.g., MSRDailyActivity3D Dataset \[71\], MSRActionPairs Dataset \[72\] and UTKinect-Action dataset \[62\]).
We summarize three different settings for our new problem in Table 3.1, including: 1) **Single modality**: we can classify the RGB/depth action data when only RGB/depth data are used for training in the target dataset, while RGB-D data in the source dataset. We aim to borrow the additional depth/RGB information existing in a source dataset to help improve the performance of original RGB/depth in target dataset. 2) **Cross modality**: we intend to recognize RGB/depth data when only the other modality depth/RGB data are used for training in the target dataset. We recover some “missing” modality knowledge by transferring the correlation of RGB and depth data from source dataset. 3) **“Missing” modality**: we can classify the RGB-D data when only RGB data exist for training in the target dataset. In a word, if there is any modality is absent, we can still classify the action videos by borrowing the complementary information, utilizing both the geometry and RGB-D structure information from another dataset.

<table>
<thead>
<tr>
<th>Application</th>
<th>Settings</th>
<th>Source domain</th>
<th>Target domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single modality</td>
<td>Setting 1</td>
<td>RGB</td>
<td>RGB-D</td>
</tr>
<tr>
<td></td>
<td>Setting 2</td>
<td>RGB</td>
<td>Depth</td>
</tr>
<tr>
<td>Cross modality</td>
<td>Setting 3</td>
<td>RGB</td>
<td>Depth</td>
</tr>
<tr>
<td></td>
<td>Setting 4</td>
<td>RGB</td>
<td>RGB</td>
</tr>
<tr>
<td>Missing modality</td>
<td>Setting 5</td>
<td>RGB</td>
<td>RGB-D</td>
</tr>
</tbody>
</table>

In this paper, we propose a transfer learning method to recover some information of “missing modality”\(^1\) in the original RGB dataset, by leveraging to the well-established RGB-D dataset, as shown in Fig. 3.1. Take Setting 1 for example, in order to make use of the additional depth information in source domain, we first find the correlation of RGB and depth modalities by a cross-modality graph constraint, then the correlation is transferred to the target domain to help recover “missing” depth knowledge. The graph also contains class information of source domain, which instructs classification in target domain. The “missing” modality is taken account as latent information in target domain, which is recovered by source data and is optimized by low-rank learning. The latent information recovered in target domain can improve the accuracy of the action recognition in target domain.

\(^1\)In this work, the missing depth data means that they are not included in the conventional RGB action databases, or we intentionally remove them from a RGB-D database to build a RGB database for evaluation.
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

There are mainly two problems for RGB-D action recognition. First, although depth data is helpful to improve the performance by providing supplemental information to RGB data, most existing action datasets contain only RGB data without depth information, lead to that most proposed methods \[73, 74\] cannot utilize the additional depth information well. Second, some methods could handle “missing” data problem, but they do not consider different modalities. Bayesian networks can handle the missing data problem \[75\], e.g., the spouse’s name for a single or regular income of a child, etc. The “missing” data could be recovered by the EM algorithm integrated with the searching and scoring method. However, this method usually deals with data of the same modality, so it is not appropriate to recover “missing” depth information directly, as well as regression methods \[76\].

The crucial point of our method is that it integrates RGB and depth modalities from different domains. The sufficient correlation of RGB and depth modalities in the source domain could help to transfer the additional depth information to the target domain. Even though there is no depth information in the target domain, we still could recover some “missing” depth information by transferring the correlation of RGB and depth modalities from source domain to improve the accuracy of RGB action recognition. Moreover, we can compensate the gap between RGB dataset and RGB-D dataset. The traditional action recognition methods are based on either RGB dataset or RGB-D dataset, however, our method finds a way to enrich the RGB dataset as “RGB-D” dataset by recovering additional “depth” information which is transferred from source domain. Additionally, different from the conventional attribute-based methods, our method is data-driven instead of manual modification.

Our previous work \[56\] only considers the label information of source domain, regardless of the unknown labels of target data during training. There are two situations for the labels of target data. First, the categories of target and source data are the same. In this case, the labels of target data could be predicted by the source data. Ding et al. \[77\] proposed a deep transfer framework by predicting the labels of target data and refining them layer by layer to transfer more knowledge from source to target. In particular, they combined all the source and target data together and updated the unknown labels of target data incrementally. The second situation is the categories of target and source data are different. Raina et al. \[78\] proposed self-taught learning to make use of unlabeled target data via sparse coding. However, the label information of target data could be explored and taken advantage when training, by transmitting common properties like semantic information or correlation of two modalities. Rohrbach et al. \[79\] proposed a semi-supervised model by transferring semantic attributes, exploiting the manifold structure of target data to improve predictions of labels in one domain. Inspired by the above work, we propose a semi-supervised framework to predict more
reliable target labels as prior knowledge, to learn an effective common subspace for classification. Particularly, we construct two graphs in source and target domains. One graph is constructed with label information of source domain, while the other is generated by neighbors of unlabeled target data. Both graphs are used for reconstructing source data, i.e., the one indicates target domain is guided by the other with label information, therefore we could obtain more reliable prediction of target labels.

This paper is the extension of our previous work [56]. The extension work contains: first, we explore a new problem that how to take advantage of unlabeled target data to transmit more correlation of RGB and depth modalities to target domain. To solve this problem, we propose a semi-supervised approach by predicting the target labels to reconstruct source domain, therefore we could transfer more correlation for better performance of recognition. Second, we add more experiments with an additional RGB-D dataset UTKinect action dataset as target domain, while the MSRdailyRGB-D action dataset and MSRpairRGB-D action dataset for source domain. Third, we analyze the results affected by several parameters in this supplementary experimental part, e.g., \textit{mode-n} dimension, projection matrices and errors along iterations.

3.2.1 Our Contributions

The significance of our method is that we find a way to borrow some “missing” information from a well-instanced domain to the target domain, and improve the performance of recognition by recovering some latent knowledge. The main contributions of this paper are listed as follows:

- We propose a semi-supervised framework to boost classification in target domain. We predict the target labels and add them in our model incrementally, which reconstructs source domain better and recovers more correlation of RGB and depth modalities in target domain.
- We construct a cross-modality graph to couple RGB and depth modalities and find their correlation, which is transferred to target domain to recover the “missing” depth modality for better classification.
- Tensor representation is used for high-dimensional action videos, preserving their original spatio-temporal structure.

3.3 Related work

There are four parts in this section, first we introduce the methods for action recognition, then we display some transfer learning methods for the recognition task, third semi-supervised
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

transfer learning approach is described and fourth tensor representation of high-dimensional data is explained.

3.3.1 Action Recognition

Action recognition aims to extract the interesting context of people from an action video, and analyze the action for specific purpose, which requests high-quality features learned from the action video. Traditional action recognition methods mainly consider RGB human action, with low-level, mid-level and high-level features. Low-level feature-based methods \cite{80,81} extracted the textual level feature (e.g., spatio-temporal interest points (STIP), bag-of-words \cite{75}, body contour \cite{82}, key poses \cite{83}) for human action representation. Recently, the low-level feature can be further processed using a deep learning technique \cite{84}. Mid-level feature, such as attributes \cite{85}, optical flow \cite{86}, context \cite{87}, HOG/HOF \cite{88}, extended SURF \cite{89}, and relations \cite{90}, is composed of clustered low-level features. High-level feature, which is learned from clustered semantic words \cite{91}, relies on both the words extraction and topology structure of the semantic spaces.

Recently, depth data have caught increasing interest because of the geometry properties and cheap Kinect sensor, therefore, many RGB-D action recognition algorithms have been proposed \cite{92,72} for RGB-D action recognition. There are also some useful features for the depth data, such as depth map \cite{93,94}, RGB-D points \cite{95}, skeleton \cite{96}, HON4D \cite{97}. Compared with the RGB data, the depth data could provide both the geometry and RGB-D structure feature, which can separate the background and foreground according to the hierarchical distance to the camera.

The latest work about RGB-D object recognition is proposed by Chen et al. \cite{98}, they recognized RGB data in the target dataset by transferring the RGB-D knowledge from the source dataset. Particularly, first they aligned the RGB and depth data in the source dataset by canonical correlation analysis, in order to reduce the diversity of the two modalities. Meanwhile, they aligned not only RGB data in both datasets, but also RGB target data and depth source data, which could transfer both the RGB and depth knowledge from the source domain to the target domain. Their main work was to align the existing RGB and depth data in both domain respectively, and only recognized the RGB data in the target dataset by transferring the RGB-D information. They did not recover “missing” depth modality in target domain, therefore there was no cross-modality problem setting.

Compared with the method in \cite{98}, we have extended the application for the test data, which could be either RGB or depth modality. Meanwhile, we define the “missing” modality as the latent information in the target dataset, and we aim to recover some latent information by borrowing the “missing” knowledge from the source dataset, which contains both RGB and depth data. We
calculate the correlation between the RGB data and depth data, which is transferred to the target RGB dataset to recover some depth information. By this procedure, the performance can be improved compared with the situation of only containing the RGB data for training.

Our method is theoretically different from the traditional RGB-D action recognition methods \cite{92,99}. RGB-D action data are existed in both training and testing procedures previously, however, there is some “missing” modality in the training phase in our model, and we intend to borrow some additional information from other datasets. The “missing” information cannot be used directly from the source dataset, but can be transferred to the target dataset to recover some helpful information. Experiments on several datasets show the better performance of our method with the recovered knowledge in term of accuracy.

### 3.3.2 Transfer Learning

Transfer learning techniques have been used increasingly in recent years, with the purpose of transferring the existing valuable knowledge to a new domain with rare information \cite{100}. There are mainly two kinds of transfer learning, the first is inductive learning, which deals with different tasks in the same domain, \textit{i.e.}, multi-task learning. The other is transductive learning \cite{101,102}, which addresses the same task in cross-domain fashion. For transductive learning, as we use in our paper, we have only one group of objects to recognize, while we use the additional information from another well-instituted domain, and finally get the better performance than learning in the single domain.

Nowadays, low-rank learning \cite{103,104} is an effective way for transfer learning in the vision recognition field. Liu et al. \cite{105,52} proposed the low-rank representation (LRR) method, which could mine the structure of the original data from different subspaces. LRR can find the lowest rank of the whole dataset, which is composed of the linear combination of the clean data and error. This low-rank representation not only discovers the global structure of dataset, but also clusters the correlated candidates. What’s more, LRR can remove noise and compensate corruption of the dataset better than other conversion methods which only deal with the Gaussian noise. For a further step, Liu et al. \cite{106} proposed a latent LRR (LatLRR) framework for the partial observed data problem. LatLRR uncovers the latent knowledge of the hidden data by reconstruction of the original dataset.

Considering the above mentioned, the low-rank structure can be integrated in the transfer learning model. LTSL \cite{103} and LRDAP \cite{104} are classical transfer learning approaches with the low-rank constraint. LTSL intends to find a subspace for both the source and target data, then the useful knowledge can be transferred to the target domain for object recognition. LRDAP aims to
rotate the source data to couple with the target data by the low-rank constraint. In their framework, the rotated source data are taken as the training data while the target data as the testing data.

### 3.3.3 Semi-Supervised Transfer Learning

Some transfer learning methods take advantage of unlabeled data by predicting their labels. There are two situations for the target labels, one is the categories of source and target data are the same. Ding et al. [77] proposed a deep transfer learning method in two domains and predicted target labels. They combined the two domains together, which are divided into labeled and unlabeled data, and predicted the labels each time to reconstruct the whole dataset. The second situation is the categories of source and target data are different. Rohrbach et al. [79] proposed a semi-supervised model by transferring semantic attributes, exploiting the manifold structure of target data to improve prediction of unlabeled data. Their method could predict the new categories of data in one domain.

Different from Rohrbach et al.’s [79] work, we aim to transfer common properties between two domains. We align source and target domains by learning a coefficient matrix, which are optimized by two graphs. One graph reflects the manifold structure of labeled source data, while the other is generated by the neighbors of unlabeled target data. Since the two graphs have a cooperation to update the coefficient matrix, the graph with unlabeled target data is guided by the other with label information, therefore we could obtain more reliable predicted labels of the target data.

### 3.3.4 Tensor Representation

Most recently, there are many tensor-based methods for recognition with high-dimensional data representation [107], e.g., face recognition [108], object classification [109], and action video recognition [110]. The tensor representation could preserve the original structure of data, while avoiding the “curse of dimensionality” caused by vectorization [111]. An action video could be presented as a third-order tensor, with the rows and columns composing an image denoted as mode-1 and mode-2 unfoldings, and the temporal information is denoted as mode-3 unfolding of the tensor. This tensor representation also considers many factors which may have influence on the results, such as multi-view and spatio-temporal information in action recognition. Motivated by this situation, we represent the RGB or depth action sequence as a third-order tensor (as a cube), while integrating the temporal information in an action sample. We can also construct an action sample as a fourth-order tensor, while the mode-3 indicates the RGB modality and mode-4 denotes the depth modality. When there is a “missing” modality, the fourth-order tensor is truncated to a third-order tensor.
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

Different from the transfer learning approaches mentioned above, in our previous work [56] we focused on the missing modality problem by transferring the correlation from the RGB-D source data to the RGB/depth target data in the tensor-based framework. By integrating a cross-modality constraint, we can calculate the correlation between the RGB and depth data in the source dataset, and this correlation is transferred to the target dataset for recovering some “missing” modality needed for recognition. Besides this, we added a global graph which contains the class information in the source dataset, which can solidify the correlation between the two modalities of the intra-class. The tensor-based transfer learning approach for high-dimensional RGB-D action recognition is inspired by recovering “missing” modality information in target domain, which achieves better performance in terms of accuracy and time efficiency compared with the state-of-the-art transfer learning methods in most cases.

Different from our previous work [56], we propose a semi-supervised model to take advantage of the unlabeled target data. We construct a graph by neighbors of unlabeled target data, and update the graph with predicted labels each time. This graph is guided by the other graph of labeled source data, therefore we could obtain more reliable predicted target labels to help improve performance. We add a challenge dataset for more experiments and give more analysis for the setting of parameters in this paper.

3.4 Latent Low-rank Tensor Transfer Learning

3.4.1 Preliminary

An N-order tensor is essentially a high-dimensional array defined as $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_n \times \ldots \times I_N}$ with $N$ directions (modes), where each direction has $I_n$ dimensions [40].

Definition 7 Mode-n unfolding

For processing the high-dimensional data, an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_n \times \ldots \times I_N}$ is usually unfolded along mode-n to be a matrix, which is represented as $X_{(n)} \in \mathbb{R}^{I_n \times (I_1 \cdot I_2 \cdot \ldots \cdot I_{n-1} \cdot I_{n+1} \ldots I_N)}$, where $\cdot$ is used for scalar product.

Definition 8 Tucker decomposition

Given an N-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ and matrices $U_n \in \mathbb{R}^{I_n \times J_n} \ (1 \leq n \leq N)$, the tucker decomposition is performed by

$$S = \mathcal{X} \times_1 U_1 \times_2 U_2 \ldots \times_n U_n \ldots \times_N U_N,$$

(3.1)
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

Figure 3.2: Illustration of tensor mode-n unfolding. An action sequence is represented as a RGB-D tensor $\mathcal{X}$, and we get mode-n unfolding matrix $X^{(n)}$ along its $x, y, z$-axis.

$$\mathcal{X} = S \times_1 U_1^T \times_2 U_2^T \ldots \times_N U_N^T$$

where $S \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_N}$ is called the core tensor.

3.4.2 Motivation

Since many existing action datasets are generated long time ago, there is only RGB modality of data in them. Although many action recognition approaches based on the RGB action data have achieved impressive performance, with the various extracted features, such as STIP, bag-of-words, HOG, SIFT, there still exists some shortages of this action representation. For example, the swimming action is similar with the waving action only considering the contour, and the feature as mentioned above may not well describe the two different actions. However, this problem can be alleviated by the depth information, which contains the distance between the limbs and the camera. Since the RGB-D action could distinguish the actions better, recently there are many corresponding action recognition approaches dealing with the RGB and depth knowledge.

Considering many of the existing action datasets only contain the RGB modality, our goal is using the RGB-D action generated recently to help recognize the traditional RGB action. The main work for us is to find a way to explore the correlation of RGB and depth modalities, and transfer the correlation to target domain to recover “missing” depth information, which helps improve the performance of recognizing RGB actions.
In the real world, the data are rarely labeled or completely unlabeled, which motivates us to generate some pseudo labels to ease the difficulty of recognition. We propose a semi-supervised approach to generate the labels of target data, which are used to capture the manifold structure of target data as well as source data, to better align the two domains and transfer more correlation of RGB and depth modalities for improvement of target recognition.

3.4.3 Latent Problem Formulation

Given source and target datasets \( X_S \in \mathbb{R}^{(I_1 \times I_2 \times I_3) \times I_S} \) and \( X_T \in \mathbb{R}^{(I_1 \times I_2 \times I_3) \times I_T} \), each of which is represented as a fourth-order tensor, where \( I_1, I_2 \) and \( I_3 \) indicate the dimensions of row, column and number of frames of one sample, and \( I_S \) and \( I_T \) mean the number of source and target samples. Both datasets have RGB and depth modalities, which are represented as:

\[
X_S = [X_S^{\cdot \text{RGB}}, X_S^{\cdot \text{D}}] \quad \text{and} \quad X_T = [X_T^{\cdot \text{RGB}}, X_T^{\cdot \text{D}}].
\]

Conventional transfer methods take one dataset with two modalities into account, e.g., \( X_S^{\cdot \text{RGB}} \rightarrow X_S^{\cdot \text{D}} \) and \( X_T^{\cdot \text{RGB}} \rightarrow X_T^{\cdot \text{D}} \), or one modality distributed in two datasets, e.g., \( X_S^{\cdot \text{RGB}} \rightarrow X_T^{\cdot \text{RGB}} \) and \( X_S^{\cdot \text{D}} \rightarrow X_T^{\cdot \text{D}} \). However, either transfer processing would fail if one modality in target dataset is lost. How can we uncover the “missing” knowledge in the transfer processing? In this paper, we propose to transfer the correlation of RGB and depth modalities by coupling them in a graph regularizer to the target domain to recover the “missing” information. We suppose that source dataset \( X_S \) contains two modalities, while target \( X_T \) only contains one.

Suppose \( X_S \) and \( X_T \) are distributed in various spaces, and \( X_S \subseteq X_T \). To transfer the latent knowledge, we suppose that they are distributed in the same subspace by a projection matrix \( U_n \in \mathbb{R}^{I_n \times J_n} \) (\( J_n \leq I_n \), and \( n = 1, 2, 3 \)), i.e., \( U_n X_S \subseteq U_n X_T \). As the previous transfer methods show, low-rank learning helps find the local structure of the source data. Inspired by this, we transfer the information in the low-rank tensor representation as

\[
\begin{align*}
\min_Z \|Z\|_* , \\
\text{s.t. } U_n X_S^{(4)} &= U_n X_T^{(4)} Z, \\
U_n^T U_n &= I, \quad n = 1, 2, 3,
\end{align*}
\]

where \( \| \cdot \|_* \) is the nuclear norm, \( X^{(4)} \) is mode-4 unfolding, while each sample is a vector here. \( Z \) is “low-rank representations” of \( X_S^{(4)} \) with respect to \( X_T^{(4)} \). Since \( X_S \) and \( X_T \) are well-aligned by \( U_n \) (\( n = 1, 2, 3 \)), here we focus on the mode-4 and omit the superscript “(4)” for simplicity.

We propose to find the relationship between \( X_S \) and \( X_T \) via Single Value Decomposition (SVD), i.e., \( U_n [X_S, X_T] = H \Sigma V^T \), where \( V = [V_S; V_T] \) by row partition. Therefore, \( U_n [X_S, X_T] \)
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

\[ H \Sigma[V_S; V_T]^T = [H \Sigma V_S^T; H \Sigma V_T^T], \]

and \( U_n X_S = H \Sigma V_S^T, U_n X_T = H \Sigma V_T^T \) can be derived. The above constraint can be written as

\[ H \Sigma V_S^T = H \Sigma V_T^T Z. \]

(3.3)

Therefore, Eq. (3.2) can be transformed into

\[ \min_Z \| Z \|_*, \]

s.t. \( V_S^T = V_T^T Z. \)

(3.4)

According to Theorem 3.1 \[106\], the optimal low-rank representation \( Z_\ast \) can be found as follows:

\[ Z_\ast = V_T V_S^T \]

(3.5)

\[ = [V_T \cdot \text{RGB}; V_T \cdot \text{D}] V_S^T, \]

where \( V_T \) has been row partitioned into \( V_T \cdot \text{RGB} \) and \( V_T \cdot \text{D} \). Then the constrained part can be rewritten as

\[ U_n X_S = U_n X_T Z_\ast \]

\[ = U_n[X_T \cdot \text{RGB}, X_T \cdot \text{D}] Z_\ast \]

\[ = U_n[X_T \cdot \text{RGB}, X_T \cdot \text{D}][V_T \cdot \text{RGB}; V_T \cdot \text{D}] V_S^T \]

\[ = U_n X_T \cdot \text{RGB} V_T \cdot \text{RGB} V_S^T + U_n X_T \cdot \text{D} V_T \cdot \text{D} V_S^T \]

\[ = U_n X_T \cdot \text{RGB} (V_T \cdot \text{RGB} V_S^T) + U_n \Sigma V_T \cdot \text{D} V_T \cdot \text{D} V_S^T \]

\[ = U_n X_T \cdot \text{RGB} \tilde{Z} + (U_n \Sigma V_T \cdot \text{D} V_T \cdot \text{D} \Sigma^{-1} U_n^T) U_n X_S \]

\[ = U_n X_T \cdot \text{RGB} \tilde{Z} + \tilde{L} U_n X_S, \]

where \( \tilde{Z} \) indicates the low-rank structure of source data on the target data only with RGB. We consider that target data and latent information are from the same low-rank subspace, and \( \tilde{L} = U_n \Sigma V_T \cdot \text{D} V_T \cdot \text{D} \Sigma^{-1} U_n^T \) should also be low-rank, as \( V_T \cdot \text{D} V_T \cdot \text{D} \) aims to uncover the block structure of \( U_n X_T \cdot \text{D} \).

Accordingly, when \( X_T \cdot \text{D} \) is unobserved (that is, \( X_T = X_T \cdot \text{RGB} \)), the objective function could be written as

\[ \min_{\tilde{Z}, \tilde{L}} \| \tilde{Z} \|_* + \| \tilde{L} \|_*, \]

s.t. \( U_n X_S = U_n X_T \tilde{Z} + \tilde{L} U_n X_S, \)

(3.7)

\[ U_n^T U_n = I, \ n = 1, 2, 3. \]

With Eq. (3.7), we can handle the problem where part of the target data is missing by referencing the recovered data from the source domain. For simplicity, we remove tildes from \( \tilde{Z} \) and \( \tilde{L} \).
In reality, the data are sometimes corrupted. So we need to add an error term as done previously in [105, 103, 52], then the general model in Eq. (3.7) is represented as

\[
\min_{Z, L, E} \|Z\|_* + \|L\|_* + \lambda \|E\|_{2,1},
\]

s.t. \( U_nX_S = U_nX_T Z + LU_nX_S + E, \) (3.8)

where \( \lambda > 0 \) is the parameter to balance the error, and \( \|E\|_{2,1} = \sum_i \sum_j (\sqrt{(E_{ij})^2}) \) is \( L_{2,1} \) norm, which can make \( E \) sample specific (column sparsity), resulting in the formula detecting some outliers.

As shown in [106], LatLRR reconstructs the latent data from both dimensions and number of samples. Our method also performs the same way. \( X_S \) is reconstructed by \( X_T \) in four modes, i.e., the first three dimensional modes and the fourth number mode by \( U_n \) and \( Z \).

In a word, if there are “missing” data in the target domain, \( U_nX_S \) can be used for reconstruction to recover the latent information. Different from [106], the “missing” modality can be used for testing, and more latent information would be uncovered by the coupling of two modalities in the source domain.

### 3.4.4 Cross-Modality Regularizer

Our model could uncover the “missing” modality in target domain by transferring the correlation of RGB and depth modalities from source domain. In this section, we use a cross-modality regularizer to capture the correlation. The coefficient matrix \( Z \) is a low-rank representation of \( X_S \), which is composed of RGB modality \( Z_{RGB} \) and depth modality \( Z_D \). We aim to couple the two modalities of the same class by \( Z \). We define the \( i \)-th column of \( Z \) as \( Z_i \) to be the \( i \)-th sample \( x_i \) of \( X_S \). Based on this definition, we construct a graph \( \mathcal{W} \) to couple the two modalities:

\[
\min_{z_i, z_j} \sum_{i=1}^{I_5} \sum_{j=1}^{I_5} (z_i - z_j)^2 w_{ij},
\]

\( \forall i, j, w_{ij} \in \mathcal{W}, \) (3.9)

where \( w_{ij} = 1 \), if \( x_i \) and \( x_j \) belong to the same class; \( w_{ij} = 0 \) otherwise. From this regularizer, we notice that Eq. (3.9) enforces \( z_i \) and \( z_j \) to be similar if \( w_{ij} \) is 1. Mathematically, Eq. (3.9) can be rewritten as \( \text{tr}(Z^T L Z) \), where \( L = D - \mathcal{W} \) and \( D \) is a diagonal matrix whose element is the rows sum of \( \mathcal{W} \). \( L \) is similar to graph Laplacian, which is extensively used in spectral clustering [53] and graph embedding [54]. However, different from them, the proposed term \( \mathcal{W} \) carries discriminative information, which is helpful to couple the RGB and depth modalities in the same class. Our final objective function is rewritten by the regularizer as
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

\[
\begin{align*}
\min_{Z,L,E} & \|Z\|_* + \|L\|_* + \lambda \|E\|_{2,1} + \frac{\beta}{2} \text{tr}(ZLZ^T) \\
\text{s.t.} & \quad U_nX_S = U_nX_TZ + LU_nX_S + E, \quad n = 1, 2, 3,
\end{align*}
\] (3.10)

where \(\beta \geq 0\) is the balanced parameter. When \(\beta = 0\) there are no couples among the same class.

The RGB and depth modalities could be well coupled by the above function, and the correlation in \(Z\) is transferred to the target domain by low-rank learning. Therefore, the “missing” information can be uncovered via the correlation. In the experiment, we show that the cross-modality regularizer could help to transfer the correlation to uncover the latent information of the target data.

3.4.5 Optimization

To solve Eq. (3.10), we rewrite the objective function by introducing some auxiliary matrices as

\[
\begin{align*}
\min_{K,Z,L,E} & \|K\|_* + \|W\|_* + \frac{\beta}{2} \text{tr}(ZLZ^T) + \lambda \|E\|_{2,1} \\
\text{s.t.} & \quad U_nX_S = U_nX_TZ + LU_nX_S + E, \\
& \quad Z = K, \quad L = W, \quad n = 1, 2, 3.
\end{align*}
\] (3.11)

Augmented Lagrangian Multiplier \[112,113\] is applied to achieve better convergence of the Eq. (3.11), the augmented Lagrangian function is:

\[
\begin{align*}
\min_{K, W, Z, L, U_n, E} & \|K\|_* + \|W\|_* + \lambda \|E\|_{2,1} \\
& + \text{tr}[Y_1^T(U_nX_S - U_nX_TZ - LU_nX_S - E)] \\
& + \text{tr}[Y_2^T(Z - K)] + \text{tr}[Y_3^T(L - W)] \\
& + \frac{\beta}{2} \text{tr}(ZLZ^T) + \frac{\mu}{2} \left[\|Z - K\|_F^2 + \|L - W\|_F^2\right] \\
& + \|U_nX_S - U_nX_TZ - LU_nX_S - E\|_F^2
\end{align*}
\] (3.12)

where \(Y_1, Y_2, Y_3\) are Lagrange multipliers and \(\mu > 0\) is a penalty parameter. All the variables in the Eq. (3.12) are optimized one-by-one iteratively. Augmented Lagrangian Multiplier \[112,113\] is employed for the above problem, which converges well even when some of the data is not smooth.

In details, we update the variables \(Z, L, E, K, W\) and \(U_n\) iteratively. For \(K\) and \(W\) we use nuclear norm and Lagrange as:

\[
K_{t+1} = \arg\min_K \frac{1}{\mu} \|K\|_* + \frac{1}{2} \|K - (Z_t + Y_2/\mu)\|_F^2.
\] (3.13)
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

\[ W_{t+1} = \arg \min_W \frac{1}{\mu} \|W\|_* + \frac{1}{2} \|W - (L_t + Y_3/\mu)\|_F^2. \]  

(3.14)

For Z:

\[ \beta Z_{t+1}L + \mu(I - X_TX_T^T)Z_{t+1} + \mu M_t = 0, \]  

(3.15)

where \( M_t = \frac{Y_2 - Y_1X_f^T}{\mu} - K_t - (X_S - L_tX_S - E_t)X_T^T. \)

For L:

\[ L_{t+1} = \left( (U_{n,t}X_S - U_{n,t}X_TZ_t - E_t)(X_S^TU_{n,t}^T) + W_t + \frac{Y_1X_S^TU_{n,t}^T - Y_3}{\mu} \right)(U_{n,t}X_SX_S^TU_{n,t}^T + I)^{-1}. \]  

(3.16)

For \( U_n, (n = 1, 2, 3): \)

\[ U_{n,t+1} = \left( -\frac{Y_1}{\mu} + E_t \right)(X_S^{(n)} - X_T^{(n)}Z_t - L_tX_S^{(n)})^{-1}, \]  

(3.17)

where \( X_S^{(n)} \) and \( X_T^{(n)} \) mean \( X_S \) and \( X_S \) unfolded in mode \( n, n = 1, 2, 3. \)

For E:

\[ E_{t+1} = \arg \min_E \frac{\lambda}{\mu} \|E\|_{2,1} + \frac{1}{2} \|E - \tilde{E}_t\|_F^2, \]  

(3.18)

where \( \tilde{E}_t = U_{n,t}X_S - U_{n,t}X_TZ_t - L_tU_{n,t}X_S + Y_1/\mu. \)

Amongst them, Eqs. (3.13)(3.14) can be solved by Singular Value Thresholding (SVT) [114], also it can be solved by fixed-rank representation [115] for faster speed and lower computational cost. Eq. (3.18) is solved by the shrinkage operator [116]. Eq. (3.15) is updated by a Lyap [117].

3.4.6 Semi-Supervised Low-Rank Transfer Learning

The previous work takes advantage of source label information only, regardless of the label of target information. However, the target label is partially known sometimes in realistic circumstances, which could be considered as an extra knowledge to help to reconstruct the source domain. We design a semi-supervised framework for low-rank transfer learning (SLTTL), where labels of target samples are learned incrementally, and being used to reconstruct source domain, which could better align the source and target domain.

The previous objective function described in Eq. (3.10) contains a constraint term \( \text{tr}(ZLZ^T) \), where source graph \( L \in \mathbb{R}^{I_S \times I_S} \) guides to learn coefficient matrix \( Z \in \mathbb{R}^{I_T \times I_S} \). In our new model, we construct a target graph \( G \in \mathbb{R}^{I_T \times I_T} \), which to guides to learn \( Z^T \in \mathbb{R}^{I_S \times I_T} \) at the same time. We
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

Algorithm 1 SLTTL (Solving Problem Eq. (3.19))

Input: $X_T, X_S, \lambda, \beta, \mathcal{L}$

Initialize: $Z = K = 0, W = L = 0, E = 0, Y_1 = 0, Y_2 = 0, Y_3 = 0, \mu_0 = 10^{-6}, \rho = 1.2, \max_{\mu} = 10^{6}, \epsilon = 10^{-6}$.

while not converged do
1. Fix the others and update $K_{t+1}$ by Eq. (3.13).
2. Fix the others and update $L_{t+1}$ by Eq. (3.16).
3. Fix the others and update $Z_{t+1}$ by Eq. (3.20).
4. Fix the others and update $W_{t+1}$ by Eq. (3.14).
5. Fix the others and update $U_{n,t+1}$ by Eq. (3.17) and $U_{n,t+1} \leftarrow \text{orthogonal}(U_{n,t+1})$.
6. Update the multipliers $Y_1, Y_2, Y_3$ by
   
   $Y_1 = Y_1 + \mu(U_nX_S - U_nX_T Z - LU_nX_S - E)$;
   
   $Y_2 = Y_2 + \mu(Z - K)$;
   
   $Y_3 = Y_3 + \mu(L - W)$.
7. Update the parameter $\mu$ by
   
   $\mu = \min(\rho\mu, \max_{\mu})$
8. Check the convergence conditions
   
   $\|U_{n,t+1}X_S - U_{n,t+1}X_T Z_{t+1} - L_{t+1}U_{n,t+1}X_S - E_{t+1}\|_{\infty} < \epsilon,$
   
   $\|W_{t+1} - L_{t+1}\|_{\infty} < \epsilon,$
   
   $\|Z_{t+1} - K_{t+1}\|_{\infty} < \epsilon.$
end while

output: $Z, W, L, K, E, U_n$.

learn labels of some target samples each time, and update $\mathcal{G}$ accordingly. Particularly, the values indicate the same class in $\mathcal{G}$ are set to be 1, while other values corresponding to samples without labels are set to be 0. The new objective function is designed as follows:

$$\min_{Z,L,E} \|Z\|_* + \|L\|_* + \lambda\|E\|_{2,1} + \frac{\beta_1}{2} \text{tr}(Z\mathcal{L}Z^T) + \frac{\beta_2}{2} \text{tr}(Z^T\mathcal{G}Z)$$

(3.19)

s.t. $U_nX_S = U_nX_T Z + LU_nX_S + E, \ n = 1, 2, 3,$

Augmented Lagrangian Multiplier [112] is applied to achieve all the variables. For $Z$:

$$\beta_1 Z_{t+1} \mathcal{L} + (\beta_2 \mathcal{G} - \mu X_T U_n^T U_nX_T + \mu I)Z_{t+1} + \mu M_t = 0,$$

(3.20)

where $M_t = \frac{Y_2 - X_T U_n^TY_1}{\rho} - K_{t+1} - X_T^TU_n^T(U_n X_S - L_{t+1}U_n X_S - E_t)$. For the other variables, the solutions are the same with the previous model. The details of the algorithm are outlined in Algorithm 1. The
parameters $\mu_0, \rho, \epsilon$ and $\max_\mu$ are set empirically, while other balance parameters are tuned for the experiment.

### 3.4.7 Complexity Analysis

We assume the unfolded datasets $X_S$ and $X_T \in \mathbb{R}^{m \times n}$ for simplicity. Since the low-rank transfer processing costs most in our method, the time complexity of Algorithm 1 concludes:

- Trace norm computation of an $n \times n$ matrix (Step 1, 4).
- Matrix multiplication and inverse (Step 2 and 5).
- Lyap equation (Step 3).

Here, we discuss the computation complexity in detail. First, the trace norm solved by SVD computation in Step 1 and Step 4 takes $O(n^3)$. When $n$ is very large, this step would cost a lot. Second, either the general multiplication or the inverse costs $O(n^3)$. Finally, Step 3 is Lyap function, which takes $O(m^3)$, and is related with the dimensionality of the data. In sum, the time complexity is $O(4n^3 + m^3)$.

### 3.4.8 Discussion

Our method could borrow some latent information from an auxiliary dataset to an existing dataset, by transferring correlation of two modalities. Some crucial properties compared with previous methods are discussed as below:

**Uncover missing depth information in the target data.** We suppose RGB-D source dataset could be aligned with RGB target dataset plus latent depth information, by low-rank constraint which reconstructs source domain and transfers the correlation to target domain. This allows us to uncover “missing” depth knowledge which is helpful to improve the performance of recognition.

**Coupling two modalities.** A cross-modality regularizer of RGB-D source data is employed to couple the two modalities, then their correlation is transferred to the target domain to help recover the “missing” knowledge.

**Capturing structure information.** We propose a tensor-based representation for RGB and depth data, to preserve the original spatio-temporal information of an action video.

The traditional transfer learning methods, like GFK, LTSL and $L^2$TSL [118], do not consider “missing” information, what they do is transferring information between modalities within one domain, or transferring between domains with single modality. Different from them, our method is based on tensor representation of data, which preserves the original spatio-temporal structure of
3.5 Experiments

In this section, we first introduce three common used action datasets and different settings, which focus on single, cross and missing modalities in Table 3.1. Then we analyze the experiment results and some properties, such as convergence, time complexity. Finally a short discussion is given.

3.5.1 Datasets

In the experimental section, we use three action datasets to evaluate the proposed method: MSRDailyActivity3D Dataset, MSRActionPairs Dataset\(^2\) and UTKinect-Action Dataset\(^3\). All have

---

\(^2\) http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc/
\(^3\) http://cvc.ece.utexas.edu/KinectDatasets/HOJRGB-D.html
two modalities, RGB images and depth images.

In the MSRDailyActivity3D Dataset, there are 16 categories of actions performed by 10 subjects, each performs every action twice. There are 320 RGB samples and 320 depth samples. Fig. 3.3(a) and 3.5(a) show the RGB and depth sequence of one sitting action.

In the MSRActionPairs Dataset, there are six pairs of actions performed by 10 people, with three trials each. There are a total of 360 RGB samples and 360 depth action samples. There are ten subjects performing three trials for each action, and the first five subjects for testing. Fig. 3.3(b) shows the RGB sequence of pushing a chair action.

In the UTKinect-Action Dataset, there are 10 categories of actions: walking, sitting down, standing up, picking up, carrying, throwing, pushing, pulling, waving hands, clapping hands. Fig. 3.4 shows the RGB and depth data respectively. There are 10 subjects, each performs all the categories twice. So there are total 200 action sequences, each contains about 5 ~ 120 frames. In this experiment, we only take the actions performed by the first trail of subject #1 as the auxiliary dataset, which contains 306 RGB and depth data respectively.

To unify the tensor size, in MSR daily and pair datasets we extract 10 frames from each video sequence at specific intervals. While in UTKinect-Action Dataset, each frame is extended to be a tensor with 10 frames by Gabor filters. We use the whole frame to extract the feature instead of tracking, because (1) the background of the video are static, (2) we apply HOG descriptors to extract the feature. Thus, the background will not contribute to the feature too much and human tracking is not necessary here.

In both datasets, RGB and depth action samples are sub-sampled to be $80 \times 80 \times 10$, whose HOG feature are extracted to represent the action samples. The HOG feature of source RGB data is shown in Fig. 3.5(b).

### 3.5.2 Experimental Settings

#### 3.5.2.1 Training and testing cases

We have three different applications according to modalities, i.e., single modality, cross modality and missing modality cases in five settings. In each setting, there are various training and testing samples respectively, which is shown in Table 3.1. By these settings, we demonstrate that our method can utilize the additional information by transferring the correlation of RGB and depth data.

---

4 http://www.vlfeat.org/overview/hog.html
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

Figure 3.4: UTKinect-Action dataset, (a) the RGB walking sequence and (b) the corresponding depth sequence.

from source domain, therefore our model has general application with various “missing” modality cases.

3.5.2.2 MSRAActionPairs Dataset

In this experiment, the testing data are from MSRAActionPairs dataset.

**Source dataset:** MSRDailyActivity3D contains 16 categories with 640 samples including 320 RGB/depth samples.

**Target dataset:** MSRAActionPairs Dataset contains 12 categories with 720 samples including 360 RGB/depth samples.

The target dataset is separated into two parts for training and testing, respectively. Therefore, we can test each part reversely by alternating orders listed in Case 1, 2 situations.

**Case 1:** Training set: actions performed by first five people; testing set: rest of actions by last five people.

**Case 2:** The training and testing sets are reversed.

3.5.2.3 MSRDailyActivity3D Dataset

In this experiment, the testing data are from MSRDailyActivity3D dataset.

**Source dataset:** MSRAActionPairs Dataset.

**Target dataset:** MSRDailyActivity3D Dataset.

The target dataset is separated into two parts for training and testing, which is listed as below.
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

Case 3: Training set: actions performed by first five people; testing set: rest of actions by last five people.
Case 4: The training and testing sets are reversed.

3.5.2.4 UTKinect-Action Dataset VS. MSRDailyActivity3D Dataset

In this experiment, the testing data are from UTKinect-Action dataset, and source data from MSRDailyActivity3D dataset.

Source dataset: MSRDailyActivity3D Dataset.
Target dataset: UTKinect-Action Dataset contains 612 samples of 10 categories with 360 RGB/depth samples. The target dataset is separated into two parts, which is listed as below.
Case 5: Training set: actions from the first five people; testing set: the rest of actions from the last five people.
Case 6: The training and testing sets are reversed.

By this case, we can see the accuracy of RGB and depth data in the UTKinect-Action Dataset by borrowing latent information from MSRDailyActivity3D Dataset.

3.5.2.5 UTKinect-Action Dataset VS. MSRActionPairs Dataset

In this experiment, the testing data are from UTKinect-Action dataset, and source data from MSRActionPairs dataset.

Source dataset: MSRActionPairs Dataset.
Target dataset: UTKinect-Action Dataset contains 612 samples of 10 categories with 360 RGB/depth samples. The target dataset is separated into two parts for training and testing, which is listed as below.
Case 7: Training set: actions from the first five people; testing set: the rest of actions from the last five people.
Case 8: The training and testing sets are reversed.

3.5.3 Compared methods

In this section we mainly use three compared methods: LTSL [119], GFK [120], and HOG-KNN.

- LTSL learns a common subspace by transferring information between two domains via a low-rank constraint, for object and face recognition.
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

Table 3.2: Accuracy (%) of Test 1 on MSRDailyActivity3D and MSRActionPairs Datasets, with label information.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFK [120]</td>
<td>64.46</td>
<td>61.67</td>
<td>20.63</td>
<td>20.63</td>
<td>No</td>
</tr>
<tr>
<td>LTSL [119]</td>
<td>57.85</td>
<td>46.67</td>
<td>6.88</td>
<td>5.63</td>
<td>No</td>
</tr>
<tr>
<td>HOG-KNN</td>
<td>76.03</td>
<td>64.17</td>
<td>19.38</td>
<td>18.13</td>
<td>No</td>
</tr>
<tr>
<td>LTTL [56]</td>
<td><strong>86.78</strong></td>
<td><strong>90.00</strong></td>
<td><strong>40.00</strong></td>
<td><strong>33.75</strong></td>
<td>Source</td>
</tr>
<tr>
<td>SLTTL</td>
<td><strong>86.78</strong></td>
<td><strong>93.33</strong></td>
<td><strong>40.00</strong></td>
<td><strong>33.75</strong></td>
<td>Source, Target</td>
</tr>
</tbody>
</table>

- GFK calculates a common subspace by finding the correlation of source and target domains for objective classification.
- The HOG-KNN is used as a benchmark without transfer learning, i.e., the source dataset is not used in the training phase.

3.5.4 Comparison results

There are two main situations in our experiments. The first situation is that there is the same modality (RGB/depth) in both training and testing phases, what we want to do is borrow the other modality, which is considered as the “supplement” knowledge from the source dataset to the target dataset, to help improve the performance of recognition. We adopt 10 group testings in the target dataset. We discuss the RGB and depth modality in Test 1, 2, 6, 7 respectively, which is listed as Setting 1 and Setting 2 in Table 3.1.

The second situation is that there are different modalities in the training and testing phases, for example, we have RGB data for training and test data for testing, and we want to borrow some “missing” depth information from source dataset to the target dataset to help improve the performance. We discuss the RGB and depth modality in Test 3, 4, 5 and Test 8, 9, 10 respectively, which is listed as Setting 3, Setting 4 and Setting 5 in Table 3.1.

3.5.4.1 Test 1

We set Training - Testing data: RGB - RGB tensor samples. The source RGB-D data and target RGB data are used for training, while the testing data is from target domain.

In Case 1, we use 640 source data and 120 target RGB data for training, and 121 RGB data for testing. Table 3.2 shows the accuracy results of all the methods with the dimension the same
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

(a) MSR daily depth image sequence

(b) MSR daily HOG feature of RGB sequence

Figure 3.5: MSRDailyActivity3D Dataset, (a) is the depth sequence of sitting, (b) is the HOG feature of RGB sequence.

as HOG-KNN. The result of SLTTL is around 10% higher than the original space, which means that the extra depth information has an obvious advantage to improve the performance of RGB data recognition. LTSL and GFK do not work well as HOG-KNN, due to transferring some negative information, which are not proper for RGB-D action recognition. The result is reasonable because: (1) There is NO complex background in object/face recognition, whose performance is better than human action action; (2) There are similar knowledge between two intra-class objects/faces, which is different from actions containing larger diversities even belong to the same class; (3) SLTTL gets better result than LTTL in some cases (like Case 2), which indicates that the predicted target labels as a prior knowledge have positive effect on the target recognition.

3.5.4.2 Test 2

We set Training - Testing data: Depth - Depth tensor samples. The source RGB-D data and target Depth data are used for training, while the testing data is from target domain. Test 1 and Test 2 are single modality fashion in Table 3.1.

Table 3.3 shows the results of depth testing, which performs better than RGB testing compared with Table 3.2. Besides, it also shows that the extra RGB information helps not as much as the transferred depth knowledge. SLTTL gets the same results as LTTL in some cases when $\beta_2$ is 0, and we get the best result of the two models.
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

Table 3.3: Accuracy (%) of Test 2 on MSRDailyActivity3D and MSRActionPairs Datasets, with label information.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFK [120]</td>
<td>71.59</td>
<td>74.58</td>
<td>36.25</td>
<td>39.38</td>
<td>No</td>
</tr>
<tr>
<td>LTSL [119]</td>
<td>60.80</td>
<td>69.49</td>
<td>5.63</td>
<td>5.63</td>
<td>No</td>
</tr>
<tr>
<td>HOG-KNN</td>
<td>90.91</td>
<td>89.27</td>
<td>34.38</td>
<td>38.75</td>
<td>No</td>
</tr>
<tr>
<td>LTTL [56]</td>
<td>91.48</td>
<td>92.09</td>
<td>40.00</td>
<td>41.25</td>
<td>Source</td>
</tr>
<tr>
<td>SLTTL</td>
<td>91.48</td>
<td>93.79</td>
<td>40.00</td>
<td>41.25</td>
<td>Source, Target</td>
</tr>
</tbody>
</table>

Table 3.4: Accuracy (%) of Test 3 on MSRDailyActivity3D and MSRActionPairs Datasets, with label information.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFK [120]</td>
<td>18.18</td>
<td>11.86</td>
<td>22.50</td>
<td>16.25</td>
<td>No</td>
</tr>
<tr>
<td>LTSL [119]</td>
<td>10.80</td>
<td>7.34</td>
<td>6.88</td>
<td>5.00</td>
<td>No</td>
</tr>
<tr>
<td>HOG-KNN</td>
<td>22.73</td>
<td>24.29</td>
<td>28.75</td>
<td>26.25</td>
<td>No</td>
</tr>
<tr>
<td>LTTL [56]</td>
<td>35.23</td>
<td>31.07</td>
<td>29.38</td>
<td>34.38</td>
<td>Source</td>
</tr>
<tr>
<td>SLTTL</td>
<td>35.23</td>
<td>31.07</td>
<td>29.38</td>
<td>34.38</td>
<td>Source, Target</td>
</tr>
</tbody>
</table>

3.5.4.3 Test 3

We set Training - Testing data: RGB - Depth tensor samples. The source RGB-D data and target RGB data are used for training, while the testing depth data is from target domain. This is cross-modality style in Table 3.1.

As we know, the RGB and depth images are two different modalities, with small similarity. However, the similarity would be higher if the “missing” modality is compensated by transferring the additional information. Table 3.4 shows the results of all the compared methods. We can see the accuracy is not as high as Test 1, 2, because of the matching between the different modalities. However, it shows that the transferred depth knowledge helps to improve the performance by more than 10% compared with the original space. SLTTL gets the same results as LTTL in all cases, which implies the predicted target labels do not work well in this cross-modality situation.
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

Table 3.5: Accuracy (%) of Test 4 on MSRDailyActivity3D and MSRActionPairs Datasets, with label information.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFK [120]</td>
<td>7.44</td>
<td>19.17</td>
<td>15.63</td>
<td>14.38</td>
<td>No</td>
</tr>
<tr>
<td>LTSL [119]</td>
<td>7.44</td>
<td>12.50</td>
<td>5.63</td>
<td>5.00</td>
<td>No</td>
</tr>
<tr>
<td>HOG-KNN</td>
<td>12.40</td>
<td>22.50</td>
<td>17.50</td>
<td>16.88</td>
<td>No</td>
</tr>
<tr>
<td>LTTL [56]</td>
<td>23.14</td>
<td><strong>23.33</strong></td>
<td><strong>35.00</strong></td>
<td><strong>31.88</strong></td>
<td>Source</td>
</tr>
<tr>
<td>SLTTL</td>
<td><strong>28.10</strong></td>
<td><strong>23.33</strong></td>
<td><strong>35.00</strong></td>
<td><strong>31.88</strong></td>
<td>Source, Target</td>
</tr>
</tbody>
</table>

3.5.4.4 Test 4

We set Training - Testing data: **Depth - RGB** tensor samples. The source RGB-D data and target Depth data are used for training, while the testing RGB data are from target domain. This is the cross-modality fashion in Table 3.1.

Table 3.5 shows the results of RGB classification by the depth images. The additional RGB information helps to improve the accuracy around 10%. However, the results are not as higher as those in Test 3, which means the additional depth information (in Test 3) helps much more than the extra RGB knowledge (in Test 4).

3.5.4.5 Test 5

We set Training - Testing data: **RGB - RGB & Depth (RGB-D)** tensor samples. The source RGB-D data and target RGB data are used for training, while the testing RGB-D data are from target domain. This is the missing modality fashion in Table 3.1.

Table 3.6 shows all the results of compared methods for RGB-D recognition. We can see that the additional depth knowledge helps to improve the accuracy by 10% compared with the original space. While LTSL and GFK may transfer the negative information from the source domain. SLTTL obtains 10% more higher accuracy than LTTL in Case 2, which indicates the semi-supervised method is affected by different training and testing data.

3.5.4.6 Test 6

In this setting, we set Training - Testing data: **RGB - RGB** tensor samples in the target UTKinect-Action dataset. In the testing phase, there is no depth data for training, so the depth information is considered as the “supplement” knowledge in the target dataset. The goal of transfer
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

Table 3.6: Accuracy (%) of Test 5 on MSRDailyActivity3D and MSRActionPairs Datasets, with label information.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFK [120]</td>
<td>37.04</td>
<td>31.99</td>
<td>21.56</td>
<td>18.44</td>
<td>No</td>
</tr>
<tr>
<td>LTSL [119]</td>
<td>23.23</td>
<td>23.23</td>
<td>6.88</td>
<td>5.31</td>
<td>No</td>
</tr>
<tr>
<td>HOG-KNN</td>
<td>42.09</td>
<td>40.40</td>
<td>24.06</td>
<td>22.19</td>
<td>No</td>
</tr>
<tr>
<td>LTTL [56]</td>
<td>51.85</td>
<td>51.52</td>
<td>34.38</td>
<td>33.44</td>
<td>Source</td>
</tr>
<tr>
<td>SLTTL</td>
<td>51.85</td>
<td>64.31</td>
<td>34.38</td>
<td>33.44</td>
<td>Source, Target</td>
</tr>
</tbody>
</table>

Figure 3.6: Illustration of performance by $U_3$. (a) RGB samples; (b) HOG feature and eigenAction.

learning in this test is to borrow some depth knowledge, which is taken as the supplement knowledge from the source dataset $X_S$ to the target RGB dataset $X_T$. The performance of $X_T$ with the additional depth information is testified in the testing phase. Fig. 3.7 shows the results, and we can see the proposed method SLTTL is competitive compared with other methods.

The learned projection matrix $U_3 \in \mathbb{R}^{10 \times 5}$ is illustrated in Fig. 3.8. We show the perfor-
Figure 3.8: Illustration of mode-n projection matrices. $U_1, U_2$ perform on the mode-1, 2 feature, while $U_3$ on the sequence dimension.

Figure 3.9: Accuracy (%) of Test 7 (Depth - Depth) on UTKinect-Action Dataset.

Test 8: RGB - Depth

Figure 3.10: Accuracy (%) of Test 8 (RGB - Depth) on UTKinect-Action Dataset.

performance of $U_3$ of SLTTL in Fig. [3.6] from which we can see that the human information is immersed in the background. However, the eigenAction (by mode-3 projection) keeps the basic contour of the pairwise human and table, which means $U_3$ can perform well by capturing the essential energy of the sequence.
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

![Figure 3.11: Convergence property of SLTTL (β₂ = 0), sum of mode-n error with different number of iterations on [Test 6: Case 5].](image)

3.5.4.7 Test 7

In this setting, we set Training - Testing data: **Depth - Depth** tensor samples in the target UTKinect-Action dataset. In the testing phase, there is no RGB data for training, so the RGB information is considered as the “supplement” knowledge in the target dataset. The goal of transfer learning in this test is to borrow some RGB knowledge, which is taken as the supplement knowledge, from the source dataset $X_S$ to the target depth dataset $X_T$. The performance of $X_T$ with the additional RGB information is assessed in the testing phase. Fig. [3.9](image) shows the results, and we can see the proposed method SLTTL is competitive compared with other methods.

3.5.4.8 Test 8

In this setting, we set Training - Testing data: **RGB - Depth** tensor samples in the target UTKinect-Action Dataset. In the testing phase, there is no depth data for training, so the depth information is considered as the “missing” knowledge in the target dataset. The goal of transfer learning in this test is to borrow some depth knowledge, which is taken as the missing knowledge, from the source dataset $X_S$ to the target RGB dataset $X_T$. The performance of $X_T$ with the additional depth information is assessed in the testing phase. Fig. [3.10](image) shows the results, and we can see the proposed method SLTTL is competitive compared with other methods. Note that in Case 6, GFK performs better than others, which denotes that the latent RGB information has an impressive positive contribution in the testing phase.
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

3.5.4.9 Test 9

In this setting, we set Training - Testing data: **Depth - RGB** tensor samples in the target UTKinect-Action dataset. In the testing phase, there is no RGB data for training, so the RGB information is considered as the “missing” knowledge in the target dataset. The goal of transfer learning in this test is to borrow some RGB knowledge, which is taken as the missing knowledge, from the source dataset $X_S$ to the target depth dataset $X_T$. The performance of $X_T$ with the additional RGB information is assessed in the testing phase. Fig. 3.12 shows the results, and we can see the proposed method SLTTL is competitive compared with other methods.

3.5.4.10 Test 10

In this setting, we set Training - Testing data: **RGB - RGB-D** tensor samples in the target UTKinect-Action dataset. In the testing phase, there is no depth data for training, so the depth information is considered as the “missing” knowledge in the target dataset. The goal of transfer
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

learning in this test is to borrow some depth knowledge, which is taken as the missing knowledge, from the source dataset \( X_S \) to the target depth dataset \( X_T \). The performance of \( X_T \) with the additional depth information is assessed in the testing phase. Fig. 3.13 shows the results, and we can see the proposed method SLTTL is competitive compared with other methods.

3.5.4.11 Convergence and Dimensionality

We use the [Test 6: Case 5] introduced in Section 3.5.4.6 to analyze the properties of the convergence and dimensionality.

Convergence

The error for a different number of iterations is shown in Fig. 3.11, which is the convergence condition given in Algorithm 1. We can see the curves are stable within 10 iterations.

Dimensional comparison

The accuracy under different dimensions of Test 6 \( \sim 10 \) is shown in Fig. 3.14, where \([1, ..., 8]\) in x-axis indicates dimensions \{[50, 50, 5], ..., [200, 200, 5]\}. We can see that SLTTL and LTTL perform better than others in most cases. LTSL and GFK are designed to deal with single modalities, therefore they do not perform well in this experiment.

3.5.4.12 Training Time

We also compare our method with GFK [120] and LTSL [119] on the training time. We use MSR daily action RGB-D as source data and MSR pair action RGB-D as target data. All the tests are run on a PC (Intel Xeon E5 2650 CPU at 2.00 GHz and 128GB memory).

Results shown in Table 3.7 indicate that our method is significantly faster than the GFK method and the LTSL method. Our method spent 0.5 minutes training, which is \( 10 \times \) faster than the GFK method and \( 20 \times \) faster than the LTSL method.

<table>
<thead>
<tr>
<th>Methods</th>
<th>GFK [120]</th>
<th>LTSL [119]</th>
<th>SLTTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training time</td>
<td>5 minutes</td>
<td>10 minutes</td>
<td>0.5 minutes</td>
</tr>
</tbody>
</table>
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

3.5.5 Discussion

We do some experiments on the common RGB-D datasets, i.e., MSRdailyActivity3D, MSRActionPairs and UTKinect-Action datasets, to show the advantages of transferring “missing” knowledge from source domain to target domain. We also compare with the state-of-the-art transfer method for comparison, and the results verify our previous expectations. The discussions are listed as below:

First, we recover the “missing” depth modality in target domain, therefore it’s reasonable that our method can outperform other compared algorithms in most cases. In our experiments, our goal is to improve the accuracy by recovering the “missing” modality information in the target domain through a cross-modality regularizer. The previous methods LTSL and GFK fail compared with the original space, as they may transfer negative information from the auxiliary dataset. Moreover, for the cross-modality fashion in Test 1, 2, 6 and 7, the conventional methods do not consider to recover latent information and to transfer the correlation between RGB and depth modalities, so they may also transfer negative information from the “missing” modality.

Second, we couple the RGB and depth modalities and transfer their correlation by our cross-modality regularizer. In the constraint \( \text{tr}(Z^T L Z) \), the Laplace matrix \( L \) is constructed according to the different categories, where the element is set to be 1 if the two samples are belong to the same category, and 0 otherwise. Moreover, the learned \( U_n \ (n = 1, 2, 3) \) is used to find the tensor subspace, with lower dimension than original space. We can well align the source and target domain in this common subspace, by the low-rank reconstruction of source domain. Meanwhile, the reduced dimension could remove redundancies of action sequences.

Third, two datasets composed of different categories are employed in our experiments. Notice, the only common property of the two datasets is that they may have the similar visual contents, e.g., upper/lower limbs moving. However, action recognition by transfer learning is very different from the object/face recognition. For face transfer learning task, different faces contain more similar context, like the structure of eyes, nose and mouth. Therefore, it’s reasonable that the previous transfer methods for action recognition do not perform well than object/face recognition. Moreover, our method performs well even under the complex background, which means that it is more robust than the previous methods mainly focusing on simple situations set manually.

Fourth, we propose a semi-supervised approach to predict the target labels, which are used in our model each time to improve classification of target data. We construct two graphs, one for labeled source data and the other for unlabeled target data. The graph for predicting labels of target
CHAPTER 3. LOW-RANK TENSOR IN TRANSFER LEARNING

data is guided by the graph of source data, therefore we could get more reliable prediction. Both graphs are used to reconstruct the source domain, and more correlation of RGB and depth modalities could be transferred to target domain, which helps to improve the performance of classification.

3.6 Conclusion

In this paper, we addressed the problem of cross-modality action recognition with unavailable information, by transferring the useful information from the well-instituted dataset to the incomplete target dataset to find some latent information. There were mainly three situations of our application, first for the single modality case, we transferred some supplement information from the source dataset to the target dataset to improve the performance of recognition. For example, there was only RGB data in the training and testing phase, we borrowed some depth knowledge from the source dataset, which was integrated with the original RGB knowledge and got better performance in term of accuracy. Second for the “missing” modality case, we transferred some “missing” information from the source to the target dataset. For instance, there were RGB data for training while depth data for testing, and we borrowed some “missing” depth information from the source dataset. Combined with the original RGB data, the additional depth knowledge helped to couple the testing data and could improve the performance. Third for the cross-modality case, e.g., there was RGB data for training while RGB-D data for testing, and we still classified the RGB-D actions by transferring the correlation from source dataset.

We proposed a semi-supervised tensor-based transfer learning method, which employed an auxiliary domain with RGB-D modalities to find a common subspace with a RGB target domain. We employed the correlation between RGB and depth modalities by a cross-modality regularizer in the source dataset, which was transferred to help improve the performance of action recognition with single modality in the target domain. Low-rank learning was used for transferring the correlation to target domain, and finding the latent information to recover some “missing” modality information for better recognition. We predicted more reliable labels of target data, which were added in our model each time to improve the performance of classification. Extensive experiments showed that our method obtained higher accuracy and lower time complexity than the state-of-the-art methods at most cases.
Figure 3.14: The accuracy under different dimensions of Test 6 ∼ 10. Different tests indicate different modality transferring, i.e., (1) RGB-RGB, (2) Depth-Depth, (3) RGB-Depth, (4) Depth-RGB and (5) RGB-(RGB-D).
Chapter 4

Deep Non-Negative Tensor Factorization for Time Alignment

4.1 abstract

Temporal alignment aims to reduce the redundancy for two action sequences from different settings or modalities. Recent work using canonical correlation analysis (CCA) optimizes warping matrices for temporal alignment. However, densely selected frames from both sequences probably results in misalignment or overlapped alignment, and may decrease recognition performance. In this paper, we propose a generic sparse canonical temporal alignment (SCTA) method, which selects key frames from both action sequences to reduce redundancy. In addition, we solve three challenges in a unified framework for intra-class video alignments: sub-action, multi-subject, multi-modality. Our goal is to reduce the diversity of intra-class samples by aligning their key frames, which are used to find a subspace for action recognition. To that end, first we model an action sequence as a third-order tensor with spatiotemporal structure. Then we design a deep non-negative tensor factorization (DNTF) scheme to find a tensor subspace in both spatial and temporal directions. Particularly, in the first layer the original tensor is decomposed into two low-rank tensors by non-negative tensor factorization (NTF), and in the second layer each low-rank tensor is further decomposed by tensor-train (TT) decomposition. In this way, we can reduce redundant aligned frames, and extract better features from key frames in an efficient manner for recognition tasks. Experiments on synthetic data, MSRDailyActivity3D, and MSRActionPairs datasets show that our method works better than competitive methods towards the three challenges.
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

Figure 4.1: Illustration of the proposed SCTA framework. Suppose $X$ is the video dataset which can be decomposed into two low-rank tensors: $W$, $H$, where $W$ is low-dimensional projection tensor, and $H$ is the corresponding coefficients. The sparse key frames selected by SCTA in the tensor space make sure that two intra-class video sequences are well-aligned without redundant information, which is able to further boost the action recognition performance in the tensor subspace sought out by our deep NTF mechanism.

4.2 Introduction

Human action recognition in realistic scenarios has rapid growth in recent years, and achieves promising performance even with complex background [121, 122]. To mitigate the impacts of noises and redundancy, key frames are extracted to better describe actions in a video, while alleviating the high-dimensional problem [123, 124, 125]. Key frames are sufficiently informative to represent action videos, and are usually obtained by clustering [126] or based on shot [127] containing the first, middle and the last frame. However, there is an unexplored problem for key frame action recognition that key frames may contain intra-class variance due to the situations of sub-action (different scales), multi-subject and multi-modality of the same action class, as shown in Fig. 4.1. This may severely affect the accuracy of action recognition.
Temporal alignment of two action sequences can alleviate the intra-class variance, and therefore address the multi-view, multi-subject and multi-modality problems above [128, 129, 130], to name a few: coupling two sequences with trajectories [131, 132], aligning two motions by dynamic time warping (DTW) [1], warping different sequences dynamically on a manifold with spatial information [133], aligning action and facial sequences via canonical temporal warping (CTW) [128], and proposing a probabilistic CTW with extra annotations [134]. However, these methods neither consider selecting key frames from a temporal series nor address the different scales of the same action. To the best of our knowledge, how to jointly select key frames and mitigate the intra-class action variance among different scales (e.g., standing and sitting to drink), different subjects and different modalities (e.g., RGB and depth data) is still unclear. To reduce the diversity of intra-class samples when doing recognition, we perform temporal alignment only on the key frames, which guides the learning process of discriminant tensor subspace.

Recently, tensor representation has been explored for human action representation [56, 135]. A tensor is a multi-dimensional array. As to the action videos, the first and second directions (modes) of a tensor indicate row and column information of a frame, and the third mode conveys temporal knowledge. Tensor representation can preserve the spatiotemporal structure of an action video, while overcoming the “curse of dimensionality” problem [136] through the learned discriminant subspaces.

In this paper, inspired by the facts above and the flexible representation of tensor structure, we propose a tensor based generic sparse canonical temporal alignment (SCTA) for action recognition. We aim to solve three challenges caused by the diversity of intra-class samples through sparse temporal alignment and discriminant tensor subspace learning. In brief, our framework includes two components: (1) key frames selection and (2) spatiotemporal alignment. In addition, we develop a novel deep non-negative tensor factorization along with the SCTA to extract low-rank discriminant action features in a deep structure. First, as considerable redundant spatiotemporal information exists in action videos, we employ low-rank decomposition to obtain a more concise representation of a tensor structure. Second, considering the positive property of the real-world data, non-negative tensor factorization (NTF) is introduced to fulfill low-rankness as well as positive coefficient values. Third, the action features are further refined in a deep structure with NTF in the first layer, followed by a tensor-train (TT) decomposition in the second layer, which can be learned in an efficient manner. Therefore, our framework is able to tackle the three problems: sub-action, multi-subject and multi-modality in action sequence alignment. Extensive experiments on synthetic dataset, MSRDailyActivity3D action, and MSRActionPairs action datasets show that our method works better than competitive methods.
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

The contributions of this work are threefold:

• Generic temporal alignment of key frames (Fig. 4.1) framework is proposed to tackle three problems: sub-action, multi-subject and multi-modality. Through aligning key frames of pairwise intra-class action sequences, our algorithm can learn a discriminant tensor subspace from two different domains.

• Key frames of pairwise sequences are learned in a sparse canonical correlation analysis fashion. It encourages zero values on weight vectors while maintains sparse non-zero values for key frames, which automatically selects appropriate key frames from pairwise feature sequences.

• A deep NTF (DNTF) scheme is designed to find the discriminant tenser subspace from a deep structure including NTF and tensor-train (TT) building blocks. The designed structure not only ensures a low-rank tensor decomposition with positive values, but also significantly reduce the time complexity.

The rest of paper is organized as the followings. In Section II, we review relevant works of key frames selection, temporal alignment and tensor subspace learning. Second, we introduce the details of our DNTF model in Section III. Then, we illustrate the temporal alignment results on both synthetic and real-world datasets in Section IV before conclusions are drawn in Section V.

4.3 Related Work

In this section, we briefly introduce related action recognition/alignment methods in three lines: (1) key frames selection, (2) temporal alignment and (3) tensor subspace learning.

Key frames selection is able to describe action sequences regardless of noises as it rules out irrelevant frames subject to diverse impact factors. In [137], Assa et al. extract the key poses from a skeleton sequence via an affinity matrix. In [138], neighbor key frames are selected by the bag-of-words model. Junejo et al. extract trajectories and calculate the self-similarity matrix (SSM) for measurement sequences [1]. In [139], key frames are selected based on optical flows from forward and backward of the sequence. Most recently, Liu et al. extract optical flow of key frames via Adaboost and calculate co-occurrence probability of all the frames for action recognition [140]. Different from theirs, in this work, we extract sparse key frames from a pair of action sequences for joint temporal alignment and action recognition.
Temporal alignment is promising in tackling multi-view, multi-subject and multi-modality problems [129]. Recently, it arouses lots research attention in action sequences and facial expression sequences alignment. Rao et al. [131, 132] align the trajectories of different videos. Junejo et al. [1] adopt DTW to synchronize multi-view human actions. Wang and Mahadevan [141] solve manifold alignment by learning a subspace and preserving the local geometry. Zhou and De la Torre [128] propose a canonical temporal warping (CTW) framework to align sequences according to both spatial and temporal correspondence, to address multi-modal and multi-dimensional problems. Compared to these works, our method not only tackle the multi-subject and multi-modality problems, but also address an unexplored sub-action problem related to different motion scales, e.g., standing or sitting to drink. In addition, the selected key frames by our model are able to boost the action recognition performance, which will be demonstrated in the experimental section.

Tensor structure for action recognition has attracted lots of attention recently, as it can represent spatiotemporal information in a natural way. Considering local geometry of action series, Lui [142] presents the action series as a third-order tensor on the Riemann manifold, and calculates the log-distance of two samples on the tangent space. It should be noted that it does not explicitly seek for a common subspace and therefore fails to adapt to unseen datasets. Jia et al. [56] propose a tensor subspace learning method by transferring depth information from the well-established source domain to the incomplete target domain to improve the performance of missing modality recognition. To explore the positive properties of data, non-negative tensor factorization (NTF) is proposed to find a subspace for face detection [143, 144] and pose recognition [145]. Recently, inspired by deep structure to extract features [146, 147], deep semi non-negative matrix factorization (deep semi-NMF) [148, 149] is proposed for multi-view face recognition with negative values as hidden features. Different from their work, we design a novel deep NTF (DNTF) scheme composed of NTF and TT layers, which runs faster than conventional tucker decomposition and meanwhile obtaining positive features interpretation in the hidden layers for realistic data.

4.4 Sparse Canonical Temporal Alignment

In this section, we use temporal alignment to discover key frames from two videos, and only use these key frames for discriminant tensor subspace learning. To that end, we first introduce the concept of canonical temporal alignment, from which we develop the sparse canonical temporal alignment (SCTA). For better understanding, we summarize the frequently used variables of this paper in Table 4.1.
### Table 4.1: Notations and Descriptions.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{X}$</td>
<td>an $N$-order tensor indicating a dataset of action videos</td>
</tr>
<tr>
<td>$\mathcal{W}$</td>
<td>projection tensor for $\mathcal{X}$</td>
</tr>
<tr>
<td>$\mathcal{H}$</td>
<td>low dimensional representation of $\mathcal{X}$ (also is coefficient of $\mathcal{W}$)</td>
</tr>
<tr>
<td>$X^{(n)}$</td>
<td>mode-$n$ unfolding matrix of tensor $\mathcal{X}$</td>
</tr>
<tr>
<td>$W_n$</td>
<td>projection matrix by mode-$n$ NTF such that $X^{(n)} = W_nH_n$</td>
</tr>
<tr>
<td>$H_n$</td>
<td>feature representation of $X^{(n)}$ such that $X^{(n)} = W_nH_n$</td>
</tr>
<tr>
<td>$D_s, D_t$</td>
<td>the $s, t$-th sequence represented as a third-order tensor</td>
</tr>
<tr>
<td>$D_s, D_t$</td>
<td>mode-3 unfolding matrices of $D_s, D_t$</td>
</tr>
<tr>
<td>$A_s$</td>
<td>sparse warping matrix selecting key frames in $D_s$</td>
</tr>
<tr>
<td>$A_t$</td>
<td>sparse warping matrix selecting key frames in $D_t$</td>
</tr>
<tr>
<td>$d_1, d_2$</td>
<td>the height and width of a frame</td>
</tr>
<tr>
<td>$k$</td>
<td>number of frames in a video</td>
</tr>
<tr>
<td>$m$</td>
<td>total number of videos in tensor $\mathcal{X}$</td>
</tr>
<tr>
<td>$n$</td>
<td>index of the mode of a tensor</td>
</tr>
<tr>
<td>$N$</td>
<td>total number of modes in tensor $\mathcal{X}$</td>
</tr>
</tbody>
</table>

Given two intra-class action sequences from class $c \in \{1, \ldots, C\}$ represented as two third-order tensors $X_s, X_t \in \mathbb{R}^{d_1 \times d_2 \times k}$, where $d_1, d_2$ indicate the dimensions of row and column of a frame, respectively, and $k$ is number of frames, and their mode-3 unfolded matrices as $X_s, X_t \in \mathbb{R}^{k \times (d_1d_2)}$. Then, the objective of canonical temporal alignment for sequences $X_s, X_t$ can be written as:

$$
\min_{A_s, A_t, W_s, W_t} \| A_sX_sW_s - A_tX_tW_t \|^2_F + \Phi(A_s, A_t, W_s, W_t),
$$

(4.1)

where $A_s, A_t \in \mathbb{R}^{k \times k}$ warp two sequences in temporal domain and $\Phi(A_s, A_t, W_s, W_t)$ is the additional regularizer w.r.t. warping functions $A_s, A_t$, and projection matrices $W_s, W_t$. In [128], the problem above is considered as a combination of DTW and CCA, which can be solved iteratively by updating one variable when fixing others.

However, in canonical temporal alignment (CTA), the alignment results that provide the correspondence between frames from two sequences may not be necessary for high-level tasks such as action recognition. As indicated by the previous work, sparse key frames from the video sequence will work better [150]. Therefore, in this section, we propose a sparse canonical temporal alignment...
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

framework that pursues sparse correspondences between two video sequences. To that end, we introduce the column-wise sparse constraint \( \| \cdot \|_{2,1} \) to the Eq. (4.1):

\[
\min_{A_s, A_t, W} \lambda_1 \| A_s X_s W - A_t X_t W \|_F^2 + \lambda_2 \| A_s \|_{2,1} + \lambda_3 \| A_t \|_{2,1} + \Phi(A_s, A_t, W),
\]

(4.2)

where \( \lambda_p \ (p = 1, 2, 3) \) is a penalty factor. In our new framework, we seek for a common discriminant tensor space span \( W \) instead of two separated projections, and the introduced constraints on \( A_s \) and \( A_t \) will help select important key frames from two sequences. We will detail the formulation of \( \Phi(A_s, A_t, W) \) and the solutions of Eq. (4.2) in Section 4.5.

4.5 Deep Non-Negative Tensor Factorization

In this section, we propose a deep non-negative tensor factorization (DNTF) method for high-dimensional action data decomposition, followed by tensor-train (TT) decomposition in the second layer for less time complexity than tucker decomposition. In this way, the deep structure can represent the multi-linear features with reasonable non-negative properties, meanwhile disentangling multiple factors for discriminant feature learning in an efficient manner. We take mode-1 DNTF as an example to illustrate this idea in Fig. 4.2.

4.5.1 Non-Negative Tensor Factorization (NTF)

Conventional tensor decomposition methods including tucker decomposition \[151\], CANDECOMP/PARAFAC (CP) \[152\] can obtain low-rank tensor structure useful for vision problems, such as human action analysis \[59\] \[56\], human brain image recovery and texture synthesis \[153\]. Considering the positive properties of action video representations \[154\], we propose to use the non-negative tensor factorization (NTF) in the first layer of our deep structure.

Given an action dataset of \( m \) videos in \( C \) different action classes represented by a 4th-order tensor \( \mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times k \times m} \), we aim to find the decomposition \( \mathcal{X} = WH \) by the following objective \[144\] \[155\]:

\[
\arg\min_{W, H} \| \mathcal{X} - WH \|_F^2,
\]

(4.3)

where \( W \) indicates the projection tensor, \( H \) indicates the reduced dimensional tensor, and \( W \geq 0, \ H \geq 0 \). The solution is obtained by two steps: (1) we perform mode-n unfolding of \( \mathcal{X} \) to obtain
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

Figure 4.2: Schematic illustration of mode-1 deep non-negative tensor factorization (DNTF). \( X \) is a third-order tensor in the first layer of DNTF. Two matrices \( W_1 \) and \( H_1 \) are obtained by mode-1 NTF, followed by TT decompositions in the second layer to obtain \( W_1^{\text{deep}} \) and \( H_1^{\text{deep}} \).

matrix \( X^{(n)} \), (2) NMF is employed to obtain mode-\( n \) projection matrix \( W_n \) and dimensional reduced sample \( H_n \). Accordingly, Eq. (4.3) is rewritten as:

\[
\arg \min_{W_n, H_n} \| X^{(n)} - W_n H_n \|_F^2, \tag{4.4}
\]

and \( W_n \) and \( H_n \) are updated by:

\[
\begin{align*}
W_n^{ij} &\leftarrow W_n^{ij} \cdot \left( \frac{(X^{(n)} H_n^{T})^{ij}}{(W_n^{T} W_n H_n)_{ij}} \right), \\
H_n^{ij} &\leftarrow H_n^{ij} \cdot \left( \frac{(W_n^{T} X^{(n)})^{ij}}{(W_n^{T} W_n H_n)_{ij}} \right). \tag{4.5}
\end{align*}
\]

where \( W_n^{ij} \) (\( H_n^{ij} \)) is an element of \( W_n \) (\( H_n \)), and \( i, j \) indicates row and column respectively. According to the Tucker tensor decomposition, the interaction of \( \mathcal{W} \) and \( \mathcal{H} \) can be represented as:

\[
\mathcal{W}\mathcal{H} = \mathcal{H} \otimes W_1 \otimes \ldots \otimes W_n \otimes \ldots \otimes W_N, \tag{4.6}
\]

where \( W_n \in \mathbb{R}^{I_n \times J_n} \) is the mode-\( n \) projection matrix, \( \mathcal{H} \in \mathbb{R}^{J_1 \times \ldots \times J_n \times \ldots \times J_N} \) is core tensor of \( \mathcal{X} \), \( I_n \) is original feature dimension, and \( J_n \) is the reduced dimension in the tensor space. Note that in our problem, \( N = 3 \).

In addition, to better align pairwise intra-class neighbors, the action labels of training samples is taken as prior knowledge to construct a graph in the manifold. We construct a pairwise
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

\[ \gamma_1 \gamma_2 \cdots \gamma_{N-1} \gamma_N \]

Figure 4.3: Illustration of tensor-train (TT) decomposition. Note rectangles indicate tensor cores, while the circles indicate auxiliary indices.

Graph \( S \in \mathbb{R}^{m \times m} \) with the discriminant information, whose element can be defined as:

\[
S_{ij} = \begin{cases} 
1, & r \text{ nearest-neighbors of the same class;} \\
0, & \text{otherwise.}
\end{cases} \tag{4.7}
\]

Here \( S \) is used to align pairwise sequences from the same class. Finally, Eq. (4.3) can be rewritten as:

\[
\arg \min_{\mathcal{W}, \mathcal{H}} \| (\mathcal{X} - \mathcal{W}\mathcal{H})S \|^2_F, \tag{4.8}
\]

where \( S \) performs on mode-4 of \( \mathcal{X} \) and \( \mathcal{H} \). Next, we will introduce the TT decomposition to disentangle the hidden factors in \( \mathcal{W} \) and \( \mathcal{H} \).

4.5.2 Tensor-Train (TT) Decomposition

Our deep mechanism aims to further find spatiotemporal factors and more precise representations of data. TT decomposition is used for the second layer of our deep NTF model for its efficiency property explained in Section 4.5.4. The execution of TT decomposition includes:

1. decompose feature representation \( \mathcal{H} \) for different factors (spatial and temporal),
2. decompose classifier (projection tensor) \( \mathcal{W} \) to reduce its dimensions, which is inspired by TensorNet [156].

Tensor-train (TT) decomposition is fast compared with common tensor decomposition, e.g., tucker decomposition due to none recursion therein [157]. Given an \( N \)-order tensor \( \mathcal{A} \in \mathbb{R}^{I_1 \times \ldots \times I_n \times \ldots \times I_N} \) where \( I_n \) is the dimension of mode-\( n \), the TT format is written as follows:

\[
\mathcal{A}(i_1, \ldots, i_N) = \mathcal{G}_1(\gamma_0, i_1, \gamma_1) \cdots \mathcal{G}_N(\gamma_{N-1}, i_N, \gamma_N), \tag{4.9}
\]

where \( \mathcal{G}_n(\gamma_{n-1}, i_n, \gamma_n) \) is an element of tensor core \( \mathcal{G}_n \) (1 < \( n < N \)) with size \( r_{n-1} \times I_n \times r_{n+1} \), \( r_n \) is mode-\( n \) rank, \( \gamma_n \) and \( i_n \) are mode-\( n \) auxiliary indices, and \( r_0 = r_N = 1 \). Fig. 4.3 shows the TT format, the circles contain the auxiliary indices \( \gamma_{n-1} \) and \( \gamma_n \) which connect two cores \( \mathcal{G}_{n-1} \) and \( \mathcal{G}_n \) in the rectangles. The tensor-train means we have to multiply all the elements of small core tensors and sum over all the indices.
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

Given an $N$-order tensor $\mathcal{X}$, we decompose it in the first layer using NTF as $\mathcal{X} = \mathcal{W}\mathcal{H}$, where $\mathcal{W}$ is a classifier and $\mathcal{H}$ is the feature representation. In the second layer, we apply TT decomposition on $\mathcal{H}$ and $\mathcal{W}$ which is formulated as:

$$\mathcal{X} = \mathcal{W}H = \mathcal{W} \prod_{n=1}^{N} U_n G_n = \mathcal{W}'G = \prod_{n=1}^{N} C_n G = CG,$$  \hspace{1cm} (4.10)

where $\mathcal{W}' = \mathcal{W} \prod_{n=1}^{N} U_n$, $G = \prod_{n=1}^{N} G_n$ and $C = \prod_{n=1}^{N} C_n$.

First, $\mathcal{H}$ is TT decomposed to obtain mode-$n$ core $G_n$ and matrix $U_n$, which contains spatiotemporal factors. Similar with deep semi-NMF [138] model, we integrate $\mathcal{W}$ and $U_n$ to be new $\mathcal{W}'$, which contains spatial and temporal factors drawn from $\mathcal{H}$. Second, $\mathcal{W}'$ is TT decomposed to obtain new mode-$n$ core $C_n$, which is similar with TensorNet model [136] to obtain low-rank transformation. Finally $\mathcal{W} \leftarrow C$ and $\mathcal{H} \leftarrow G$ in the deep layer of our model. Specifically, given a third-order tensor $\mathcal{X}$, we take the mode-$l$ DNTF for instance. There are two layers for decomposition, the first layer we perform NTF on mode-$l$ matrix $X^{(l)}$ to get two low-rank matrices $W_1$ and $H_1$, i.e., $X^{(1)} = W_1 H_1$. Then $W_1$ and $H_1$ are further decomposed by TT decomposition in the second layer, i.e., $X^{(1)} = W_1 H_1 \overset{TT}{\rightarrow} W_1 U_1 G_1 = W_1' G_1 \overset{TT}{\rightarrow} C_1 G_1$. Then $W_1^{\text{deep}} \leftarrow C_1$ and $H_1^{\text{deep}} \leftarrow G_1$. Next we will introduce our objective function and DNTF scheme in details.

### 4.5.3 Objective Function and Solutions

Given a dataset with $m$ tensor actions $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times m}$, $\mathcal{X}_s \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is the $s$-th sample. We decompose $\mathcal{X}$ by Eqs. (4.3)~(4.4) to obtain mode-$n$ projection matrix $W_n$ ($n = 1, 2, 3$), and $s$-th low-dimensional sample is obtained by $D_s = \mathcal{X}_s \times_1 W_1^{-1} \times_2 W_2^{-1} \times_3 W_3^{-1}$. $D_s$ is mode-3 unfolded to be $D_s \in \mathbb{R}^{I_3 \times (I_1 J_2)}$, where each row indicates one frame of the sequence. So does for another action sequence $D_t$ in the same class. Considering spatial decomposition in Eq. (4.8) and temporal alignment in Eq. (4.1), our objective function is formulated as:

$$\min_{\mathcal{W}, \mathcal{H}} \| (\mathcal{X} - \mathcal{W}\mathcal{H}) S \|_F^2 + \sum_{c=1}^{C} \sum_{s,t \in c} \lambda_1 \| A_s D_s - A_t D_t \|_F^2 + \lambda_2 \| A_s \|_{2,1} + \lambda_3 \| A_t \|_{2,1},$$ \hspace{1cm} (4.11)

s.t. $A_s D_s D_s^T A_s^T = I$, $A_t D_t D_t^T A_t^T = I$,

where $A_s, A_t \in \mathbb{R}^{I_3 \times J_3}$, $I \in \mathbb{R}^{J_3 \times J_3}$ is an identity matrix, and $\lambda_p$ ($p = 1, 2, 3$) is penalty coefficient of each item. The constrains keep the solution non-trivial. In our objective function, the first item
finds the subspace by performing Tucker decomposition when keeping spatial features have non-negative values in practice. The second item aligns the two series of key frames to handle sub-action, multi-subject and multi-modality problems. The third and fourth items are used to select the key frames of intra-class samples by sparse constraint to eliminate temporal redundancy. Next we will introduce solutions to the function by jointly optimizing deep non-negative factorization and temporal sparse weight allocation.

As the learning problem in Eq. (4.25) is not jointly convex over all unknown variables, we propose to use the Lagrange Multiplier method to optimize: \( W, H, A_s \) and \( A_t \). Let us first write down the Lagrange Multiplier function:

\[
L = \|(X - WH)S\|_F^2 + \sum_{c=1}^{C} \sum_{s, t \in c} \lambda_1 \|A_s D_s - A_t D_t\|_F^2 + \lambda_2 \|A_s\|_{2, 1} + \lambda_3 \|A_t\|_{2, 1} + \text{tr} \left( Y_1 (A_s D_s A_s^T - I) \right) + \text{tr} \left( Y_2 (A_t D_t A_t^T - I) \right),
\]

(4.12)

where \( Y_1 \) and \( Y_2 \) are Lagrangian multipliers. All the variables are optimized iteratively, including two-layer decomposition, which are listed as following.

**4.5.3.1 First Layer of DNTF**

The first order gradients of \( L \) with respect to different variables equal to 0, including: mode-\( n \) projection matrix \( W_n \), low-dimensional sample \( H_n \) and warp matrices of temporal direction \( A_s, A_t \).

**Update \( W_n \):**

\[
W_n^{ij} \leftarrow W_n^{ij} \cdot \frac{(X^{(n)} S S^T H_n^T)^{ij}}{(W_n H_n S S^T H_n^T)^{ij}},
\]

(4.13)

where \( i, j \) indicates the row and column of \( W_n \).

**Update \( H_n \):**

\[
H_n^{ij} \leftarrow H_n^{ij} \cdot \frac{(X^{(n)} S W_n^T S^T)^{ij}}{(W_n H_n S W_n^T S^T)^{ij}}.
\]

(4.14)

**Update \( A_s \):**

\[
A_s^{ij} \leftarrow A_s^{ij} \cdot \frac{(I + \frac{Y_1}{\lambda_1})^{-1} A_t D_t D_t^T - \lambda_2 \|A_s\|_{2, 1}}{(D_s D_s^T)^{ij}},
\]

(4.15)
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

Algorithm 1 SCTA (Solving Problem Eq. (4.20))

Input: $X, \lambda_1, \lambda_2, \lambda_3$

Initialize: $A_s = A_t = 0$.

while not converged do
  for Mode-n alternation do
    1. First layer of DNTF and update $W_n^{ij}$ by Eq. (4.13).
    2. First layer of DNTF and update $H_n^{ij}$ by Eq. (4.14).
    3. First layer of DNTF and update $A_s^{ij}$ by Eq. (4.15).
    4. First layer of DNTF and update $A_t^{ij}$ by Eq. (4.16).
    5. TT decomposition by Eq. (4.10).
    6. Second layer of DNTF and update $W_n^{ij}$ by Eq. (4.13).
    7. Second layer of DNTF and update $H_n^{ij}$ by Eq. (4.14).
    8. Second layer of DNTF and update $A_s^{ij}$ by Eq. (4.15).
    9. Second layer of DNTF and update $A_t^{ij}$ by Eq. (4.16).
    10. Check the convergence conditions $\|X - WH\|_2 < \epsilon$.
  end for
end while

Output: $W_n, H_n$.

Update $A_t$:

$$A_t^{ij} \leftarrow A_t^{ij} \cdot \frac{((I + \frac{\lambda_2}{\lambda_1})^{-1}A_sD_sD_t^T - \lambda_3\|A_t\|_2,1)^{ij}}{(D_tD_t^T)^{ij}}. \quad (4.16)$$

4.5.3.2 Second Layer of DNTF

We update $W_n$ and $H_n$ by TT decomposition analyzed in Eq. (4.10).

Update $W_n$: As $\mathcal{W}' = \prod_{n=1}^{N} C_n$, we perform $W_n \leftarrow C_n$, then Eq. (4.13) is used to update $W_n$.

Update $H_n$: As $\mathcal{H} = \prod_{n=1}^{N} U_nG_n$, we perform $H_n \leftarrow G_n$, then Eq. (4.14) is used to update $H_n$.

Update $A_s$ and $A_t$: $A_s$ and $A_t$ are updated by Eq. (4.15) and (4.16). The procedure of solution is shown in Algorithm 1.

4.5.4 Time Complexity Analysis

Given an $N$-order tensor $X \in \mathbb{R}^{I_1 \times \ldots \times I_n \times \ldots \times I_N}$ where $I_n$ and $r_n$ are the mode-$n$ dimension and rank, we discuss the time complexity of the key decomposition steps. For simplicity, we skip
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

the subscript, i.e., \( I_n \rightarrow I \) and \( r_n \rightarrow r \). We mainly compare TT decomposition in our DNTF model with high-order single value decomposition (HOSVD), which takes \( O(NIR^3) \) operations. For a TT decomposition, each core \( G_n(r_{n-1}, I_n, r_{n+1}) \) is unfolded to be a matrix \( G_n \in \mathbb{R}^{(Ir) \times r} \) through SVD needs and will take \( O(Ir^3) \) operations for each mode. Therefore, there are in total \( O(NIr^3) \) steps for the TT decomposition of \( X \). We can see that HOSVD takes much more time than the TT decomposition when \( N \gg 3 \).

4.5.5 Model Comparison

The most relative works to our model include: 1) generalized canonical time warping (GTW) \(^{[129]}\) for temporal alignment and 2) subspace alignment model (SA) \(^{[158]}\) for recognition. We set two sequences \( D_s, D_t \in \mathbb{R}^{k \times (d_1, d_2)} \) and warping matrices \( A_s, A_t \in \mathbb{R}^{k \times k} \) as Eq. (4.1) defines.

4.5.5.1 Temporal Alignment Model

GTW finds spatiotemporal correlations based on CCA, and adds a soft penalty on the warping path by minimizing:

\[
\min_{W_s, W_t} \sum_{s,t=1}^{m} \| W_s^T D_s A_s - W_t^T D_t A_t \|_F^2 + \sum_s \eta \| F_l Q a_s \|_2^2,
\]

where \( W_s \) and \( W_t \) are spatial transformations, \( F_l \in \mathbb{R}^{l \times l} \) is the first order differential operator and \( Q a_s \in \mathbb{R}^l \) is the warping path.

Compared to GTW, our model aligns two sequences by the key frames one-by-one, which eliminates the redundant frames or overlapping of sequences by sparse learning:

\[
\min_{A_s, A_t} \lambda_1 \| A_s D_s - A_t D_t \|_F^2 + \lambda_2 \| A_s \|_{2,1} + \lambda_3 \| A_t \|_{2,1},
\]

where \( A_s \) and \( A_t \) are sparse warping matrices to select key frames by \( L_{2,1} \) norm.

4.5.5.2 Subspace Alignment (SA) Model

The state-of-the-art subspace alignment methods are usually used for domain adaption, e.g., SA aligns the subspaces of two domains. Given source domain data \( D_s \) and target domain data \( D_t \), first PCA is performed on both domains to find two subspaces \( P_S \) and \( P_T \), then SA aligns the two subspaces by:

\[
T_S \leftarrow D_s (P_S P_S^T P_T), \quad T_T \leftarrow D_t P_T,
\]

85
where $T_S$ and $T_T$ are transformed data whose distances are measured in a new subspace.

Different from SA, our model aligns intra-class samples distributed in two domains element-by-element, particularly, frame-by-frame in the action sequences. Additionally, we find one shared subspace for both domains instead of two, by a DNTF mechanism:

$$\min_{W, H} \|X - WH\|_F^2 S + \sum_{c=1}^{C} \sum_{s,t=1}^{m} \lambda_1 \|A_s D_s - A_t D_t\|_F^2,$$  \hspace{1cm} (4.20)

where $W$ is used for find a new common tensor subspace, and the second term is the frame-by-frame alignment of intra-class samples in two domains. In addition, our model is designed for 3 challenges: sub-action, multi-subject and multi-modality, which are not fully solved by the state-of-the-art.

### 4.6 Experiment

This section includes three experiments: (1) temporal alignment of synthetic data, (2) generic temporal alignment for three problems: sub-action, multi-subject and multi-modality and (3) systematic evaluations of action recognition on two realistic datasets.

#### 4.6.1 Datasets & Experiment Setting

In this subsection, there are three experiment settings: (1) synthetic temporal alignment (Section 4.6.2); (2) DNTF mechanism with two layers for subspace alignment comparison (Section 4.6.3); (3) action recognition with first layer (NTF) under different scenarios (Section 4.6.4).

We evaluate on two popular datasets: MSRDailyActivity3D action dataset\(^1\) and MSRActionPairs action dataset\(^2\). Fig. 4.5 shows the drinking action and picking up sequences from the two datasets. In both datasets, we explore RGB and depth image modalities for three challenges discussed in this paper. The three datasets are introduced below.

#### 4.6.1.1 Synthetic Dataset

We generate three sequences randomly to compare the temporal alignment results of different methods.

---

\(^1\)http://users.eecs.northwestern.edu/jwa368/my_data.html
\(^2\)http://www.cs.ucf.edu/oreifej/HON4D.html
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

Figure 4.4: Synthetic data evaluations. Original triple sequences $X_i, i \in \{1, 2, 3\}$ are generated first, with additional Gaussian noises in the third dimension. Spatiotemporal warping functions are calculated by pDTW, pDDTW, pIMW, pCTW, GTW and SCTA, respectively. pCTW, GTW and SCTA are based on CCA to align the homogeneous resources, and rule out the noises from the third dimension. sub-figure on upper right shows different warping paths, while that on bottom right indicates mean alignment errors.

4.6.1.2 MSRDailyActivity3D Dataset

In this dataset, there are 16 different actions performed by 10 subjects, each of which acts twice. The actions are: drink, eat, read book, call cellphone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lie down on sofa, walk, play guitar, stand up, and sit down. We use the cropped depth data in our experiment. First, each action is sub-sampled to $80 \times 80 \times 10$, and then we use a Gabor filter to extract feature from the sequence.

4.6.1.3 MSRActionPairs Dataset

This dataset includes 12 action categories into six pairs: picking up a box and picking down a box, lifting a box and placing a box, pushing a chair and pulling a chair, wearing a hat and taking off a hat, putting on a backpack and taking off a backpack, Sticking a poster and Removing a poster. Each action has 10 instances, each of which plays three trials. So there are total 360 samples and each category contains 30 samples. In this experiment, we explore the HOG feature instead of Gabor to improve the performance of SSM method. Each action sample size is $84 \times 53 \times 20$ after extracting HOG features.

87
4.6.2 Synthetic Temporal Alignment

In this subsection, we generate three sets of signals to evaluate the proposed SCTA and others to demonstrate the performance of key frames selection incorporated with temporal alignment. Notably, both GTW and SCTA can align spatiotemporal features, but the main difference between them lies in that SCTA employs sparse constraint to select key frames and rules out noisy frames of a pair of action sequences. Next, we detail the competitive methods used in this experiment:

4.6.2.1 Procrustes dynamic time warping (pDTW)

pDTW is an extension of DTW, which is proposed for shape alignment [129]. pDTW aligns two sequences by minimizing:

\[
J_{pDTW}(A_{s/t}) = \sum_{s,t=1}^{m} \frac{1}{2} \|D_s A_s - D_t A_t\|^2_F, \tag{4.21}
\]

where \(A_{s/t} \in \{0, 1\}\) is the warping matrix and \(D_{s/t}\) is \(s/t\)-th sequence drawn from \(m\) samples.

4.6.2.2 Procrustes derivative dynamic time warping (pDDTW)

pDDTW is based on DDTW [159], which uses derivatives of features. pDDTW aligns two sequences by minimizing:

\[
J_{pDDTW}(A_{s/t}) = \sum_{s,t=1}^{m} \frac{1}{2} \|D_s F_s A_s - D_t F_t A_t\|^2_F. \tag{4.22}
\]
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

where $F_{s/t}$ is the first order differential operator.

4.6.2.3 Procrustes iterative motion warping (pIMW)

IMW iteratively handles time warping and spatial transformation of two sequences \cite{160}, and pIMW is extended to align multiple sequences by minimizing:

$$J_{pIMX}(A_{s/t}, R_{s/t}, O_{s/t}) = \sum_{s,t=1}^{m} \frac{1}{2} \| (D_s \circ R_s + O_s) A_s - (D_t \circ R_t + O_t) A_t \|_F^2$$

$$+ \sum_{s=1}^{m} \left( \eta_s^a \| R_s F_s^a \|_F^2 + \eta_s^b \| O_s F_s^b \|_F^2 \right),$$

where $R_{s/t}, O_{s/t}$ are scaling and translating parameters. $F_{s/t}^a, F_{s/t}^b$ are first order differential operators.

4.6.2.4 Procrustes canonical time warping (pCTW)

pCTW minimizes the distance of two sequences in low dimensional space, and aligns the warping paths of them by:

$$J_{pCTW}(W_s, W_t, A_s, A_t) = \sum_{s,t=1}^{m} \| W_s^T D_s A_s - W_t^T D_t A_t \|_F^2 + \phi(W_s) + \phi(W_t),$$

where $\phi(W) = \frac{1}{1-\eta} \|W\|_F^2$, and $W_s, W_t$ satisfy the orthogonal constraints:

$$\begin{align*}
W_s^T \left( (1-\eta) D_s A_s A_s^T D_s^T + \eta I \right) W_s &= I, \\
W_t^T \left( (1-\eta) D_t A_t A_t^T D_t^T + \eta I \right) W_t &= I,
\end{align*}$$

where $\eta \in \{0, 1\}$ is a penalty between the error and regularization terms.

Fig. 4.4 shows the results of temporal alignment of triple sequences of synthetic data. It can be seen that pDTW fails because of distorted spatial sequences. The feature derivatives of pDDTW do not well capture the structure of sequences. pIMW overfits the sequences and the noise (third spatial component), whereas pCTW and GTW can successfully select features therefore removing the noisy dimension. SCTA performs not only spatial but also temporal feature selection (key frames selection), and yields small alignment error.
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

Table 4.2: Accuracy of sub-action and multi-subject problems.

<table>
<thead>
<tr>
<th>Problem</th>
<th>JTM</th>
<th>GFK</th>
<th>LSSA</th>
<th>$\lambda_p = 0$</th>
<th>Ours-I</th>
<th>Ours-II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-action</td>
<td>0.79</td>
<td>0.86</td>
<td><strong>0.90</strong></td>
<td>0.80</td>
<td>0.89</td>
<td><strong>0.90</strong></td>
</tr>
<tr>
<td>Multi-subject</td>
<td>0.77</td>
<td>0.79</td>
<td>0.84</td>
<td>0.84</td>
<td><strong>0.87</strong></td>
<td><strong>0.87</strong></td>
</tr>
</tbody>
</table>

4.6.3 Three Challenges in Temporal Alignment

In this subsection, we design three experiments to demonstrate the capability of our method to address the three challenges in temporal alignment. A subset of MSRActionPairs dataset is used for the evaluations where three body appearances, two modalities of 10 subjects are selected. Here we briefly introduce the competitive methods in this section:

- Transfer Joint Matching (TJM) \[161\] minimizes the variance between source and target data in a new subspace by assigning less penalty on source data irrelevant to target data by a kernel mapping.
- GFK \[162\] learns many intermediate subspaces on a manifold to align source and target domains for transfer learning.
- SA \[158\] finds the subspaces of source and target domains and transforms source subspace by an affinity matrix to couple target subspace.
- LSSA \[163\] performs kernelized SA based on selecting landmarks from source and target domains.

**Sub-action problem.** In the same class, there are some shape variations in details, for example, drinking action when people standing or sitting on sofa. Considering this partial variation, we represent an action sequence as a hierarchical structure including common part and individual part called sub-action, and we aim to mitigate the diversity by taking individual part into account.

**Multi-subject problem.** Different people perform the same action in different manners, like velocity and motion scale. We aim to reduce the variations between different people, and maximize the coherence of the same action. Table 4.2 shows the performances of sub-action and multi-subject problems.

**Multi-modality problem.** Different modalities may help to improve performance as a complement to each other. Here we employ RGB and depth data for multi-modality setting. We create four different scenarios for this problem: (1) Training-Testing setting: RGB-Depth modalities of 10
Table 4.3: Accuracy of multi-modality problem, Train-Test: RGB-Depth & Depth-RGB.

<table>
<thead>
<tr>
<th>Mode</th>
<th>TJM</th>
<th>GFK</th>
<th>SA</th>
<th>LSSA</th>
<th>$\lambda_p = 0$</th>
<th>Ours-I</th>
<th>Ours-II</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB-Depth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-subject</td>
<td>0.08</td>
<td>0.11</td>
<td>0.05</td>
<td>0.17</td>
<td>0.26</td>
<td>0.37</td>
<td>0.37</td>
</tr>
<tr>
<td>Sub-action</td>
<td>0.08</td>
<td>0.05</td>
<td>0.06</td>
<td>0.19</td>
<td>0.24</td>
<td>0.25</td>
<td>0.33</td>
</tr>
<tr>
<td>Depth-RGB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-subject</td>
<td>0.08</td>
<td>0.07</td>
<td>0.10</td>
<td>0.17</td>
<td>0.19</td>
<td>0.31</td>
<td>0.37</td>
</tr>
<tr>
<td>Sub-action</td>
<td>0.08</td>
<td>0.07</td>
<td>0.12</td>
<td>0.16</td>
<td>0.20</td>
<td>0.36</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Figure 4.6: Accuracy of DNTF on MSRActionPairs dataset for multi-subject problem.

subjects. Particularly, RGB data are used for training and reference and we evaluate the labels of new depth data. (2) Training-Testing setting: Depth-RGB modalities of 10 person. (3) Training-Testing setting: RGB-Depth modalities of three sub-actions. RGB data for training and reference and depth for testing. (4) Training-Testing setting: Depth-RGB modalities of three sub-actions. Here we employ some recent subspace alignment methods for comparison.

Table 4.2 shows the results of the first two problems. $\lambda_p = 0$ ($p = 1, 2, 3$) is the degenerated model of our method. “Ours-I” indicates our single layer model by NTF, and “Ours-II” means our two layers model DNTF with both NTF and TT. In most cases, our methods achieve better results. Fig. 4.6 shows the accuracies under different dimensions, from which we can see that DNTF obtains higher accuracy in lower dimensional space on each mode.

Table 4.3 shows the accuracy of subspace alignment methods for cross-modality experiments, i.e., different modalities for training and testing. We can see that LSSA performs better than SA, which verifies that the landmarks based method is reasonable. Note $\lambda_p = 0$ ($p = 1, 2, 3$) stands
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

Figure 4.7: Visualization of temporal alignment of key frames from two intra-class action sequences with 20 frames. The first two rows show the result for multi-subject challenge in “depth” modality, from which we can see that the key frames of “putting things on chair” actions are well aligned. The last two rows show the result for multi-modality challenge, from which we can see that the key frames of “putting things on floor” actions are aligned as well.

for DNTF model without sparse temporal alignment. Both DNTF with key frames selection Ours-I and deep model Ours-II obtain better accuracy in most cases, which demonstrates the deep structure is able to extract more discriminant features for action recognition. Fig. 4.7 shows the alignment results for multi-subject and multi-modality problems. We can see that the key poses of an action are captured and aligned properly.

4.6.4 Action Recognition Results

4.6.4.1 Competitive Methods

In this subsection we introduce two competitive methods, and three scenarios with different parameters setting of our model for comparisons.

- Discriminant Non-Negative Tensor Factorization (DsNTF) [164] integrates the Fisher criterion into the NTF for discriminant feature learning.

- Self-Similarity Matrix (SSM) [11] can measure two action sequences frame-by-frame, and is insensitive to multi-view problem and individual diversity.
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

Figure 4.8: Illustration of temporal key frames alignment on MSRDailyActivity3D dataset. Left: Sit action. Solid lines connect the key frames between two sequences. Middle: SSM of two action sequences by $\mathbb{I}$. The red curve is the realistic aligned frames while the green dots are the aligned key frames. Right: SSM by our method after sampling. The green dots indicate the key frames while the red curve shows the aligned path.

• Scenario 1 ($S_1$): $\lambda_p = 0$. Sparse temporal alignment is removed from our model.

• Scenario 2 ($S_2$): $A_s/t = \mathbb{I}$. Temporal alignment is adopted but no sparse constraint.

• Scenario 3 ($S_3$): $\lambda_p > 0$ and $A_s, A_t \neq \mathbb{I}$, which indicates both key frames selection and temporal alignment. To evaluate the key frames selection, we set $\lambda_p \in \{0, 1, 1000\}$, to show the effect of different weights for the recognition task.

4.6.4.2 MSRDailyActivity3D Dataset

To better illustrate the correspondence between frames, we introduce the concept of self-similarity matrix (SSM). SSM is a $k \times k$ matrix indicates the pairwise distances of all $k$ frames, and each element is calculated by: $\|(X)^i - (X)^j\|$, where $(X)^i$ and $(X)^j$ indicate the features of the $i$- and $j$-th frames, respectively. The entry $(i, j)$ in SSM tends to be larger if the two frames are significantly different. A few SSMs drawn from MSRDailyActivity3D dataset shows the selected key frames from two sequences (Fig. 4.8). The left sub-figure is the schematic diagram of our STCA method, which selects the key frames of two intra-class action sequences (drinking) for alignment. The middle subfigure illustrates the SSM of two sequences frame-by-frame. Note that the green dots are key frames aligned manually, and the red curve is the corresponding aligned path by $\mathbb{I}$. The right subfigure shows the key frames selected automatically and aligned path on SSM by our method. In brief, most of the key frames locate in the dark areas with lower SSM values, which indicates the frame pairs from two sequences (x,y-axis) with large similarity.

As there are 10 subjects in each category, we use 5, 6, 7, 8, 9 subjects for training each time, and the rest for testing. In this experiment, we select the dimension settings $[10, 10, 10]$ and
Figure 4.9: Mean accuracy of the proposed method with $\lambda_1 \in [0, 1]$, $\lambda_2 = \lambda_3 = 1$ under different dimension settings on two datasets. Note $\lambda_1 = 0$ is baseline, and we can see the accuracy under $\lambda_1 > 0$ is higher than that of $\lambda_1 = 0$.

[40, 40, 10] to see the performance under relative lower and higher dimensional spaces. Corresponding results are shown in Table 4.4 and Table 4.5, respectively. From Table 4.4 we can see that the better performance is obtained with the increasing training number when $\lambda_1 = 1$, which also overcomes the situation of larger $\lambda_1 (10^3)$. Note in this experiment we set $\lambda_2 = \lambda_3 = 1$ when $\lambda_1 > 0$. It can be concluded that the best parameter is obtained at $\lambda_1 \in [0, 1]$. In addition, we can see that $A_{s/t} = I$ performs better than $\lambda_1 = 0$ in most cases, meaning temporal alignment plays a key role for accuracy. The corresponding result of parameter tuning is shown in Fig. 4.9(a). In general, $\lambda_1 > 0$ performs better than $\lambda_1 = 0$. The mean accuracy curve calculated under all the dimensions is increasing and reaches the peak at $\lambda_1 = 1$, meaning the key frames selection is effective on action recognition task.

From Table 4.5 we can see the best result is obtained at $\lambda_1 = 1$. DsNTF is competitive with ours, while SSM performs worse in most cases. We believe the reason is that the size of dimensions or frames length is insufficient for SSM to find the similarity of two sequences. From Table 4.4 ∼ 4.5 we can see that our performance is better than others given the increasing number of subjects for training.
Figure 4.10: First three sub-figures: mode-n error in different iterations on MSRActionPairs dataset. Fourth sub-figure: accuracy under mode-n \((n = 1, 2)\) dimensions. We can see that mode-3 dimension \(\text{Dim}(3) = 7\) achieve relatively good results.

### 4.6.4.3 MSRActionPairs Dataset

As there are 30 samples in each category, we use 16 ∼ 20 samples for training respectively, and the rest for testing. We use the dimension settings \([20, 20, 20]\) and \([40, 40, 40]\) for evaluations. The corresponding results are shown in Tables 4.6, 4.7. Similarly, in this experiment we set \(\lambda_2 = \lambda_3 = 1\) when \(\lambda_1 > 0\). In Table 4.6 we can see \(\lambda_1 = 1\) is comparative to other methods, slightly worse than \(\lambda_1 = 0\) \((\approx 2\%)\) under 20 training samples. However, the mean accuracy of the former is consistently higher than the latter in Fig. 4.9(c). In addition, we can find that \(A_{s/t} = I\) is better than \(\lambda_1 = 0\) in most cases, meaning temporal alignment plays a positive role for accuracy. On the other hand, the SSM method get improved compared to the results in the last experiment. We believe the proper features and dimension are critical for SSM.

In Table 4.7 we can see SSM’s performance decreases along with increasing number of training data. The main reason is that it does not have a training process, and therefore its accuracy is not necessarily related to the number of training samples. Fig. 4.9(d) shows the accuracy under \(\lambda_1 \in [0, 1]\) with 20 training samples, which indicates that the best performance is obtained at \(\lambda_1 = 0.3\). From Table 4.6 ∼ 4.7 we can see that our accuracy is higher than others given the
### Table 4.4: Accuracy (%) of MSRDailyActivity3D with setting $[10,10,10]$. 

<table>
<thead>
<tr>
<th>#Subs</th>
<th>DsNTF</th>
<th>SSM</th>
<th>$\lambda_1 = 0$</th>
<th>$\Phi(\cdot)$</th>
<th>$\lambda_1 = 0.1$</th>
<th>$\lambda_1 = 1$</th>
<th>$\lambda_1 = 10^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>27.50</td>
<td>25.63</td>
<td>36.88</td>
<td>33.13</td>
<td>29.38</td>
<td>31.88</td>
<td>51.25</td>
</tr>
<tr>
<td>6</td>
<td>44.38</td>
<td>28.13</td>
<td>36.88</td>
<td><strong>54.38</strong></td>
<td>44.38</td>
<td>45.00</td>
<td>51.88</td>
</tr>
<tr>
<td>7</td>
<td>57.50</td>
<td>25.00</td>
<td>50.00</td>
<td>56.25</td>
<td>47.50</td>
<td>60.00</td>
<td><strong>61.25</strong></td>
</tr>
<tr>
<td>8</td>
<td>58.13</td>
<td>23.44</td>
<td>65.63</td>
<td>69.38</td>
<td>64.38</td>
<td><strong>72.50</strong></td>
<td>60.63</td>
</tr>
<tr>
<td>9</td>
<td>80.00</td>
<td>34.38</td>
<td>80.63</td>
<td>78.75</td>
<td>82.50</td>
<td><strong>83.75</strong></td>
<td>80.63</td>
</tr>
</tbody>
</table>

### Table 4.5: Accuracy (%) of MSRDailyActivity3D with setting $[40,40,10]$. 

<table>
<thead>
<tr>
<th>#Subs</th>
<th>DsNTF</th>
<th>SSM</th>
<th>$\lambda_1 = 0$</th>
<th>$\Phi(\cdot)$</th>
<th>$\lambda_1 = 0.1$</th>
<th>$\lambda_1 = 1$</th>
<th>$\lambda_1 = 10^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>33.75</td>
<td>31.25</td>
<td><strong>40.63</strong></td>
<td>28.75</td>
<td>22.50</td>
<td>35.63</td>
<td><strong>40.63</strong></td>
</tr>
<tr>
<td>6</td>
<td><strong>53.13</strong></td>
<td>32.03</td>
<td>35.00</td>
<td>43.75</td>
<td>51.25</td>
<td>33.75</td>
<td>40.00</td>
</tr>
<tr>
<td>7</td>
<td>39.38</td>
<td>35.42</td>
<td>33.13</td>
<td>58.13</td>
<td>59.38</td>
<td><strong>63.13</strong></td>
<td>55.63</td>
</tr>
<tr>
<td>8</td>
<td>66.88</td>
<td>34.38</td>
<td>68.13</td>
<td>65.00</td>
<td>59.38</td>
<td><strong>71.25</strong></td>
<td>58.75</td>
</tr>
<tr>
<td>9</td>
<td>78.75</td>
<td>46.88</td>
<td>77.50</td>
<td>80.63</td>
<td>75.63</td>
<td><strong>81.25</strong></td>
<td>80.63</td>
</tr>
</tbody>
</table>

Increasing number of training samples at most cases.

Fig. 4.10(a)–4.10(c) show mode-$n$ error along different iterations, when $\lambda_1 = 0$, 0.3 and 1000 respectively. We can see that the error is stable within a few iterations, which indicates that our method converges well on realistic data. Fig. 4.10(d) shows the accuracy under different dimensions of mode-1,2, from which we can see that the better result is obtained by mode-3 with dimension $\text{Dim}(3) = 7$. In summary, the results above indicate that the performance is optimized by proper mode-$n$ dimensions. Either insufficient or redundant information will affect the performance.

#### 4.6.5 Parameters Analysis & Time Complexity

In this subsection, we systematically analyze four factors of our DNTF model with two layers on MSRAActionPairs dataset, including (1) penalty parameters $\lambda_p$ ($p = 1, 2, 3$), (2) objective function value (OFV), (3) signal-to-noise ratio (SNR), and (4) time complexity comparison.
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

Table 4.6: Accuracy (%) of MSRActionPairs with setting [20, 20, 20].

<table>
<thead>
<tr>
<th>#Train</th>
<th>DsNTF</th>
<th>SSM</th>
<th>$\lambda_1 = 0$</th>
<th>$\Phi(\cdot)$</th>
<th>$\lambda_1 = 0.1$</th>
<th>$\lambda_1 = 1$</th>
<th>$\lambda_1 = 10^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>67.26</td>
<td>72.62</td>
<td>64.88</td>
<td>46.43</td>
<td>67.26</td>
<td>78.57</td>
<td>64.88</td>
</tr>
<tr>
<td>17</td>
<td>75.64</td>
<td>76.28</td>
<td>60.26</td>
<td><strong>83.33</strong></td>
<td>76.28</td>
<td>80.77</td>
<td>65.38</td>
</tr>
<tr>
<td>18</td>
<td>73.61</td>
<td>73.61</td>
<td>63.89</td>
<td>74.31</td>
<td>72.92</td>
<td><strong>79.86</strong></td>
<td>78.47</td>
</tr>
<tr>
<td>19</td>
<td>70.45</td>
<td>75.00</td>
<td>49.24</td>
<td>68.94</td>
<td>58.33</td>
<td>71.21</td>
<td><strong>77.27</strong></td>
</tr>
<tr>
<td>20</td>
<td>73.33</td>
<td>74.17</td>
<td><strong>78.33</strong></td>
<td>57.50</td>
<td>67.50</td>
<td>76.67</td>
<td>60.83</td>
</tr>
</tbody>
</table>

Table 4.7: Accuracy (%) of MSRActionPairs with setting [40, 40, 20].

<table>
<thead>
<tr>
<th>#Train</th>
<th>DsNTF</th>
<th>SSM</th>
<th>$\lambda_1 = 0$</th>
<th>$\Phi(\cdot)$</th>
<th>$\lambda_1 = 0.1$</th>
<th>$\lambda_1 = 1$</th>
<th>$\lambda_1 = 10^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>55.95</td>
<td><strong>83.33</strong></td>
<td>70.83</td>
<td>75.00</td>
<td>51.79</td>
<td>58.93</td>
<td>56.55</td>
</tr>
<tr>
<td>17</td>
<td>64.74</td>
<td><strong>86.54</strong></td>
<td>64.10</td>
<td>68.59</td>
<td>51.28</td>
<td>30.77</td>
<td><strong>86.54</strong></td>
</tr>
<tr>
<td>18</td>
<td>45.14</td>
<td><strong>85.42</strong></td>
<td>25.00</td>
<td>76.39</td>
<td>77.08</td>
<td>57.64</td>
<td>81.94</td>
</tr>
<tr>
<td>19</td>
<td>84.09</td>
<td>81.82</td>
<td>42.42</td>
<td><strong>87.88</strong></td>
<td>84.09</td>
<td>68.94</td>
<td>74.27</td>
</tr>
<tr>
<td>20</td>
<td>57.50</td>
<td>80.83</td>
<td>64.17</td>
<td>67.50</td>
<td><strong>90.00</strong></td>
<td>83.33</td>
<td>75.83</td>
</tr>
</tbody>
</table>

4.6.5.1 Penalty Parameters $\lambda_p$

We consider two problems to illustrate the role of $\lambda_p$, i.e., (1) sub-action problem, (2) multi-modality problem, by 10-fold multi-subject tests with RGB-depth as the Train-Test setting. Here we evaluate the settings: $\lambda_1 = 1$, $\lambda_2 \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0, 10, 100, 1000\}$ and $\lambda_3 = \lambda_2 + 0.1$. Fig. 4.11(a) illustrates the results of the first problem, and we can see that higher accuracy is obtained when $\lambda_2$ and $\lambda_3 \in [0.2, 1.0]$, which outperforms the performance when $\lambda_2 = 0$. Fig. 4.11(b) illustrates the similar trends. The result indicates that key frames selection helps to improve the performance of recognition.

4.6.5.2 Objective Function Value (OFV)

For the subspace dimension setting [10, 10, 5], we calculate the OFV of each mode as shown in Fig. 4.12. We can see that OFVs of all modes become stable within 10 iterations, which indicates that DNTF model converges well.
CHAPTER 4. DEEP NON-NEGATIVE TENSOR FACTORIZATION FOR TIME ALIGNMENT

Figure 4.11: Accuracy with different penalty factors $\lambda_2$ and $\lambda_3$ for left: sub-action and right: multi-modality challenges. We can see that the accuracy under $\lambda_2, \lambda_3 > 0$ is better than that of $\lambda_2 = 0$.

4.6.5.3 Signal-to-Noise Ratio (SNR)

Root relative squared error (RRSE) is used to reveal how the tensor decomposition is affected by mode-n dimensions. Specifically, given a tensor $\mathcal{X}$ after non-negative decomposition by our model, we have $\mathcal{X} = \mathcal{W} \mathcal{H}$. Then, we add different levels of Gaussian noises (30dB∼-5dB), so the decomposition with contamination is $\tilde{\mathcal{X}} = \tilde{\mathcal{W}} \tilde{\mathcal{H}}$, where $\tilde{\mathcal{H}}$ is the perturbed tensor with noises. Then, the RRSE is defined as:

$$RRSE = \frac{\|\mathcal{H} - \tilde{\mathcal{H}}\|_F}{\|\mathcal{H}\|_F}. \quad (4.26)$$

Fig.4.13 shows the RRSE under different dimensions, with 96 and 192 samples, respectively. We can see that increasing dimensions yield higher RRSE in most cases while smaller dimensions lead to lower RRSE. This consists with the phenomenon of higher accuracy under small dimensions shown in Fig. 4.6.

4.6.5.4 Time Complexity Comparison

We compare the running time of two tensor decomposition methods in our model: tucker decomposition and tensor-train (TT) in the second layer. Fig.4.14 shows the running time of them under different data scales. The results justify that using TT for DNTF improves the running time compared with the tucker decomposition.

4.7 Conclusions

In this paper, we proposed a discriminant deep tensor decomposition method for sparse temporal alignment applicable to sub-action, multi-subject, multi-modality problems. First, we temporally aligned the key frames of intra-class action sequences by sparse learning, which significantly
Figure 4.12: Objective function value (OFV) of our model on MSRAActivityPairs dataset. Left: mode-1,2 OFV. Right: mode-3 OFV. The OFVs of all modes will not change after a few iterations.

Figure 4.13: RRSE under various noise levels (SNR). Left: dataset size is 96. Right: dataset size is 192.

Figure 4.14: Running time comparisons of TT and tucker decomposition given different numbers of training data.

reduces redundancy and overlapping of the intra-class samples and reduces the diversity caused by different scales, subjects and modalities. Second, to find discriminant representations for the sparse temporal alignment problem, we designed a deep NTF (DNTF) mechanism to extract discriminant features, with NTF and TT decomposition as the building blocks for the two layers. In the experiment section, we conducted extensive experiments on both synthetic and realistic datasets to demonstrate the effectiveness of our method. In addition, we analyzed key parameters for a better understanding of the proposed model.
Chapter 5

Conclusion

In this study, we focus on the recent advance of human action recognition methods, presented action data as high-order tensor to preserve spatiotemporal information, and proposed three novel algorithms for different problems in action recognition task. In the following chapters, we detailed our new algorithms, and discussed different problems existing in action recognition task.

In Chapter 2, we proposed a novel framework for tensor subspace learning. We utilized the low-rank constraint to find the subspace dimension automatically. We calculated the mode-n projection matrix by low-rank learning, and the rank is set to be subspace dimension. There are two benefits of our model. First, the number of dimension would not cause redundancy or insufficiency for recognition in the subspace. Second, we integrated discriminant information in our model for recognition in the subspace.

In Chapter 3, we proposed a novel framework for missing-modality problem by transfer learning. Suppose in the target domain, there are RGB data for training but depth data for testing, which would not get good performance because of different modalities. We call it missing-modality problem. Given an auxiliary source domain containing both RGB and depth data, in the training phase we explored the correlation of RGB and depth modalities in the source domain, which is transferred to the target domain to help recover additional ”missing“ depth information. In this way, we aimed to improve accuracy of recognizing missing modality data in the target domain. Experiments demonstrated our method could improve accuracy compared with other transfer learning methods.

In Chapter 4, we proposed a novel framework for intra-class diversity problems, e.g., multi-subject, multi-modality and sub-action. We selected key frames of two intra-class action sequences, and aligned them by the key frames. This procedure aims to reduce the diversity within class. Then we found a tensor subspace by deep Non-Negative Tensor Factorization to extract discriminative
feature for recognition. Experiments demonstrated our method could improve accuracy compared with other subspace alignment methods.

In my future research, I will continue work on time series data like video, natural language and document with different problems. For example, large spatiotemporal corrupted action videos for recognition problem could be solved by deep Auto-Encoder, and temporal prediction or recognition problem can be solved by Long-Short-Term-Memory (LSTM).
Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


