Exploring Intrinsic Structures from Samples: Supervised, Unsupervised, and Semisupervised Frameworks

WANG, Huan

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Philosophy in Information Engineering

Supervised by

Prof. Xiaoxuan Tang

©The Chinese University of Hong Kong August 2007

The Chinese University of Hong Kong holds the copyright of this thesis. Any person(s) intending to use a part or whole of the materials in the thesis in a proposed publication must seek copyright release from the Dean of the Graduate School.
Sample sets we are facing today are growing larger and larger, yet the processing and analysis of these data are often difficult due to the curse of dimensionality as well as the high computational cost involved. Then how to derive the information conveyed by huge number of samples becomes a challenging problem in computer vision. Fortunately, these high dimensional data usually lie approximately on an underlying compact low dimensional manifold, which may turn the problem tractable. Substantive works have been devoted to unveiling the intrinsic structure of the manifold data, among which the popular ones include ISOMAP, LLE and Laplacian Eigenmap. In computer vision applications, other kind of feature structures also exist, for example, geometric structures. In the problem of feature matching and object recognition, the geometric locations of features can be utilized to boost the algorithm performance and great improvements have been achieved.

Generally supervised algorithms can achieve a higher accuracy compared with the unsupervised ones due to the pertinence obtained from the training process. In real applications we may not be able to obtain enough training labels while the unlabeled samples may still contribute to training. Inspired by this
view, semisupervised frameworks are investigated a lot recently. Combining the labeled and the unlabeled points, semisupervised algorithms can produce more accurate output especially when the sample labels are rare.

In this thesis we first propose a novel iterative algorithm for trace ratio optimization in the supervised discriminant analysis algorithms. Consequently, the optimization process is introduced to the supervised tensor subspace learning problem and a convergent solution is deduced. We then move on to the semisupervised framework and design a transductive regression algorithm on multi-class data. For unsupervised learning, we also propose a spectral embedding algorithm that maximizes the global variance and at the same time keeps the local structures for image clustering. All these algorithms utilize the graph to represent the sample manifolds and manifold structures are further exploited for the guidance of classification as well as regression. Finally, a correspondence propagation framework (RCP) for object registration is proposed and the geometric structures of feature points are explored to facilitate the matching process. Inspired by the semisupervised learning algorithms, RCP can incorporate the human interaction in the correspondence searching process and combined with the automatic feature matching methods, it can also work in an unsupervised manner.
Acknowledgement

First I would like to give my sincere gratitude to my supervisor, Prof. Xiaoou Tang, who was guiding and supporting me during the past two years. His valuable comments and suggestions are of great help not only in the thesis writing but also in the way the research is carried out. I am also grateful to my co-supervisor Prof. Jianzhuang Liu, for his assistance in the thesis preparation.

This thesis could never have been written without the support of Dr. Shuicheng Yan, a great mentor and a good friend of mine. His intuition and deep insights provide me a never ending stream of ideas, suggestions and comments.

Special thanks to my dear friend Dahua Lin for his technical assistance in both programming and thesis writing. I benefit a lot from the inspiring discussion with him.

Finally, I would like to thank my parents for their constant encouragement to keep me in high spirits.
This work is dedicated to my parents and friends who are encouraging and supporting me all along.
Contents

Abstract i

Acknowledgement iii

1 Introduction 1
   1.1 Learning Frameworks ......................... 1
   1.2 Sample Representation ....................... 3

2 Background Study 5
   2.1 Tensor Algebra ............................... 5
       2.1.1 Tensor Unfolding (Flattening) .......... 6
       2.1.2 Tensor Product .......................... 6
   2.2 Manifold Embedding and Dimensionality Reduction .............. 8
       2.2.1 Principal Component Analysis (PCA) ... 9
       2.2.2 Metric Multidimensional Scaling (MDS) .... 10
       2.2.3 Isomap ................................... 10
       2.2.4 Locally Linear Embedding (LLE) ......... 11
       2.2.5 Discriminant Analysis .................... 11
       2.2.6 Laplacian Eigenmap ....................... 14
       2.2.7 Graph Embedding: A General Framework .... 15
       2.2.8 Maximum Variance Unfolding .............. 16

3 The Trace Ratio Optimization 17
   3.1 Introduction .................................. 17
3.2 Dimensionality Reduction Formulations: Trace Ratio vs. Ratio Trace ............................................. 19
3.3 Efficient Solution of Trace Ratio Problem ............ 22
3.4 Proof of Convergence to Global Optimum .......... 23
  3.4.1 Proof of the monotonic increase of $\lambda^n$ .... 23
  3.4.2 Proof of $V^n$ convergence and global optimum for $\lambda$ .................................................. 24
3.5 Extension and Discussion .............................. 27
  3.5.1 Extension to General Constraints .......... 27
  3.5.2 Discussion ...................................... 28
3.6 Experiments ........................................... 29
  3.6.1 Dataset Preparation .......................... 30
  3.6.2 Convergence Speed ........................... 31
  3.6.3 Visualization of Projection Matrix .......... 31
  3.6.4 Classification by Linear Trace Ratio Algorithms with Orthogonal Constraints ........... 33
  3.6.5 Classification by Kernel Trace Ratio algorithms with General Constraints ............. 36
3.7 Conclusion ............................................. 36

4 A Convergent Solution to Tensor Subspace Learning 40
  4.1 Introduction ......................................... 40
  4.2 Subspace Learning with Tensor Data .............. 43
    4.2.1 Graph Embedding with Tensor Representa-
      tion ............................................. 43
    4.2.2 Computational Issues ........................ 46
  4.3 Solution Procedure and Convergence Proof ....... 46
    4.3.1 Analysis of Monotonous Increase Property 47
    4.3.2 Proof of Convergence ......................... 48
  4.4 Experiments ........................................... 50
    4.4.1 Data Sets ................................... 50
    4.4.2 Monotonicity of Objective Function Value 51
6.5.2 Synthetic Data: Three-class Cyclones . . 83
6.5.3 Human Age Estimation . . . . . . . . . . 84
6.6 Conclusions . . . . . . . . . . . . . . . . . . . . 86

7 Correspondence Propagation 88
7.1 Introduction . . . . . . . . . . . . . . . . . . . . 88
7.2 Problem Formulation and Solution . . . . . . . 92
  7.2.1 Graph Construction . . . . . . . . . . . . . . 92
  7.2.2 Regularization on categorical Product Graph 93
  7.2.3 Consistency in Feature Domain and Soft
       Constraints . . . . . . . . . . . . . . . . . . . . 96
  7.2.4 Inhomogeneous Pair Labeling . . . . . . . . 97
  7.2.5 Reliable Correspondence Propagation . . . 98
  7.2.6 Rearrangement and Discretizing . . . . . . . 100
7.3 Algorithmic Analysis . . . . . . . . . . . . . . . . 100
  7.3.1 Selection of Reliable Correspondences . . . 100
  7.3.2 Computational Complexity . . . . . . . . . . . 102
7.4 Applications and Experiments . . . . . . . . . . . 102
  7.4.1 Matching Demonstration on
       Object Recognition Databases . . . . . . . . . . 103
  7.4.2 Automatic Feature Matching on
       Oxford Image Transformation Database . . . . 104
  7.4.3 Influence of Reliable Correspondence Num-
       ber . . . . . . . . . . . . . . . . . . . . . . . . . 106
7.5 Conclusion and Future Works . . . . . . . . . . . 106

8 Conclusion and Future Work 110

Bibliography 113
List of Figures

2.1 Tensor: multi-dimensional (or multi-way) arrays of components. ............................... 6
2.2 Tensor unfolding: the 3-nd order tensor can be unfolded in 3 ways to derive matrices comprising its mode-1, mode-2 and mode-3 vectors. ............................ 7
2.3 Demonstration of Graph construction of Graph Embedding Framework. (a) intrinsic graph. (b) penalty graph. .................................................. 15

3.1 $\|V^n - V^{n-1}\|$ error vs. iteration number, and the trace difference $|Tr[V^{nT}(S_p - \lambda^n S_t)V^n]|$ (implies the error between $V^n$ and the optimum) vs. iteration number. (a-b) FERET database, and (c-d) CMU PIE database. ............................................ 30
3.2 Visualization of the projection matrix $W$ of PCA, ratio trace based LDA, and trace ratio based LDA (ITR) on the FERET database. ......................... 32
3.3 Recognition error rates over different dimensions. The configuration is N3T7 on the CMU PIE database. For LDA and MFA, the dimension of the preprocessing PCA step is $N-Nc$. .......................... 34
3.4 Testing classification errors on three UCI databases for both linear and kernel-based algorithms. Results are obtained from 100 realizations of randomly generated 70/30 splits of data. .................. 35
4.1 The value of the objective function (4.7) vs. iteration number. (a) USPS database (b) ORL database, and (c) CMU PIE database. Here the traditional method means the solution procedure based ratio trace optimization.

4.2 The step difference of the projection matrices vs. iteration number. (a,d) USPS database, (b,e) ORL database, and (c,f) CMU PIE database.

4.3 Recognition error rate (%) vs. iteration number. (a) ORL database(G4P6), and (b) CMU PIE database(G4P6).

5.1 Embedding Visualization of 3 Clusters

5.2 Some clustering results from 6 clusters

5.3 Generalization capability in the cases with different percentages of training samples. The cluster number is 10.

6.1 Demonstration of the inter-class relations. Labeled samples are marked by ‘+’ and the inter-manifold relations are indicated by the dashed lines. The manifolds are aligned and the sample labels are propagated across manifolds to pilot the regression.

6.2 Demonstration of reinforced landmark correspondence. The similarity score of $a_1$ and $b_2$ is reinforced by the scores of neighbor pairs.

6.3 Regression on the nonlinear Two Moons Dataset. (a) Original Function Value Distribution. (b) Traditional Graph Laplacian Regularized Regression (separate regressors for different classes). (c) Two Class TRIM. (d) Two Class TRIM on RKHS. Note the difference in the area indicated by the rectangle.
6.4 Regression on Cyclone Dataset: (a) Original Function Values. (b) Traditional Graph Laplacian Regularized Regression (separate regressors for different classes). (c) Three Class TRIM. (d) Three Class TRIM on RKHS. 83

6.5 Class Distribution of the Cyclone Dataset 85

6.6 TRIM vs traditional graph Laplacian regularized regression for the close set evaluation on YAMAHA database. 86

6.7 Open set evaluation for the kernelized regression on the YAMAHA database. (a) Regression on the training set. (b) Regression on out-of-sample data. 87

7.1 Flowchart of correspondence propagation from reliable correspondence priors for feature matching. 91

7.2 Demonstration of categorical graph product. 94

7.3 The number of correct matches v.s. the number of automatically selected reliable correspondences on the first two images of Graffiti database. 104

7.4 Feature matching results on Caltech 101 and ETH-80 example images. The first row of each figure is the performance of (a) semi-supervised matching by manual pairwise correspondence labeling, (b) semi-supervised matching of obscure correspondence guidance, and (c) automatic version of our algorithm; and the second row demonstrates the baseline results. The correspondence number of the two figures within the same column is the same. 108
7.5 Automatic feature matching score on the Oxford real image transformation dataset. The transformations include viewpoint change ((a) Graffiti and (b) Wall sequence), image blur ((c) bikes and (d) trees sequence), zoom and rotation ((e) bark and (f) boat sequence), illumination variation ((g) leuven) and JPEG compression ((h) UBC).
List of Tables

3.1 Recognition error rates (%) of Ratio Trace + Kernel Discriminant Analysis (RKDA), Trace Ratio + Kernel Discriminant Analysis (ITR-KDA), Ratio Trace + Kernel MFA (RKMFA), and Trace Ratio + Kernel MFA (ITR-KMFA) on the three face databases. .......................... 33

3.2 Recognition error rates (%) of PCA, PCA \((N-N_c)+\)Ratio Trace based LDA (RLDA), PCA \((N-N_c)+\)Trace Ratio based LDA (ITR-LDA), PCA \((N-1)+\)Ratio Trace based LDA (RLDA2), PCA \((N-1)+\)Trace Ratio based LDA (ITR-LDA2), PCA \((N-N_c)+\)Ratio Trace based MFA (RMFA), PCA \((N-N_c)+\)Trace Ratio based MFA (ITR-MFA), PCA \((N-1)+\)Ratio Trace based (RMFA2), PCA \((N-1)+\)Trace Ratio based MFA (ITR-MFA2), and the method without dimensionality reduction on the three face databases. Note that the boldtype numbers are those with the best classification accuracies for LDA and MFA respectively. ...... 39

4.1 Recognition error rates (%) on the ORL database. 53

4.2 Recognition error rates (%) on the PIE database. 53

5.1 Clustering accuracies of MUC, NCut, PCA-Kmeans and K-Means in the cases with different cluster numbers. .................. 64
Chapter 1

Introduction

Summary

This work is mainly based on my conference papers [52] [51] [49] [50] [57] [58].

Recent decades observe a tremendous growth of information and the real-world applications today are facing mountains of sample data. The data interpretation remains a tough issue. How to represent these data, how to express the complex relations among the data samples, and how to exploit the information delivered by those samples remain to be quite active areas these days.

1.1 Learning Frameworks

Machine learning is becoming a trend in computer vision these days. Substantive vision topics are more or less relevant to the learning problem, such as face recognition, object recognition, object tracking, pose estimation and so on. In fact, the human vision system also shares a close relationship with the human learning process. Thus it is quite natural to introduce machine learning algorithms into the computer vision area. Traditional
learning algorithms can be categorized into two folds: unsupervised learning and supervised learning. Unsupervised learning algorithms simply give the output from the incoming data and no training process is needed. Unsupervised algorithms are generally very fast and easy to implement, while their accuracy is limited due to the lack of pertinency with respect to the given samples. Supervised learning algorithms have two phases. First the algorithm is not determined and some parameters as well as algorithmic structures remain unknown. Thus a training set must be provided to solve those unknown parameters in the algorithm. Then, for the testing data, the algorithm gives predictions according to both the incoming samples and the parameters learnt from training set. Since the parameters and algorithmic structures are derived from the training set rather than fixed, the algorithm is more flexible and thus is supposed to be more applicable in the prediction of incoming testing set on the assumption that the testing samples share similar distribution with the training samples.

One issue for the supervised learning algorithms is that in real applications we may not be able to obtain enough labeled training data. The supervised algorithms tend to get overfit when the training set is not large enough. Moreover, the high dimensionality of the sample data we are facing today also deteriorates the performance of supervised algorithms over the small training set. To fix this problem, recently the semisupervised learning framework evolved, which allows the existence of unlabeled data in the training set. Semisupervised framework exploits the knowledge of the marginal distributions from the unlabeled samples and at the same time utilizes the label information provided by the labeled samples. Thus it has the potential to produce a more reasonable estimation of the incoming samples especially when the training labels are rare.
1.2 Sample Representation

To facilitate the analysis for multi-modality data, recently the tensor based representation has been investigated broadly. DATER [59] introduces the tensor representation to the discriminant analysis problem and explores the interrelation among different dimensions. The lower-dimensional discriminative subspaces are derived for feature selection. Two-Dimensional Linear Discriminant Analysis (2DLDA) [63] concentrates on a special case of tensor, i.e., 2nd-order tensor. Similar to all the discriminant analysis algorithm, the objective of 2DLDA also turns out to be a trace ratio optimization problem and the objective is optimized iteratively from one dimension to another.

To capture the intrinsic manifold structure in the sample data, graph theory has been developed and broadly utilized. Tensor Subspace Analysis (TSA) [26] extends the idea of 2DLDA and introduces the manifold representation into tensor subspace learning. Integrating the graph theory for the manifold learning and the tensor representation, TSA achieves a higher accuracy compared to the traditional discriminant analysis algorithms in the real-world face data sets. The problem of TSA lies in the fact that the objective is still a trace ratio problem and generally no closed form solution exists. Instead, TSA turns to solve a ratio trace problem and the deduced optimization process is not convergent. In this thesis, we propose a novel iterative algorithms that directly optimize the trace ratio and the convergence of our algorithm is also proved. Experimental results demonstrate the superiority of our proposed algorithm against the state-of-the-art algorithms.

We are not confined in the supervised framework and manifold structures. The concept of manifold and the corresponding graph representation can be also be used in the semisupervised regression, correspondence searching as well as unsupervised
data clustering. Besides the manifolds, other data structures, such as geometric distributions can also be utilized to boost the algorithmic performance in computer vision applications. We also present a Reliable Correspondence Propagation (RCP) framework that utilizes both geometric structures and feature agreements to guide the image registration. What’s more, combined with simple automatic approaches, the proposed framework can easily switch between the semisupervised and unsupervised manners.

The structure of this thesis is as follows: first we give a general review of the tensor algebra, discriminant analysis and manifold learning in the introduction chapter. The iterative procedure for the trace ratio solution is systematically described in Chapter 3 [52]. Chapter 4 [51] further employs the trace ratio optimization procedure in the tensor subspace learning and proposes a convergent solution. Semisupervised regression and unsupervised data clustering are presented in Chapter 5 [49] and Chapter 6 [48]. Finally the Reliable Correspondence Propagation framework is proposed in Chapter 7. We conclude the thesis in Chapter 8.

End of chapter.
Chapter 2

Background Study

Summary

This chapter gives an introduction of preliminary notations of tensor algebra and a general review of manifold embedding algorithms. The concepts and introduction in this chapter is mainly based on the works of [59] [47] [27] and [40]

2.1 Tensor Algebra

Vector representation is broadly used in face recognition. Facial images are first vectorized and the analysis as well as recognition process are then applied on the vector space. In real-world applications, the construction of natural image can be affected by multifarious factors, such as illumination, scene structure and imaging. For example, in face recognition applications, the human facial images vary with respect to lighting conditions, pose variations, expression changes and so on. To dig through the intrinsic relationship among different factors, there is a need to fuse or arrange the data in an ordered way so that further analysis can be processed. Tensor is a natural representation for
the fusion and arrangement of the sample data. Furthermore, the traditional vectorized representation can be regarded as a special case of tensor (1D tensor).

Tensors, known as multilinear mapping over a set of vector spaces, are high order generalization of vectors (first order tensor) and matrices (second order tensor). Tensor can also be regarded as $n$-way arrays or multidimensional matrix or $n$-mode matrix. The order of a tensor $A \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_n}$ is $n$. Figure 2.1 demonstrates the elementary organization of a 3-order tensor.

2.1.1 Tensor Unfolding (Flattening)

Tensor unfolding, also called tensor flattening, is the process to convert a 3 or more order tensor into a $2^n d$ order tensor, i.e., matrix. Specifically, the tensor unfolding fixes one dimension of the tensor and place the tensor elements along the expansion of the other dimensions. The unfolding is named $n$-way unfolding if the $n^{th}$ dimension is fixed during the rearrangement.

2.1.2 Tensor Product

Assume tensor $A$ is an $n^{th}$ order tensor with $A \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_n}$. The dimension of tensor $B$ is the same as that of tensor $A$. The
CHAPTER 2. BACKGROUND STUDY

Figure 2.2: Tensor unfolding: the 3-rd order tensor can be unfolded in 3 ways to derive matrices comprising its mode-1, mode-2 and mode-3 vectors.

inner product of then tensor $A$ and $B$ is defined as

$$< A, B > = \sum_{i_1=1, \ldots, i_n=1}^{m_1 \ldots m_n} A_{i_1 \ldots i_n} B_{i_1 \ldots i_n};$$ (2.1)

Consequently, tensor norm is defined as

$$\|A\| = \sqrt{< A, A >}.$$ (2.2)

A special case of the tensor norm is the Febenius norm $\|A\|_F$ of the 2\textsuperscript{nd}-order tensor, i.e., the matrix. The Euclidean distance between the tensor $A$ and $B$ is

$$d = \|A - B\|.$$ (2.3)

The mode-$k$ product of a tensor $A$ and a $m_k$ by $m'_k$ matrix $U$ is defined as

$$B = A \times_k U,$$ (2.4)

where

$$B_{i_1 \ldots i_{k-1},j,i_{k+1} \ldots i_n} = \sum_{i=1}^{m_k} A_{i_1 \ldots i_{k-1},i,i_{k+1} \ldots i_n} \times U_{i,j}, j = 1, \ldots, m'_k.$$ (2.5)
In terms of tensor unfolding, the tensor mode-$k$ product can be fulfilled using the matrix multiplication of the unfolded tensor $A_{(k)}$ and $U$, i.e.,

$$B_{(k)} = U A_{(k)},$$

(2.6)

where $A_{(k)}$ and $B_{(k)}$ are mode-$k$ unfolding of tensor $A$ and $B$ respectively.

Assume that the dimensions of tensor $A$, matrix $U$ and matrix $V$ satisfy the mode-$k$ tensor product, then the following properties of mode-$k$ tensor product hold:

1. $A \times_k U \times_l V = (A \times_k U) \times_l V$ (2.7)
   $$= (A \times_l V) \times_k U$$ (2.8)
   $$= A \times_l V \times_k U$$ (2.9)

2. $(A \times_l U) \times_l V = A \times_l (VU)$ (2.10)

### 2.2 Manifold Embedding and Dimensionality Reduction

Large scale and high dimensional data are ubiquitous in real-world applications, yet the processing and analysis of these data are often difficult due to the curse of dimensionality as well as the high computational cost involved. Usually, these high dimensional data lie approximately on an underlying compact low dimensional manifold, which may turn the problem tractable. A topological space $M \subseteq \mathbb{R}^m$ is a manifold if for every $x \in M$, an open set $O \subset M$ exists such that: 1) $x \in O$, 2) $O$ is homeomorphic to $\mathbb{R}^n$, and 3) $n$ is fixed for all $x \in M$. The fixed $n$ is referred to as the dimension of the manifold $M$.

Revealing the inner manifold structures of the high dimensional data can help us to carry out dimensionality reduction. Substantive methods have been proposed on this issue.
2.2.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a popular statistical analysis method. The aim of PCA is to find the directions, i.e., the bases of a subspace, that best preserve the energy of the projected data, or, minimize the reconstruction error. Let $x_i$ denote the $i^{th}$ sample vector and $\bar{x}$ is the sample mean. Utilizing matrix representations, we assume $X$ to be the column sample matrix and $\bar{X}$ denote the matrix constructed by the sample mean. The objective of PCA can be expressed as two equivalent forms:

$$\hat{W} = \arg \max_{W^TW=I} \sum_i \|W^T(x_i - \bar{x})\|^2$$

$$= \arg \max_{W^TW=I} \|W^T(X - \bar{X})\|^2,$$  \hspace{1cm} (2.11)

or,

$$\hat{W} = \arg \min_{W^TW=I} \sum_i \|x_i - \bar{x} - WW^T(x_i - \bar{x})\|^2$$

$$= \arg \min_{W^TW=I} \|X - \bar{X} - WW^T(X - \bar{X})\|^2,$$ \hspace{1cm} (2.12)

where $\|A\|^2$ is the Fubini norm of matrix $A$. Equation 2.11 can be expressed as:

$$\hat{W} = \arg \max_{W^TW=I} Tr(W^T(X - \bar{X})(X - \bar{X})^TW)$$

$$= \arg \max_{W^TW=I} Tr(W^TCW),$$ \hspace{1cm} (2.13)

where $Tr(A)$ denotes the trace of matrix $A$ and $C$ is the covariance matrix of samples. Thus the optimization of the objective can be solved by spectral methods.

PCA can be used to uncover the linear hyperplane of the high dimensional data, while it may not work well if the data manifold has some nonlinear structures. One solution is to utilize the 'kernel trick' to derive a kernelized version of PCA. The issue arising from the Kernel PCA (KPCA) is that our knowledge about the the Reproducing Kernel Hilbert Space (RKHS)
is rather limited and the selection of kernel parameters remains a problem.

### 2.2.2 Metric Multidimensional Scaling (MDS)

Metric Multidimensional Scaling aims at a low dimensional representation that preserves the pairwise inner product. Denote the output samples as \( \{y_i\} \), the objective can be expressed as:

\[
\arg \min_y \sum_{ij} (x_i \cdot x_j - y_i \cdot y_j)^2.
\] (2.14)

The solution to 2.14 can be obtained from spectral decomposition of the Gram matrix, i.e.,

\[
G_{ij} = x_i \cdot x_j.
\] (2.15)

Denote the top \( m \) eigenvectors of \( G \) as \( \{\phi_d\}_{d=1}^m \) and the corresponding eigenvalues as \( \{\lambda_d\}_{d=1}^m \). The output of MDS algorithm is a concatenation of weighted eigenvectors, i.e., \( y_{id} = \sqrt{\lambda_d} \phi_{di} \).

### 2.2.3 Isomap

Different from PCA which minimizes the reconstruction error, Isomap seeks a low dimensionality representation of the data that best preserves the pairwise geodesic distance along the data manifold. To represent the relations of sample points, a graph is first constructed using \( k \)-nearest neighbors. The graph edges are weighted based on the Euclidean distance between the nearest neighbors. Pairwise geodesic distances \( S \) are then calculated along the graph using the Djikstra’s algorithm. Finally the Gram matrix of inner products is derived from the pairwise geodesic distance matrix \( S \) by the transformation \( G = -(I - uu^T)S(I - uu^T)/2 \), where \( I \) is the \( n \times n \) identity matrix and \( u = \frac{1}{n}(1, 1, ..., 1)^T \). The Gram matrix derived is then fed as input to MDS and the low dimensional representation is obtained.
2.2.4 Locally Linear Embedding (LLE)

The objective of Locally Linear Embedding (LLE)\cite{39} is to preserve the local linear structure of nearby input patterns. LLE is also based on the graph constructed to describe the pairwise relations of samples along the manifold. The graph can be constructed either by $k$-nearest neighbors or by connecting all neighbors within a ball of fixed radius. Then the reconstruction coefficients are obtained by minimizing the reconstruction errors:

$$W = \arg \min_{\sum_j w_{ij} = 1} \sum_i |x_i - \sum_j w_{ij} x_j|^2. \quad (2.16)$$

Finally the low dimensional representation can be obtained by solving the following quadratic optimization:

$$\Phi(Y) = \sum_i |y_i - \sum_j w_{ij} y_j|^2, \quad (2.17)$$

where the reconstruction coefficients are derived from 2.16. The optimization, also called optimal embedding, can be found by computing the bottom vectors of the matrix $M = (I - W)^T (I - W)$.

2.2.5 Discriminant Analysis

Linear Discriminant Analysis

Linear discriminant analysis tries to maximize the between class scatter and meanwhile minimize the within class variance through a linear projection into a certain subspace, i.e.,

$$W^* = \min_W \frac{\sum_{i=1}^N \|W^T x_i - W^T \bar{x}_c\|^2}{\sum_{c=1}^{N_c} n_c \|W^T \bar{x}_c - W^T \bar{x}\|^2}, \quad (2.18)$$

where $\bar{x}_c$ is the mean of samples belonging to the $c$-th class, $\bar{x}_{c_i}$ is the sample mean of the class corresponding to $x_i$ and $\bar{x}$ is the mean of all samples.
2.18 can be formulated as trace ratio optimization problem, i.e.,

\[
\arg \max_W \left\{ \frac{\text{Tr}(W^T S_b W)}{\text{Tr}(W^T S_w W)} \right\},
\]

(2.19)

where \( S_b \) and \( S_w \) are the between-class scatter matrix and within-class scatter matrix respectively.

**Kernel Discriminant Analysis**

Kernel trick can be easily used in the Linear Discriminant Analysis framework for an extension to the nonlinear discriminant analysis. The projection to be derived from the high dimensional kernel feature space can be represented as a linear combination of the mapped training features, i.e.

\[
W = \sum_i \alpha_i \phi(x_i),
\]

or

\[
W = \Phi \times A
\]

Where \( A \) is the coefficient matrix:

\[
A = \begin{pmatrix}
\alpha_{11} & \alpha_{12} & \ldots & \alpha_{1m} \\
\alpha_{21} & \alpha_{22} & \ldots & \alpha_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha_{n1} & \alpha_{n2} & \ldots & \alpha_{nm}
\end{pmatrix}
\]

and

\[
\Phi = [\phi(x_1), \phi(x_2), \ldots, \phi(x_n)]
\]

Here \( n \) is the number of training sample points and \( m \) is the dimension to be reduced.

The projected data features are then represented as:

\[
\Phi' = W^T \times \Phi = A^T \times \Phi^T \times \Phi = A^T \times K
\]
The $S_w$ in the mapped feature space is:

$$S_w = \sum_{i=1}^{N} (\phi(x_i) - \overline{\phi(x)})^T(\phi(x_i) - \overline{\phi(x)}) = \Phi(I - \sum_{c=1}^{N_c} \frac{1}{n_c} e^c e^T) \Phi^T,$$

and $S_b$:

$$S_b = C - S_w = \Phi(I - \frac{1}{N} ee^T) \Phi^T - \Phi(I - \sum_{c=1}^{N_c} \frac{1}{n_c} e^c e^T) \Phi^T = \Phi\left(\sum_{c=1}^{N_c} \frac{1}{n_c} e^c e^T - \frac{1}{N} ee^T\right) \Phi^T,$$

(2.20)

where $e$ is an $N$ dimensional vector with $e = [1, 1, ... , 1]^T$ and $e^c$ is an $N$ dimensional vector with $e^c(i) = 1$, if $c = l_i; 0$, otherwise.

The Fisher quotient is then converted into:

$$J(A) = \frac{A^T \times K(I - \sum_{c=1}^{N_c} \frac{1}{n_c} e^c e^T) K^T \times A}{A^T \times K(\sum_{c=1}^{N_c} \frac{1}{n_c} e^c e^T - \frac{1}{N} ee^T) K^T \times A}$$

The constraint $W^T \times W = I$ now is changed to:

$$A^T \Phi^T \times \Phi A = A^T \times K \times A = I$$

All in all, the problem is converted into a constrained generalized Rayleigh Quotient problem:

$$max_A J(A)$$

with respect to the constraint:

$$A^T K A = I$$
2.2.6 Laplacian Eigenmap

Laplacian Eigenmap exploits the Graph Laplacian of the constructed graph and converts the manifold embedding problem into a minimization of

\[ Tr(Y^TLY), \]
\[ \text{w.r.t. } Y^TDY = I, \]  \hspace{1cm} (2.21)

where \( D \) is diagonal weight matrix whose entries are column sums of \( W \), i.e., \( d_{ii} = \sum_j w_{ji} \) and \( L \) is the Graph Laplacian with \( L = D - W \). By minimizing the objective 2.21, the algorithm imposes a smoothness condition over the data manifold on the possible solutions.

The low dimensional embedding from the eigenvectors corresponding to the lowest eigenvalues of the generalized eigenvalue decomposition:

\[ Ly = \lambda Dy, \]  \hspace{1cm} (2.22)

Locality Preserving Projections

Locality Preserving Projections (LPP) is the linearized version of Laplacian Eigenmap. It is introduced to solve the out-of-sample problem. Suppose the low dimensional embedding can be expressed using linear projections from the original data, i.e., \( y^T = a^T X \). The objective of Laplacian Eigenmap is then converted into:

\[ \arg \min_{a^TXDX^Ta=1} a^T XLX^Ta. \]  \hspace{1cm} (2.23)

Kernel trick can also be used to give a nonlinear extension, and the optimization turns out to be a generalized eigen-decomposition problem:

\[ KLK\alpha = \lambda KD\alpha, \]  \hspace{1cm} (2.24)

where \( K \) is the kernel Gram matrix and \( \alpha \) is the coefficient matrix to be solved.
2.2.7 Graph Embedding: A General Framework

Since the formulation of all the algorithms described above turns out to be a trace ratio optimization problem, a general framework, i.e., Graph Embedding [55], is proposed to incorporate all the above-mentioned algorithms. First two graphs, i.e., the intrinsic graph and penalty graph, are constructed based on different criterions according to different algorithms. Then the common objective is formulated as:

\[
y^* = \arg\min_{y^TBy = d} \sum_{i \neq j} ||y_i - y_j||^2W_{ij} = \arg\min_{y^TBy = d} y^T Ly, \quad (2.25)
\]

where \(d\) is a constant, \(L\) is the Laplacian matrix of the intrinsic graph and \(B\) is the constraint matrix defined to avoid a trivial solution of the objective function. \(B\) typically is a diagonal matrix for scale normalization and may also be defined as Laplacian matrix of the penalty graph.

From the graph embedding view, most popular dimensionality reduction algorithms are special version of graph embedding objective according to different intrinsic graphs, penalty graphs
or scale normalization items. And a series algorithms can also be regarded as the linearized, kernelized or tensorized version of the original algorithms.

### 2.2.8 Maximum Variance Unfolding

[53] proposes the Maximum variance unfolding (MVU) algorithm that ‘stretches’ the manifold by maximize the sample variance and at the same time preserves the local neighborhood distances and angles. Thus the problem is formulated as the following optimization problem:

\[
\text{maximize} \quad \sum_{ij} \| y_i - y_j \|^2,
\]

w.r.t.

(1) \quad \sum_i y_i = 0.

(2) \quad \| y_i - y_j \|^2 = D_{ij}

for all connected pairs, \hspace{1cm} (2.26)

where \( y_i \) is the embedded sample vector to be derived and \( D_{ij} = \| x_i - x_j \|^2 \) is the pairwise distance preserved between connected pairs. The optimization of 2.26 can be solved by semidefinite programming.

\[ \square \text{End of chapter.} \]
Chapter 3

The Trace Ratio Optimization

Summary

To explore the intrinsic structures of samples lying in high dimensional space, we are inclined to carry out dimensionality reduction before data processing, during which the structural information of samples is utilized to boost the performance. A large series of dimensionality reduction algorithms end up with a trace ratio optimization. Traditional algorithms turn to an approximation of ratio trace optimization procedure. In this chapter, we propose a novel algorithm that directly optimizes the trace ratio objective. We then carry out both theoretical and experimental evaluation on our algorithm.

3.1 Introduction

Variations in a set of high-dimensional data, such as images, often have an underlying low-dimensional structure that compactly characterizes the changes among these observations [3][45]. To uncover this low-dimensional structure of the data, dimensionality reduction has been an active research topic in computer
Yan et al. [60] claimed that most traditional algorithms for dimensionality reduction can be unified within a general framework, called Graph Embedding. This framework derives a low-dimensional feature space which preserves the adjacency relationship between different sample pairs in addition to constraints from scale normalization or a penalty graph [60]. Within this context, many algorithms for dimensionality reduction involve a search for a transformation matrix $W$ that maximizes a term $Tr(W^T S_p W)$ and at the same time minimizes another term $Tr(W^T S_l W)$, where matrices $S_p$ and $S_l$ are both positive semidefinite. The natural solution to these dual objectives is to pose a trace ratio optimization problem, namely $\max_W Tr(W^T S_p W) / Tr(W^T S_l W)$, which however does not have a closed-form solution. Generally, the trace ratio problem is often simplified into a more tractable one called the ratio trace problem: $\max_W Tr[(W^T S_l W)^{-1} (W^T S_p W)]$. The ratio trace problem can be efficiently solved with the generalized eigenvalue decomposition method [20]. However, its solution may deviate from the original objectives and suffers from the fact that it is invariant under any non-singular transformation, which may lead to uncertainty in subsequent processing such as classification and clustering.

In this work, we tackle the original trace ratio problem, and present a procedure to directly optimize the objective function $Tr(W^T S_p W) / Tr(W^T S_l W)$ by assuming that the column vectors of $W$ are unitary and orthogonal to each other. The procedure iteratively optimizes the objective function, and the projection matrix $W^n$ of the $n$-th step is obtained by solving a corresponding trace difference problem $\max_W Tr[W^T (S_p - \lambda^n S_l) W]$ where $\lambda^n$ is the trace ratio value computed from $W^{n-1}$. Therefore, the sub-problem in each step can be efficiently solved with the eigenvalue decomposition method [20]. A detailed proof is provided to justify that $\lambda^n$ will increase monotonically until reaching the
global optimum; the convergence of the projection matrix $W^n$ is also proven. In addition, this procedure is extended to handle the trace ratio problem with a more general constraint that $W^T CW = I$, where matrix $C$ is positive semidefinite. This extension provides the exact solution to kernel-based subspace learning algorithms with trace ratio formulations.

The rest of the chapter is organized as follows. Section 2 presents a detailed comparison of the trace ratio and ratio trace formulations for dimensionality reduction. Then, the iterative procedure to solve the trace ratio problem is introduced in Section 3, and Section 4 gives the convergence proof. The extension for more general constraints and its related works are introduced in Section 5. Experimental results are presented in Section 6, and we conclude this paper in Section 7.

### 3.2 Dimensionality Reduction Formulations: Trace Ratio vs. Ratio Trace

For a classification problem, assume that the training data are given as $\{x_i | x_i \in \mathbb{R}^m\}_{i=1}^N$, where $N$ is the number of training samples. The corresponding class labels of the samples are denoted as $\{c_i | c_i \in \{1, ..., N_c\}\}_{i=1}^N$, where $N_c$ is the number of classes, and the number of samples belonging to the $c$-th class is denoted as $n_c$. In practice, dimensionality reduction is in great demand owing to the fact that the effective information for classification often lies within a lower dimensional feature space.

A simple but effective way for dimensionality reduction is to find a matrix $W = [w_1, w_2, ..., w_d] \in \mathbb{R}^{m \times d}$ ($\text{Rank}(W) = d$, $\|w_k\| = 1$, $k = 1, 2, \ldots, d$) to transform the original high-dimensional data $x$ into a low-dimensional form $y \in \mathbb{R}^d$ (usually $d \ll m$) as

$$y = W^T x. \quad (3.1)$$

Many algorithms [3][45][60] with various motivations have
been proposed to find such a $W$. Yan et al. [60] claimed that
most dimensionality reduction algorithms can be unified into a
general framework, namely graph embedding which is described
as follows.

Let $G = \{X, S\}$ be an undirected weighted graph with vertex
set $X$ and similarity matrix $S \in \mathbb{R}^{N \times N}$. Each element of the real
symmetric matrix $S$ measures the similarity between a pair of
vertices. The diagonal matrix $D$ and the Laplacian matrix $L$ of a
graph $G$ are defined as $L = D - S$ and $D_{ii} = \sum_{j \neq i} S_{ij}$, $\forall i$. For a
specific dimensionality reduction algorithm, there may exist two
graphs, the intrinsic graph $G = \{X, S\}$ and the penalty graph
$G^p = \{X, S^p\}$ with $L^p = D^p - W^p$ and $D^p_{ii} = \sum_{j \neq i} S^p_{ij}$, $\forall i$. The
intrinsic graph characterizes data properties that the algorithm
favors and the penalty graph describes properties that the algo-

$$\arg\min_W \frac{\sum_{i \neq j} \|W^T x_i - W^T x_j\|^2 S_{ij}}{\sum_{i \neq j} \|W^T x_i - W^T x_j\|^2 S^p_{ij}}, \quad (3.2)$$

which can be further formulated in trace ratio form [60]:

$$\arg\max_W \frac{Tr(W^T X L^p X^T W)}{Tr(W^T X L X^T W)} = \frac{Tr(W^T S_p W)}{Tr(W^T S_l W)}, \quad (3.3)$$

where $S_p = X L^p X^T$ and $S_l = X L X^T$. Note that for the graph
embedding framework in [60], the denominator of (3.2) can also
be defined as a constraint for scale normalization, which will
also result in the trace ratio optimization problem as in the
latter part of (3.3).

Many popular dimensionality reduction algorithms, such as
Linear Discriminant Analysis (LDA) [3] and the nonparametric
algorithm Marginal Fisher Analysis (MFA) [60] (or similarly,
Local Discriminant Embedding [15]), can be formulated in the
above graph embedding framework. For example, LDA searches
for a subspace that minimizes intra-class scatter and at the same
time maximizes inter-class scatter:

$$W^* = \min_W \frac{\sum_{i=1}^N \| W^T x_i - W^T \bar{x}_c \|^2}{\sum_{c=1}^{N_c} n_c \| W^T \bar{x}_c - W^T \bar{x} \|^2}, \quad (3.4)$$

where $\bar{x}_c$ is the mean of samples belonging to the $c$-th class and
$\bar{x}$ is the mean of all samples. From the graph embedding point
of view in (3.3), the similarity matrices for the intrinsic and
penalty graphs of LDA are defined as

$$S_{ij} = \frac{\delta_{c_i,c_j}}{n_{c_i}}, i \neq j, \quad (3.5)$$

$$S_{ij}^p = \frac{1}{N} - S_{ij}, i \neq j, \quad (3.6)$$

where $\delta_{c_i,c_j}=1$ if $c_i=c_j$, and $\delta_{c_i,c_j}=0$ otherwise.

This optimization problem is typically nonconvex, and there
does not exist a closed-form solution for the general trace ratio
problem (3.3); hence such problems are often transformed into
the simpler yet inexact ratio trace problem, which is equivalent
to the determinant ratio problem [20]. For (3.3), the corresponding
ratio trace (determinant ratio) form is

$$W^* = \arg \max_W Tr[(W^T S_l W)^{-1}(W^T S_p W)] \quad (3.7)$$

$$= \arg \max_W \frac{|W^T S_p W|}{|W^T S_l W|} \quad (3.8)$$

which can be directly solved with the generalized eigenvalue de-
composition (GEVD) method:

$$S_p w_k = \tau_k S_l w_k \quad (3.9)$$

where $\tau_k$ is the $k$-th largest eigenvalue of the GEVD with the
Corresponding eigenvector $w_k$, and $w_k$ constitutes the $k$-th col-
umn vector of the matrix $W$.

**Remarks.** Despite the existence of a closed-form solution
for ratio trace optimization problem, the obtained solution does
not necessarily best optimize the corresponding trace ratio optimization algorithm, which is the essential objective function for general dimensionality reduction. For supervised dimensionality reduction algorithms, this approximation may sacrifice the potential classification capability of the derived low-dimensional feature space, which is demonstrated later in our experiments. This motivates the need for a procedure to directly solve the trace ratio optimization problem.

### 3.3 Efficient Solution of Trace Ratio Problem

In this section, we present an efficient procedure to solve the trace ratio problem with the assumption that \( W^T W = I_d \). Denote \( S_t = S_p + S_l \), then the trace ratio optimization problem in (3.3) is equivalent to

\[
W^* = \arg \max_{W^T W = I_d} \frac{Tr(W^T S_p W)}{Tr(W^T S_l W)}.
\] (3.10)

We have \( 0 \leq Tr(W^T S_p W)/Tr(W^T S_l W) \leq 1 \), and the maximum value 1 of (3.10) corresponds to the maximum of (3.3), namely \(+\infty\). Without losing generality, we instead solve for (3.10) in the following. Our procedure consists of two steps.

1. **Remove the Null Space of \( S_t \) with Principal Components Analysis (PCA) [45]**. The matrices \( S_p \) and \( S_l \) are both positive semidefinite, and the intersection of their null spaces is equal to the null space of \( S_t \), namely, \( \{x \mid S_t x = 0\} \). As the null space of \( S_t \) does not contain discriminating information for the training data (\( x^T S_p x = 0 \) and \( x^T S_l x = 0 \)), they may be removed from the solution space without sacrificing accuracy. Assume that the singular value decomposition of matrix \( S_t \) is

\[
S_t = U \Lambda U^T,
\]
where $\Lambda = [\lambda_1, \lambda_2, \ldots, \lambda_{m'}]$, $\lambda_k > 0$, $k = 1, 2, \ldots, m'$, and $m'$ is the number of positive singular values of $S_t$. Then the solution is constrained to lie within the space spanned by the column vectors of $U$, namely, $W = UV, V \in \mathbb{R}^{m' \times d}$, and the problem defined in (3.10) is changed to

$$V^* = \arg \max_{V^T V = I_d} \frac{Tr(V^T S^u_p V)}{Tr(V^T S^u_t V)}. \tag{3.11}$$

where $S^u_p = U^T S_p U$ and $S^u_t = U^T S_t U$. Then, the denominator of the objective function (3.11) is always positive for non-zero $V$, that is, $S^u_t$ is positive definite.

2. Iterative optimization. Here, we first introduce our iterative algorithm to solve (3.11). Its theoretical justifications will be presented in Section-3.4. In each step, we solve a trace difference problem

$$V^* = \arg \max_{V^T V = I_d} Tr[V^T (S^u_p - \lambda^n S^u_t) V],$$

where $\lambda^n$ is the trace ratio value calculated from the projection matrix $V^{n-1}$ of the previous step. The detailed procedure is listed in Algorithm 1.

3.4 Proof of Convergency to Global Optimum

3.4.1 Proof of the monotonic increase of $\lambda^n$

Denote the objective function of (3.11) as

$$J(V) = \frac{Tr(V^T S^u_p V)}{Tr(V^T S^u_t V)}. \tag{3.16}$$

Then, the monotonic increase of $\lambda^n$ is guaranteed by the following theorem.

**Lemma-1.** For Algorithm 1 to solve the trace ratio optimization problem, we must have

$$J(V^n) \geq J(V^{n-1})$$

namely $\lambda^{n+1} \geq \lambda^n. \tag{3.17}$
Proof. Denote $g_n(V) = Tr(V^T(S_p^u - \lambda^n S_t^u)V)$, then $g_n(V^{n-1}) = 0$. Moreover, from Algorithm 1 and the assumption that $V^T V = I_d$, we have [29]

$$\sup_{V^T V = I_d} g_n(V) = \sum_{k=1}^{d} \tau_k^n.$$ 

Also, $g_n(V^n) = \sum_{k=1}^{d} \tau_k^n$ from (3.14) and (4.12). Then we have

$$g_n(V^n) \geq g_n(V^{n-1}) = 0.$$ 

Namely, $Tr[V^{nT}(S_p^u - \lambda^n S_t^u)V^n] \geq 0$. As the matrix $S_t^u$ is positive definite, we obtain

$$\frac{Tr(V^{nT} S_p^u V^n)}{Tr(V^{nT} S_t^u V^n)} \geq \lambda^n,$$

that is,

$$J(V^n) \geq J(V^{n-1}),$$

namely $\lambda^{n+1} \geq \lambda^n$. $\square$

From Theorem-1, we can conclude that the trace ratio value will monotonically increase.

3.4.2 Proof of $V^n$ convergence and global optimum for $\lambda$

To prove the convergence of the projection matrix $V^n$, we first introduce the concept of point-to-set mapping and some related lemmas [35]. The power set $\wp(\chi)$ of a set $\chi$ is the collection of all subsets of $\chi$. A point-to-set map $\Omega$ is a function: $\chi \rightarrow \wp(\chi)$ [28]. In our iterative procedure to the trace ratio optimization problem, the map from $V^{n-1}$ to $V^n$ can be considered as a point-to-set map, since each $V^n$ with any orthogonal transformation will not change the value of the objective function $J(V^n)$.

Strict Monotonicity [28]. An algorithm is a point-to-set map $\Omega: \chi \rightarrow \wp(\chi)$. Given an initial point $X_1$, the algorithm
generates a sequence of points via the rule that \( X_n \in \Omega(X_{n-1}) \). Suppose \( J : \chi \to \mathbb{R}_+ \) is a continuous, non-negative function; an algorithm is called \textit{strictly monotonic} if 1) \( Y \in \Omega(X) \) implies that \( J(Y) \geq J(X) \), and 2) \( Y \in \Omega(X) \) and \( J(Y) = J(X) \) imply that \( Y = X \).

To prove the convergence of the projection matrix \( V^n \), we will utilize the following lemma.

\textbf{Lemma-2 \([35]\).} Assume that the algorithm \( \Omega \) is strictly monotonic with respect to \( J \) and that it generates a sequence \( \{X_n\} \) which lies in a compact set. If \( \chi \) is normed, then \( \|X_n - X_{n-1}\| \to 0 \). If \( \Omega \) is closed at an accumulation point \( \hat{X} \), \( \lim_{n \to \infty} X^n = \hat{X} \), then \( \hat{X} \) is a fixed point, namely \( \{\hat{X}\} = \Omega(\hat{X}) \).

In our proposed iterative procedure (3.12-4.12) for the trace ratio optimization problem, let \( \chi = \mathbb{R}^{m' \times d} \cap \varphi \), where the set \( \varphi = \{X \mid \exists V \in \mathbb{R}^{m' \times d}, X \text{ is the solution of (3.12-4.12) with } V^{n-1} = V \} \). Then the iterative algorithm from (3.12) to (4.12), denoted as \( \Omega \), is a point-to-set map (the set contains one point when \( V^n \) is constrained to have been obtained from (3.12-4.12)), and it generates a sequence of points via the rule that \( V^n = \Omega(V^{n-1}) \). This algorithm is strictly monotonic as proven below.

\textbf{Lemma-3} The iterative algorithm (3.12-4.12) for the trace ratio optimization problem is strictly monotonic with respect to \( J = J(V) \) as defined in (3.16).

\textbf{Proof.} It is obvious that \( J(V) \) is a continuous, non-negative function. From Lemma-1, we have \( J(V^n) \geq J(V^{n-1}) \), hence the first condition for strict monotonicity is satisfied. For the second condition, if \( Y = \Omega(X) \) and \( J(Y) = J(X) \), then \( \max_V Tr[V^T(S_p^u - \lambda S_t^u)V] = 0 \) where \( \lambda = Tr(X^T S_p^u X) / Tr(X^T S_p^u X) \); otherwise, we have \( J(Y) > J(X) \). As \( S_t^u \) is positive definite, we have \( \max_V \frac{Tr(V^T S_p^u V)}{Tr(V^T S_p^u V)} = \lambda \). Since both \( X \) and \( Y \) achieve the maximum of the objective function \( J(V) \), they both maximize \( Tr[V^T(S_p^u - \lambda S_t^u)V] \) [24]. As the maximum of \( Tr[V^T(S_p^u - \lambda S_t^u)V] \) lies in the
space spanned by the first $d$ eigenvectors of the matrix $(S_p^n - \lambda S_t^n)$ \cite{29}, there only exists one orthogonal transform between these two matrices. Given that both $X$ and $Y$ are constrained in the set $\varphi$, they are orthogonal transform invariant as in (4.12); therefore, we have $X = Y$. Then, we can conclude that the iterative algorithm (3.12-4.12) for the trace ratio optimization problem is strictly monotonic with respect to $J = J(V)$ as defined in (3.16).

All possible $\lambda^n$ computed from the set $\{V \mid V^T V = I_d, V \in \mathbb{R}^{n' \times d}\}$ will constitute a compact set. As described above, all members of $\chi$ are computed from (3.14) and (4.12); hence $\chi$ is compact owing to the continuity of the mapping from $\lambda^n$ to $V^n$. A more detailed proof is omitted here.

Based on the above lemmas, we can have the following theorem on the convergence of the $\lambda^n$ to the global optimum.

**Theorem-1.** For the iterative procedure (3.12-4.12) defined in Algorithm 1, we have $\|V^n - V^{n-1}\| \to 0$. Denote $\lim_{n \to \infty} V^n = V$, then $V \in \arg \max V \frac{Tr(V^T S_p V)}{Tr(V^T S_t V)}$, that is, $\lambda^n$ will monotonically increase and converge to the global optimum.

**Proof.** From Lemma-2 and Lemma-3, we can directly reach the conclusion that $\|V^n - V^{n-1}\| \to 0$. From Lemma-2, we have $J(V) = J(\Omega(V))$. According to the proof of Lemma-3, we have $V \in \arg \max V \frac{Tr(V^T S_p V)}{Tr(V^T S_t V)}$.

As proven in Lemma-1, $\lambda^n$ will monotonically increase, hence we can conclude that $\lambda^n$ will monotonically increase and converge to the global optimum along with the corresponding projection matrix $V^n$. \hfill $\Box$

\footnote{It is deduced with the assumption that there do not exist duplicated eigenvalues for $(S_p^n - \lambda S_t^n)$.}
3.5 Extension and Discussion

3.5.1 Extension to General Constraints

As mentioned previously, we have the assumption $W^TW = I$ for the trace ratio optimization problem. In practice, a more general constraint $W^TCW = I$, where $C$ is any positive semidefinite matrix, may be imposed. Here, we take as example the kernelization of the graph embedding framework in [60], where $C$ is a kernel matrix, to introduce how to solve the trace ratio optimization problem with general constraints.

The intuition of the kernel trick is to map the data from the original input feature space to another higher dimensional Hilbert space as $\phi: x \rightarrow \mathcal{F}$, and then perform linear dimensionality reduction in this new feature space. This approach is well suited to algorithms that need only to compute the inner product of data pairs $k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. Assuming that the transformation matrix $W = [\phi(x_1), \phi(x_2), \cdots, \phi(x_N)]\tilde{W}$ and $K$ is the kernel Gram matrix with $K_{ij} = \phi(x_i) \cdot \phi(x_j)$, we have the following optimization problem from (3.3):

$$\max_{\tilde{W}^TK \tilde{W} = I} \left\{ \frac{Tr(\tilde{W}^TKL\rho K\tilde{W})}{Tr(W^TKLKW)} = \frac{Tr(\tilde{W}^TS_p^{k}\tilde{W})}{Tr(W^TS_p^{k}W)} \right\}, \quad (3.18)$$

where the constraint matrices are $S_p^{k} = KL\rho K$ and $S_l^{k} = K\rho L K$.

Assume that the singular value decomposition of the kernel matrix $K$ is $K = U_k\Lambda_k(U_k)^T$, where $\Lambda_k$ is a diagonal matrix with positive diagonal elements. Let $\tilde{W} = U_k\Lambda_k^{-1/2}W_1$, then we can simplify the optimization problem (3.18) into

$$\max_{W_1^T W_1 = I} \frac{Tr(W_1^T S_p^1 W_1)}{Tr(W_1^T S_l^1 W_1)}, \quad (3.19)$$

where the constant matrix $S_p^1 = \Lambda_k^{-1/2}U_k^T S_p^k U_k \Lambda_k^{-1/2}$ and $S_l^1 = \Lambda_k^{-1/2}U_k^T S_l^k U_k \Lambda_k^{-1/2}$. Now this optimization problem is converted
into the form in (3.3), and hence we can use the proposed Algorithm 1 to search for the global optimum.

3.5.2 Discussion

**Trace Ratio vs. Ratio Trace:** Trace ratio and ratio trace present two different formulations to the general dimensionality reduction problem. They are interlinked in the following aspects. First, for the trace ratio formulation, the objective function is invariant under any orthogonal transformation of $V$; while for the ratio trace formulation, it is invariant under any non-singular transformation matrix of $V$. Hence, the former is invariant for classification if based on Euclidean distance, while different solutions of the latter may change the similarity and thus is unstable for classification. Second, the ratio trace formulation has a closed-form solution and is more efficient compared to the trace ratio formulation. Third, for the ratio trace formulation, the column vectors of the projection matrix are not required to be orthogonal, hence it essentially puts different weights on different projection directions. Assume that the singular value decomposition of $W$ is $W = U_w \Lambda_w V_w^T$; then, as the right orthogonal matrix $V_w$ will not change the similarity if based on the Euclidean distance, the projection directions encoded by the column vectors of $U_w$ are given different weights from the diagonal elements of $\Lambda_w$. Finally, the trace ratio formulation is the essential formulation for general dimensionality problem, which directly leads to the superiority of the solution from the trace ratio formulation over that from the ratio trace formulation.

**Relationship with Guo’s Work [24]:** Guo et al. [24] proposed a method to solve the trace ratio problem. Our proposed algorithm is different from Guo’s in many aspects. First, Guo’s work proves convergence of only the trace ratio value, and does
not prove the convergence of the projection matrix $V$; while our algorithm provides convergence proofs for both the trace ratio value and the projection matrix $V$. Second, Guo’s work utilizes the dichotomy method to select the trace ratio value for their trace difference formulation, which commonly exhibits slow convergence of the trace ratio value. As shown in the experiment section, our proposed method converges much faster than Guo’s work. Third, in our proposed algorithm, the trace ratio value increases monotonically, hence it is guaranteed that the performance improves step-by-step; while in Guo’s work, the derived trace ratio value may fluctuate. Finally, our algorithm is proposed for general dimensionality reduction problems and further extended for solving kernel-based subspace learning problems formulated in trace ratio form. The work [56] also discussed the trace ratio problem and applied the multi-scale search for pursuing the solution; hence also suffers from the same issues as Guo’s work.

### 3.6 Experiments

In this section, our proposed Iterative algorithm for the Trace Ratio (ITR) optimization problem is systematically evaluated in four aspects, taking the LDA and MFA [60] algorithms as instances of trace ratio problems. The first is the evaluation of convergence speed in comparison to Guo’s work [24]; the second is visualization of the projection matrices of ITR compared to PCA and the ratio trace based LDA; the third is evaluation of the classification capability of the derived low-dimensional feature spaces from linear dimensionality reduction algorithms; and the fourth is evaluation of the classification capability of the derived low-dimensional feature space for kernel-based dimensionality reduction algorithms.
3.6.1 Dataset Preparation

In our experiments, we use six data sets. The first three are the benchmark face databases FERET, ORL, and CMU PIE\(^1\) with high-dimensional features. For the face databases, all images are aligned by fixing the locations of the two eyes. From the FERET database, we use seventy people with six images for each person; the images are normalized in size to 56-by-46 pixels. The ORL database contains 400 images of 40 persons, where each image is normalized in size to 56-by-46 pixels. The CMU PIE (Pose, Illumination, and Expression) database contains more than 40,000 facial images of 68 people. In our experiment, a subset of five near frontal poses (C27, C05, C29, C09 and C07)

\(^1\)Available at http://www.face-rec.org/databases/.
and illuminations indexed as 08 and 11 is used. Each person has ten images and all the images are normalized to 64-by-64 pixels. The other three data sets are wine, iris and ionosphere (iono) from the UC Irvine repository\footnote{Available at \url{http://www.ics.uci.edu/mlearn/MLRepository.html}}; this data has relatively small feature dimensions.

### 3.6.2 Convergence Speed

In this subsection, the convergence property of our proposed algorithm ITR is evaluated by a comparison to Guo’s method. The convergence property is evaluated in two aspects. One is the convergence of the projection matrix $V^n$, determined according to the difference of $V^n$ and $V^{n-1}$ ($\|V^n - V^{n-1}\|$). The other is the speed of $|Tr[V^{nT}(S_p - \lambda^n S_t)V^n]|$ converging to zero. As described in the proof of Lemma-3, the largest trace ratio value results in $Tr[V^{nT}(S_p - \lambda^n S_t)V^n] = 0$; hence this evaluation measures the convergence speed of the trace ratio value to the global optimum, and the accuracy of the projection matrix $V$.

The FERET and CMU PIE databases are used for these evaluations. For both ITR and Guo’s method, we optimize the objective function in (3.11) and from the LDA algorithm. Detailed results are shown in Figure 3.1, from which we can see that ITR can converge much faster than Guo’s method. Commonly, ITR converges after about 5 iterations. Moreover, the accuracy of the trace ratio value, characterized by the value of $|Tr[V^{nT}(S_p - \lambda^n S_t)V^n]|$, from the ITR algorithm is much better than that from Guo’s method.

### 3.6.3 Visualization of Projection Matrix

In this subsection, we examine the visual properties of the projection matrix $W$ computed by our proposed ITR algorithm
within LDA, and compare it to the traditional ratio trace formulation within LDA and to the PCA algorithm. The FERET and ORL databases are used for this experiment. From the FERET database, three images of each subject are randomly selected for the computation of the projection matrices, while four images of each subject are taken from the ORL database. For the ITR algorithm, the reduced feature dimension is set to $d = 10$ for the computation of the projection matrix. For LDA related algorithms, we first conduct PCA to reduce the dimension to $N - N_c$ as in [3], and then perform LDA. The final projection matrix is the product of the PCA projection matrix and the LDA projection matrix.

The column vectors of the projection matrix are reshaped into a matrix of the original image size. All the results are shown in Figure 3.2. They demonstrate that PCA vectors look similar to a face, which agrees with the motivation of PCA; while the results from LDA related algorithms are more noisy, which indicates that the most discriminating features perhaps do not possess explicit global semantics.
### Classification by Linear Trace Ratio Algorithms with Orthogonal Constraints

Table 3.1: Recognition error rates (%) of Ratio Trace + Kernel Discriminant Analysis (RKDA), Trace Ratio + Kernel Discriminant Analysis (ITR-KDA), Ratio Trace + Kernel MFA (RKMFA), and Trace Ratio + Kernel MFA (ITR-KMFA) on the three face databases.

<table>
<thead>
<tr>
<th>Database</th>
<th>RKDA</th>
<th>ITR-KDA</th>
<th>KMFA</th>
<th>ITR-KMFA</th>
</tr>
</thead>
<tbody>
<tr>
<td>FERET</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N3T3</td>
<td>8.0</td>
<td>6.3</td>
<td>8.0</td>
<td>4.8</td>
</tr>
<tr>
<td>N2T4</td>
<td>26.4</td>
<td>17.5</td>
<td>26.4</td>
<td>17.5</td>
</tr>
<tr>
<td>ORL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N4T6</td>
<td>6.7</td>
<td>6.7</td>
<td>6.7</td>
<td>5.0</td>
</tr>
<tr>
<td>N3T7</td>
<td>14.3</td>
<td>12.1</td>
<td>14.3</td>
<td>12.1</td>
</tr>
<tr>
<td>N2T8</td>
<td>26.6</td>
<td>22.2</td>
<td>26.6</td>
<td>21.6</td>
</tr>
<tr>
<td>PIE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N4T6</td>
<td>2.4</td>
<td>1.3</td>
<td>2.4</td>
<td>1.1</td>
</tr>
<tr>
<td>N3T7</td>
<td>3.4</td>
<td>2.7</td>
<td>3.4</td>
<td>2.7</td>
</tr>
<tr>
<td>N2T8</td>
<td>11.9</td>
<td>7.7</td>
<td>11.9</td>
<td>6.9</td>
</tr>
</tbody>
</table>

In this subsection, we conduct classification experiments on the face databases with high feature dimensions. Our proposed ITR procedure is compared to the ratio trace solution using LDA and MFA as examples. For MFA related algorithms, the number of nearest neighbors of each sample is fixed as 4, and the number of shortest pairs from different classes is set as 40. To speed up model training, PCA is conducted as a preprocessing step for LDA/MFA related algorithms. Two PCA dimensions are tested before LDA/MFA: one is $N-N_c$, which is equivalent to the Fisherface algorithm [3] for LDA; the other is $(N-1)$, and in this case, LDA and MFA are implemented by first transforming the objective functions into the form of Eqn. (3.11) for avoiding the singular value issue. With our ITR algorithm, we also test these two dimensions for PCA before formally performing
CHAPTER 3. THE TRACE RATIO OPTIMIZATION

Figure 3.3: Recognition error rates over different dimensions. The configuration is N3T7 on the CMU PIE database. For LDA and MFA, the dimension of the preprocessing PCA step is $N-N_c$.

ITR, and the maximum iteration number $T_{\text{max}}$ in Algorithm 2 is set to 16 in all the experiments. For comparison, the classification result from PCA and that on the original gray-level features without dimensionality reduction are also reported as the baselines, denoted as ‘PCA’ and ‘w/o DR.’ in the tables and figures. In all experiments, the Nearest Neighbor method is used for final classification. For each database, we test various configurations of training and testing sets, where $NxTy$ indicates that $x$ images of each subject are randomly selected for model training and the remaining $y$ images of each subject are used for testing. Detailed results are listed in Table 3.2, and recognition error rates over different feature dimensions for the experiment N3T7 on the CMU PIE database are displayed in Figure 3.3. From the results, we can reach the conclusion that the trace ratio formulation generally outperforms the corresponding ratio trace formulation in terms of classification capability of the derived low-dimensional feature space, with only one exception for $N^2T8$ of RLDA2 in the ORL database.

We also conduct classification experiments on the UCI databases with features of low dimensions. The sample data are normalized such that each feature has a standard deviation of one,
Figure 3.4: Testing classification errors on three UCI databases for both linear and kernel-based algorithms. Results are obtained from 100 realizations of randomly generated 70/30 splits of data.

and no PCA step is used for preprocessing before LDA/MFA since the feature dimension is already relatively small for these data sets. For LDA/MFA related algorithms, the best result of all possible LDA/MFA feature dimensions is reported. We randomly split each data set 100 times into training (70%) and testing (30%) sets, and the classification errors (mean and standard deviation of the testing error) are charted in Figure 3.4. These results again validate the superiority of the trace ratio formulation over the ratio trace formulation.
3.6.5 Classification by Kernel Trace Ratio algorithms with General Constraints

We also evaluate the effectiveness of the ITR procedure for solving the trace ratio problem with general constraints. Kernel-based LDA and MFA are used as examples for the trace ratio and ratio trace formulations. In all the experiments, the Gaussian Kernel $\exp\{-\|x - y\|^2/\delta^2\}$ is used, and parameter $\delta$ is set as $\delta = 2^{(n-10)/2.5}\delta_0$, $n = 0, 1, \ldots, 20$, where $\delta_0$ is the standard derivation of the training data set. The reported result is the best one among the 21 configurations. The three face databases and the three UCI databases are used for experiments. Detailed results are listed in Table 2 and Figure 3.4. They show that for both kernel-based Discriminant Analysis and kernel-based MFA algorithms, the trace ratio based solutions are consistently superior to the ratio trace based solutions.

Discussion: Over the past few decades, many algorithms have been proposed for dimensionality reduction [64][61]; even for just the PCA+LDA/MFA paradigm, numerous different procedures have been proposed on how to select the PCA dimension, and there are two parameters in MFA that can be tuned for better performance. In this work, we did not try to evaluate all the algorithms to determine which one is best; instead, we claim that for each algorithm, solutions based on the trace ratio formulation are better than those from the corresponding ratio trace versions. For the PCA+LDA/MFA paradigm, when the PCA dimension is fixed, this benefit of the trace ratio formulation can also be gained in the LDA/MFA step.

3.7 Conclusion

In this chapter, an efficient iterative procedure (ITR) has been proposed to directly solve the trace ratio optimization problem.
The convergence of the projection matrix and the global optimality of the trace ratio value were proven. ITR truly provides the optimal solution for the joint objectives of most popular dimensionality reduction algorithms, and it converges much faster than the algorithm proposed by Guo et al. The superiority of solutions from ITR over those from the ratio trace formulation has been extensively verified by large number of experiments on various data sets and algorithms.

☐ End of chapter.
Algorithm 1 Iterative Procedure to Solve the Trace Ratio Optimization Problem

1: Initialize $V^0$ as an arbitrary columnly orthogonal matrix;
2: For $n=1, 2, \ldots, N_{\text{max}}$, Do
   1. Compute the trace ratio value $\lambda^n$ from the projection matrix $V^{n-1}$:
      \[
      \lambda^n = \frac{\text{Tr}[V^{n-1T}S_pV^{n-1}]}{\text{Tr}[V^{n-1T}S_tV^{n-1}]}.
      \] (3.12)
   2. Construct the trace difference problem as
      \[
      V^n = \arg \max_{V^T V = I_d} \text{Tr}[V^T (S_p - \lambda^n S_t) V].
      \] (3.13)
   3. Solve the trace difference problem using the eigenvalue decomposition method:
      \[
      (S_p - \lambda^n S_t)v^n_k = \tau^n_k v^n_k, \]
      where $\tau^n_k$ is the $k$-th largest eigenvalue of $(S_p - \lambda^n S_t)$ with the corresponding eigenvector $v^n_k$.
   4. Reshape the projection matrix for the sake of orthogonal transformation invariance:
      (a) Set $V^n = [v^n_1, v^n_2, \ldots, v^n_d]$, where $d$ is the desired lower feature dimension;
      (b) Let $S^n_t = V^n(V^n)^T S_t V^n (V^n)^T$;
      (c) Conduct singular value decomposition as
      \[
      S^n_t = V^n \Lambda^n V^n^T.
      \] (3.15)
   5. If $\|V^n - V^{n-1}\| < \sqrt{m'd} \varepsilon$ ($\varepsilon$ is set to $10^{-4}$ in this work), then break.
3: Output $V = V^n$.  

CHAPTER 3. THE TRACE RATIO OPTIMIZATION
Table 3.2: Recognition error rates (%) of PCA, PCA \((N-N_c)\)+Ratio Trace based LDA (RLDA), PCA \((N-N_c)\)+Trace Ratio based LDA (ITR-LDA), PCA \((N-1)\)+Ratio Trace based LDA (RLDA2), PCA \((N-1)\)+Trace Ratio based LDA (ITR-LDA2), PCA \((N-N_c)\)+Ratio Trace based MFA (RMFA), PCA \((N-N_c)\)+Trace Ratio based MFA (ITR-MFA), PCA \((N-1)\)+Trace Ratio based MFA (RMFA2), PCA \((N-1)\)+Trace Ratio based MFA (ITR-MFA2), and the method without dimensionality reduction on the three face databases. Note that the boldtype numbers are those with the best classification accuracies for LDA and MFA respectively.

<table>
<thead>
<tr>
<th>Configure</th>
<th>Unsupervised</th>
<th>LDA Related Algorithms</th>
<th>MFA Related Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FERET</strong></td>
<td>w/o DR.</td>
<td>PCA</td>
<td>RLDA</td>
</tr>
<tr>
<td>N3T3</td>
<td>17.1</td>
<td>16.2</td>
<td>8.6</td>
</tr>
<tr>
<td>N2T4</td>
<td>32.9</td>
<td>33.2</td>
<td>21.1</td>
</tr>
<tr>
<td><strong>ORL</strong></td>
<td>w/o DR.</td>
<td>PCA</td>
<td>RLDA</td>
</tr>
<tr>
<td>N4T6</td>
<td>12.1</td>
<td>10.8</td>
<td>11.7</td>
</tr>
<tr>
<td>N3T7</td>
<td>18.6</td>
<td>17.5</td>
<td>15.7</td>
</tr>
<tr>
<td>N2T8</td>
<td>28.4</td>
<td>27.5</td>
<td>28.4</td>
</tr>
<tr>
<td><strong>PIE</strong></td>
<td>w/o DR.</td>
<td>PCA</td>
<td>RLDA</td>
</tr>
<tr>
<td>N4T6</td>
<td>14.3</td>
<td>11.3</td>
<td>4.0</td>
</tr>
<tr>
<td>N3T7</td>
<td>19.0</td>
<td>15.9</td>
<td>7.0</td>
</tr>
<tr>
<td>N2T8</td>
<td>22.4</td>
<td>17.5</td>
<td>18.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Configure</th>
<th>Unsupervised</th>
<th>LDA Related Algorithms</th>
<th>MFA Related Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FERET</strong></td>
<td>w/o DR.</td>
<td>PCA</td>
<td>RLDA</td>
</tr>
<tr>
<td>N3T3</td>
<td>17.1</td>
<td>16.2</td>
<td>9.5</td>
</tr>
<tr>
<td>N2T4</td>
<td>32.9</td>
<td>33.2</td>
<td>20.7</td>
</tr>
<tr>
<td><strong>ORL</strong></td>
<td>w/o DR.</td>
<td>PCA</td>
<td>RLDA</td>
</tr>
<tr>
<td>N4T6</td>
<td>12.1</td>
<td>10.8</td>
<td>10.8</td>
</tr>
<tr>
<td>N3T7</td>
<td>18.6</td>
<td>17.5</td>
<td>15.4</td>
</tr>
<tr>
<td>N2T8</td>
<td>28.4</td>
<td>27.5</td>
<td>27.5</td>
</tr>
<tr>
<td><strong>PIE</strong></td>
<td>w/o DR.</td>
<td>PCA</td>
<td>RLDA</td>
</tr>
<tr>
<td>N4T6</td>
<td>14.3</td>
<td>11.3</td>
<td>4.2</td>
</tr>
<tr>
<td>N3T7</td>
<td>19.0</td>
<td>15.9</td>
<td>7.3</td>
</tr>
<tr>
<td>N2T8</td>
<td>22.4</td>
<td>17.5</td>
<td>16.9</td>
</tr>
</tbody>
</table>
Chapter 4

A Convergent Solution to Tensor Subspace Learning

Summary

The trace ratio optimization procedure we proposed in Chapter 3 is not confined in the analysis of samples with vector representation. Recently, growing interest is drawn to the tensor representation of samples and lots of works have been devoted to the subspace analysis with tensor representation. Unfortunately, no convergent solution is proposed due to the discrepancy among the objectives along different dimensions caused by the traditional ratio trace optimization. Since our proposed procedure directly optimizes the trace ratio, the integration with tensor subspace analysis leads to a convergent solution.

4.1 Introduction

Subspace learning algorithms [11] such as Principal Component Analysis (PCA) [45] and Linear Discriminant Analysis (LDA) [3] traditionally express the input data as vectors and often in
a high-dimensional feature space. In real applications, the ex-
tracted features are usually in the form as a multidimensional
union, i.e. a tensor, and the vectorization process destroys this
intrinsic structure of the original tensor form. Another draw-
back brought by the vectorization process is the curse of dimen-
sionality which may greatly degrade the algorithmic learnability
especially in the small sample size cases.

Recently substantial efforts have been devoted to the employ-
ment of tensor representation for improving algorithmic learn-
ability [46]. Among them, 2DLDA [63] and DATER [59] are
tensorized from the popular vector-based LDA algorithm. Al-
though the initial objectives of these algorithms are different,
they all end up with solving a higher-order optimization prob-
lem, and commonly iterative procedures were used to search for
the solution. A collective problem encountered by their solu-
tion procedures is that the iterative procedures are not guar-
anteed to converge, since in each iteration, the optimization
problem is approximately simplified from the Trace Ratio form
\[
\arg \max_{U^k} \frac{\text{Tr}(U^k S^p_k U^k)}{\text{Tr}(U^k S^k U^k)}
\]
to the Ratio Trace form
\[
\arg \max_{U^k} \text{Tr}[ (U^k S^k U^k)^{-1} (U^k S^p_k U^k) ]
\]
in order to obtain a closed-form solution for each iteration. Consequently, the de-
\[\text{Matrices } S^p_k \text{ and } S^k \text{ are both positive semidefinite and more detailed definitions are as afterward.}\]
problem \( \max_{U^k} Tr[U^k T (S_k^p - \lambda S_k) U^k] \) where \( \lambda \) is the objective function value computed from the \( (U^k|_{k=1}^{kn}) \) obtained from the previous iteration. Then, each iteration is efficiently solved with the eigenvalue decomposition method [20]. A detailed proof is presented to justify that \( \lambda \), namely the value of the objective function, will increase monotonically, and also we prove that the projection matrix \( U^k \) will converge to a fixed point based on the point-to-set map theories [28].

It is worthwhile to highlight some aspects of our solution procedure to general subspace learning based on tensor representation here:

1. The value of the objective function is guaranteed to monotonically increase; and the multiple projection matrices are proved to converge. These two properties ensure the algorithmic effectiveness and applicability.

2. Only eigenvalue decomposition method is applied for iterative optimization, which makes the algorithm extremely efficient; and the whole algorithm does not suffer from the singularity problem that is often encountered by traditional generalized eigenvalue decomposition method used to solve the ratio trace optimization problem.

3. The consequent advantage brought by the sound theoretical foundation is the enhanced potential classification capability of the derived low-dimensional representation from the subspace learning algorithms.

The rest of this Chapter is organized as follows. Section II reviews the general subspace learning based on tensor representation, and then we introduce our new solution procedure along with the theoretical convergency proof in section III. By taking Marginal Fisher Analysis (MFA) algorithm proposed in [60] as an example, we verify the convergency properties of the new
proposed solution procedure and the classification capability of the derived low-dimensional representation is examined with a set of experiments on the benchmark face databases in Section IV.

4.2 Subspace Learning with Tensor Data

In this section, we present a general subspace learning framework by encoding data as tensors of arbitrary order, extended from the one proposed by [60] and taking the data inputs as vectors. The concepts of tensor inner production, mode-\(k\) production with matrix, and mode-\(k\) unfolding are referred to the work of [59].

4.2.1 Graph Embedding with Tensor Representation

Denote the sample set as \(X = [X_1, X_2, \ldots, X_N], X_i \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_n}, \)
\(i = 1, \ldots N,\) with \(N\) as the total number of samples. Let \(G = \{X, S\}\) be an undirected similarity graph, called an intrinsic graph, with vertex set \(X\) and similarity matrix \(S \in \mathbb{R}^{N \times N}\). The corresponding diagonal matrix \(D\) and the Laplacian matrix \(L\) of the graph \(G\) are defined as

\[
L = D - S, \quad D_{ii} = \sum_{j \neq i} S_{ij} \quad \forall \ i. \quad (4.1)
\]

The task of graph embedding is to determine a low-dimensional representation of the vertex set \(X\) that preserves the similarities between pairs of data in the original high-dimensional feature space. Denote the low-dimensional embedding of the vertices as \(Y = [Y_1, Y_2, \ldots, Y_N]\), where \(Y_i \in \mathbb{R}^{m'_1 \times m'_2 \times \ldots \times m'_n}\) is the embedding for the vertex \(X_i\), with the assumption that \(Y_i\) is the mode-\(k\) production of \(X_i\) with a series of column orthogonal matrices \(U^k \in \mathbb{R}^{m_k \times m'_k}\),

\[
Y_i = X_i \times_1 U^1 \times_2 U^2 \ldots \times_n U^n, \quad U^{kT}U^k = I_{m'_k}, \quad (4.2)
\]
where $I_{m_k'}$ is an $m_k'\times m_k'$ identity matrix. To maintain similarities among vertex pairs according to the graph preserving criterion [60], we have

$$
(U^k)^*_{k=1} = \arg \min_{U^k|k=1} \sum_{i \neq j} \frac{\| Y_i - Y_j \|^2 S_{ij}}{f(U^k|k=1)}\quad (4.3)
$$

$$
= \arg \min_{U^k|k=1} \sum_{i \neq j} \frac{\| (X_i - X_j) \times_k U^k|k=1 \|^2 S_{ij}}{f(U^k|k=1)},\quad (4.4)
$$

where $f(U^k|k=1)$ is a function that poses extra constraint for the graph similarity preserving criterion. Here $U^k|k=1$ means the sequence $U^1$, $U^2$ to $U^n$ and so for the other similar representations in the following parts of this work. Commonly, $f(U^k|k=1)$ may have two kinds of definitions. One is for scale normalization, that is,

$$
f(U^k|k=1) = \sum_{i=1}^N \| X_i \times_k U^k|k=1 \|^2 B_{ii},\quad (4.5)
$$

where $B$ is a diagonal matrix with non-negative elements. The other is a more general constraint which relies on a new graph, referred to as penalty graph with similarity matrix $S^p$, and is defined as

$$
f(U^k|k=1) = \sum_{i \neq j} \| (X_i - X_j) \times_k U^k|k=1 \|^2 S^p_{ij}.\quad (4.6)
$$

Without losing generality, we assume that the constraint function is defined with penalty matrix for simplicity; and for scale normalization constraint, we can easily have the similar deduction for our new solution procedure. Then, the general formulation of the tensor-based subspace learning is expressed as

$$
\arg \max_{U^k|k=1} \frac{\sum_{i \neq j} \| (X_i - X_j) \times_k U^k|k=1 \|^2 S^p_{ij}}{\sum_{i \neq j} \| (X_i - X_j) \times_k U^k|k=1 \|^2 S_{ij}}.\quad (4.7)
$$
Recent studies [43] [62] [63] [59] have shown that dimensionality reduction algorithms with data encoded as high-order tensors usually outperform those with data represented as vectors, especially when the number of training samples is small. Representing images as 2D matrices instead of vectors allows correlations between both rows and columns to be exploited for subspace learning.

Generally, no closed-form solution exists for (4.7). Previous works [63] [59] utilized iterative procedures to search for approximate solutions. First, the projection matrices $U_1, \ldots, U_n$ are initialized arbitrarily; then each projection matrix $U_k$ is refined by fixing the other projection matrices $U_1, \ldots, U_{k-1}, U_{k+1}, \ldots, U_n$ and solving the optimization problem:

$$U_k^* = \arg \max_{U_k} \frac{\sum_{i \neq j} \|U_k^T Y_i^k - U_k^T Y_j^k\|^2 S_{ij}^p}{\sum_{i \neq j} \|U_k^T Y_i^k - U_k^T Y_j^k\|^2 S_{ij}}$$  \hspace{1cm} (4.8)

$$= \arg \max_{U_k} \frac{\text{Tr}(U_k^T S_k^p U_k)}{\text{Tr}(U_k^T S_k^k U_k)}$$  \hspace{1cm} (4.9)

where $Y_i^k$ is the mode-$k$ unfolding matrix of the tensor $\tilde{Y}_i = X_i \times_1 U^1 \ldots \times_{k-1} U^k-1 \times_{k+1} U^{k+1} \ldots \times_n U^n$ and $S_k = \sum_{i \neq j} S_{ij}(Y_i^k - Y_j^k)(Y_i^k - Y_j^k)^T$.

The optimization problem in (4.9) is still intractable, and traditionally its solution is approximated by transforming the objective function in (4.9) into a more tractable approximate form, namely, Ratio Trace form,

$$U_k^* = \arg \max_{U_k} \text{Tr}((U_k^T S_k^p U_k)^{-1}(U_k^T S_k^k U_k))$$  \hspace{1cm} (4.10)

which can be directly solved with the generalized eigenvalue decomposition method. However, this distortion of the objective function leads to the computational issues as detailed in the following subsection.
4.2.2 Computational Issues

As the objective function in each iteration is changed from the trace ratio form (4.9) to the ratio trace form (4.10), the deduced solution can satisfy neither of the two aspects: 1) the objective function value in (4.7) can monotonously increase; and 2) the solution \((U_1, U_2, \ldots, U_n)\) can converge to a fixed point. In this work, we present a convergent solution procedure to the optimization problem defined in (4.7).

4.3 Solution Procedure and Convergency Proof

In this section, we first introduce our new solution procedure to the tensor-based subspace learning problems, and then give the convergency proof to the two aspects mentioned above.

As described above, there does not exist closed-form solution for the optimization problem (4.7), and we solve the optimization problem also in an iterative manner. For each iteration, we refine one projection matrix by fixing the others and an efficient method is proposed for this refinement. Instead of solving a ratio trace optimization problem (4.10) for an approximate solution, we transform the trace ratio optimization problem (4.9) into a trace difference optimization problem defined as

\[
U^*_k = \arg \max_{U_k} Tr(U_k^T(S_p^k - \lambda S^k)U_k),
\]  

(4.11)

where \(\lambda\) is the value of objective function (4.7) computed from the projection matrices of the previous iteration.

Though the iterative procedure may converge to a local optimum for the optimization problem (4.7), it can monotonously increase the objective function value as proved later, which directly leads to its superiority over the ratio trace based optimization procedure, since the step-wise solution of the latter is not necessarily optimal for (4.9).
We iteratively refine the projection matrices, and the detailed solution procedure to solve the tensor-based general subspace learning problem is listed in Algorithm 2.

### 4.3.1 Analysis of Monotonous Increase Property

Rewrite the objective function of (4.7) as

\[
G(U^k_{|k=1}^n) = \frac{\sum_{i \neq j} \| (X_i - X_j) \times_k U^k_{|k=1}^n \|^2 S_{ij}^p}{\sum_{i \neq j} \| (X_i - X_j) \times_k U^k_{|k=1}^n \|^2 S_{ij}},
\]

and then we have the theory as below:

**Theorem-1.** By following the terms in Algorithm-2 and Eqn. (4.13), we have

\[
G(U^1_t, \ldots, U^{k-1}_t, U^k_t, U^{k+1}_t, \ldots, U^n_t) \leq G(U^1_t, \ldots, U^{k-1}_t, U^k_t, U^{k+1}_t, \ldots, U^n_t).
\]

**Proof.** Denote \( g(U) = Tr(U^T (S_k^p - \lambda S^k) U) \) where

\[
\lambda = G(U^1_t, \ldots, U^{k-1}_t, U^k_t, U^{k+1}_t, \ldots, U^n_t),
\]
then we have
\[ g(U^k_{t-1}) = 0. \]
Moreover, from \( U^T U = I_{m_k} \), it is easy to prove that
\[ \sup g(U) = \sum_{j=1}^{m'_k} \lambda_j. \]
From Algorithm 2, we have \( g(U^k_t) = \sum_{j=1}^{m'_k} \lambda_j \), and hence
\[ g(U^k_t) \geq g(U^k_{t-1}) = 0. \]
Then, \( Tr(U^k_t S^p_k - \lambda S^k)U^k_t \geq 0 \). As matrix \( S^k \) is positive semidefinite \(^2\), we have
\[ \frac{Tr(U^k_t S^p_k U^k_t)}{Tr(U^k_t S^k U^k_t)} \geq \lambda, \]
that is,
\[ G(U^0_{t|o=1, n|o=k}, U^0_{t-1|o=k+1}) \leq G(U^0_{t|o=1, n|o=k}, U^0_{t-1|o=k+1}) \]
From theorem-1, we can conclude that the value of the objective function monotonously increases.

4.3.2 Proof of Convergency

To prove the convergency of the projection matrices \( U^1, U^2, \ldots, U^n \), we need the concept of point-to-set map. The power set \( \varphi(\chi) \) of a set \( \chi \) is the collection of all subsets of \( \chi \). A point-to-set map \( \Omega \) is a function: \( \chi \rightarrow \varphi(\chi) \). In our solution procedure to tensor-based subspace learning, the map from \( (U^k_t|_{o=1}) \) to \( (U^k_t|_{k=1}) \) can be considered as a point-to-set map, since each \( U^k_t \) is invariant under any orthogonal transformation.

\(^2\)Though \( S^k \) may have zero eigenvalues, \( Tr(U^k_t S^k U^k_t) \) will be positive when \( m'_k \) is larger than the number of the zero eigenvalues.
Strict Monotony. An algorithm is a point-to-set map \( \Omega: \chi \to \wp(\chi) \). Given an initial point \( x_0 \), an algorithm generates a sequence of points via the rule that \( x_t \in \Omega(x_{t-1}) \). Suppose \( J: \chi \to \mathbb{R}_+ \) is a continuous, non-negative function, an algorithm is called strict monotony if 1) \( y \in \Omega(x) \) implies that \( J(y) \geq J(x) \), and 2) \( y \in \Omega(x) \) and \( J(y) = J(x) \) imply that \( y = x \).

Let set \( \chi \) be the direct sum of the orthogonal matrix space \( \mathbb{O}_{m \times m'} \), that is, the data space \( \chi = \mathbb{O}_{m_1 \times m'_1} \oplus \mathbb{O}_{m_2 \times m'_2} \oplus \ldots \oplus \mathbb{O}_{m_n \times m'_n} \), then the Algorithm 2 produces a point-to-set algorithm with respect to \( J(x) = G(U^k|_{k=1}^n) \), and it can be proved to be strictly monotononic as follows.

Theorem-2. The point-to-set map from Algorithm 2 is strictly monotonic.

Proof. From theorem-1, we have \( G(U^k_{t-1}|_{k=1}^n) \leq G(U^k_t|_{k=1}^n) \), and hence the first condition for strict monotony is satisfied. For the second condition, we take \( U^1 \) as an example to prove that this condition is also satisfied. If \( G(U^k_{t-1}|_{k=1}^n) = G(U^k_t|_{k=1}^n) \), then from the proof of theorem-1, we have \( g(U^1_{t-1}) = g(U^1_t) \) with \( \lambda = G(U^k_{t-1}|_{k=1}^n) \) and \( S^k, S^p_k \) computed from \( (U^k_{t-1}|_{k=1}^n) \). From the proof of theorem-1, we can have that there only exists one orthogonal transformation\(^3\) between \( U^1_{t-1} \) and \( U^1_t \). As shown in Algorithm 2, this kind of orthogonal transformation has been normalized by the reshaping step, hence we have \( U^1_{t-1} = U^1_t \). Similarly, we can prove that \( U^k_{t} = U^k_{t-1} \) for \( k = 1, 2, \ldots, n \), hence the second condition is also satisfied and the Algorithm 2 is strictly monotonic.

Theorem-3 [35]. Assume that the algorithm \( \Omega \) is strictly monotonic with respect to \( J \) and it generates a sequence \( \{x_t\} \) which lies in a compact set. If \( \chi \) is normed, then \( \|x_t - x_{t-1}\| \to 0 \).

From theorem-3, we can have the conclusion that the obtained \((U^k|_{k=1}^n)\) will converge to a local optimum, since the \( \chi \) is

---

\(^3\)This claim is based on the assumption that there do not exist duplicated eigenvalues in (4.11).
compact and with norm definition.

4.4 Experiments

In this section, we systematically examine the convergency properties of our proposed solution procedure to tensor-based subspace learning. We take the Marginal Fisher Analysis (MFA) as an instance of general subspace learning, since MFA has shown to be superior to many traditional subspace learning algorithms such as Linear Discriminant Analysis (LDA); more details on the MFA algorithm is referred to [60]. Then, we evaluate the classification capability of the derived low-dimensional representation from our solution procedure compared with the traditional procedure proposed in [63] and [59]. For tensor-based algorithm, the image matrix, $2^d$ tensor, is used as input, and the image matrix is transformed into the corresponding vector as the input of vector-based algorithms.

4.4.1 Data Sets

Three real-world data sets are used. One is the USPS handwritten dataset \(^4\) of 16-by-16 images of handwritten digits with pixel values ranging between -1 and 1. The other two are the benchmark face databases, ORL and CMU PIE \(^5\). For the face databases, affine transform is performed on all the samples to fix the positions of the two eyes and the mouth center. The ORL database contains 400 images of 40 persons, where each image is normalized to the size of 56-by-46 pixels. The CMU PIE (Pose, Illumination, and Expression) database contains more than 40,000 facial images of 68 people. In our experiment, a subset of five near frontal poses (C27, C05, C29, C09 and C07)

---

\(^4\) Available at: http://www-stat-class.stanford.edu/tibs/ElemStat-Learn/data.html

\(^5\) Available at http://www.face-rec.org/databases/.
and illuminations indexed as 08 and 11 are used and normalized to the size of 32-by-32.

Figure 4.2: The step difference of the projection matrices vs. iteration number. (a,d) USPS database, (b,e) ORL database, and (c,f) CMU PIE database.

4.4.2 Monotonicity of Objective Function Value

In this subsection, we examine the monotonicity property of the objective function value from our solution procedure compared with the optimization procedure that step-wisely transforms the objective function into the ratio trace form. The USPS, ORL and PIE databases are used for this evaluation. The detailed results are shown in Figure 4.1. It is observed that the traditional ratio trace based procedure does not converge, while our new solution procedure guarantees the monotonous increase of the objective function value and commonly our new procedure will converge after about 4-10 iterations. Moreover, the final converged value of the objective function from our new procedure
is much larger than the value of the objective function for any iteration of the ratio trace based procedure.

### 4.4.3 Convergency of the Projection Matrices

To evaluate the solution convergence property compared with the traditional ratio trace based optimization procedure, we calculate the difference norm of the projection matrices from two successive iterations and the detailed results are displayed in Figure 4.2. It demonstrates that the projection matrices converge after 4-10 iterations for our new solution procedure; while for the traditional procedure, heavy oscillations exist and the solution does not converge. As shown in Figure 4.3, the recognition rate is sensitive to the oscillations caused by the unconvergent projection matrices and the classification accuracy is degraded dramatically.

![Figure 4.3: Recognition error rate (%) vs. iteration number. (a) ORL database(G4P6), and (b) CMU PIE database(G4P6).](image)

**Figure 4.3:** Recognition error rate (%) vs. iteration number. (a) ORL database(G4P6), and (b) CMU PIE database(G4P6).

### 4.4.4 Face Recognition

In this subsection, we conduct classification experiments on the benchmark face databases. The Tensor Marginal Fisher Analysis algorithm based on our new solution procedure (TMFA_TR)
Table 4.1: Recognition error rates (%) on the ORL database.

<table>
<thead>
<tr>
<th>Method</th>
<th>G3P7</th>
<th>G4P6</th>
<th>G5P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o DR.</td>
<td>28.57</td>
<td>24.17</td>
<td>21.5</td>
</tr>
<tr>
<td>LDA</td>
<td>17.86</td>
<td>17.08</td>
<td>11.00</td>
</tr>
<tr>
<td>MFA_RT</td>
<td>17.50</td>
<td>16.25</td>
<td>10.50</td>
</tr>
<tr>
<td>MFA_TR</td>
<td>13.93</td>
<td>10.00</td>
<td>6.50</td>
</tr>
<tr>
<td>TMFA_RT</td>
<td>12.14</td>
<td>11.67</td>
<td>5.00</td>
</tr>
<tr>
<td>TMFA_TR</td>
<td>11.07</td>
<td><strong>6.67</strong></td>
<td><strong>4.00</strong></td>
</tr>
</tbody>
</table>

Table 4.2: Recognition error rates (%) on the PIE database.

<table>
<thead>
<tr>
<th>Method</th>
<th>G3P7</th>
<th>G4P6</th>
<th>G5P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o DR.</td>
<td>49.89</td>
<td>31.75</td>
<td>30.16</td>
</tr>
<tr>
<td>LDA</td>
<td>18.82</td>
<td>19.84</td>
<td>18.10</td>
</tr>
<tr>
<td>MFA_RT</td>
<td>16.55</td>
<td>15.61</td>
<td>13.65</td>
</tr>
<tr>
<td>MFA_TR</td>
<td>14.97</td>
<td>13.49</td>
<td>9.52</td>
</tr>
<tr>
<td>TMFA_RT</td>
<td>14.74</td>
<td>14.29</td>
<td><strong>3.81</strong></td>
</tr>
<tr>
<td>TMFA_TR</td>
<td><strong>13.61</strong></td>
<td><strong>12.17</strong></td>
<td>9.52</td>
</tr>
</tbody>
</table>

is compared with the traditional ratio trace based Tensor Marginal Fisher Analysis (TMFA_RT), LDA, Ratio Trace based MFA (MFA_RT) and Trace Ratio based MFA (MFA_TR), where MFA_TR means to conduct tensor-based MFA by assuming $n=1$. To speed up model training, PCA is conducted as a preprocess step for vector-based algorithms. The PCA dimension is set as $N-N_c$ ($N$ is the sample number and $N_c$ is the class number), which is equivalent to the case for Fisherface algorithm [3]. The same graph configuration with nearest neighbor $k = 3$ for the intrinsic graph and $k_p = 40$ for the penalty graph is adopted for all the MFA based algorithms. Since the traditional tensor subspace learning algorithms do not converge, we terminate the process after 3 iterations.
For comparison, the classification result on the original gray-level features without dimensionality reduction is also reported as the baseline, denoted as 'w/o DR.' in the result tables. In all the experiments, the Nearest Neighbor method is used for final classification. All possible dimensions of the final low-dimensional representation are evaluated, and the best results are reported. For each database, we test various configurations of training and testing sets for the sake of statistical confidence, denoted as 'GxPy' for which \( x \) images of each subject are randomly selected for model training and the remaining \( y \) images of each subject are used for testing. We selected representative frontal faces as gallery images and the rest serve as the probe images. To overcome the effect caused by local optimum, we report the best results over 3 independent runs for the tensor based algorithms. The detailed results are listed in Table 4.1 and 4.2. From these results, we can have the following observations:

1. TMFA_TR mostly outperforms all the other methods concerned in this work, with only one exception for the case \( G5P5 \) on the CMU PIE database.

2. For vector-based algorithms, the trace ratio based formulation (MFA_TR) is consistently superior to the ratio trace based one (MFA_RT) for subspace learning.

3. Tensor representation has the potential to improve the classification performance for both trace ratio and ratio trace formulations of subspace learning.

4.5 Conclusions

In this Chapter, a novel iterative procedure was proposed to directly optimize the objective function of general subspace learning based on tensor representation. The convergence of the projection matrices and the monotonicity property of the objective
function value were proven. To the best of our knowledge, it is the first work to give a convergent solution for general tensor-based subspace learning.

End of chapter.
Algorithm 2. Procedure to Tensor-based Subspace Learning

1: **Initialization.** Initialize $U_0^1, U_0^2, \ldots, U_0^n$ as arbitrary column orthogonal matrices.

2: **Iterative optimization.**
   
   For $t=1,2,\ldots, T_{\text{max}}$, Do
   
   For $k=1,2,\ldots, n$, Do
   
   1. Set $\lambda = \frac{\sum_{i \neq j} \| (X_i - X_j) \times_o U_{k-1}^1 \times_o U_{k-1}^n \|_2^2 S_{ij}^p}{\sum_{i \neq j} \| (X_i - X_j) \times_o U_{k-1}^1 \times_o U_{k-1}^n \|_2^2 S_{ij}^p}$.
   
   2. Compute $S_k^p$ and $S_k^p$ as in (4.9) based on the projection matrices $U_1^t, \ldots, U_{k-1}^t$ and $U_{k+1}^{t-1}, \ldots, U_{n-1}^{t-1}$.
   
   3. Conduct Eigenvalue Decomposition:
      
      $$(S_k^p - \lambda S_k^p)v_j = \lambda_j v_j, \quad j = 1, \ldots, m'_k,$$
      
      where $v_j$ is the eigenvector corresponding to the $j$-th largest eigenvalue $\lambda_j$.
   
   4. Reshape the projection directions for the sake of orthogonal transformation invariance:
      
      (a) Set $V = [v_1, v_2, \ldots, v_{m'_k}]$;
      
      (b) Let $S^v = V V^T (\sum_i X_i^k X_i^T) V V^T$, where $X_i^k$ is the mode-$k$ unfolding of the tensor $X_i$;
      
      (c) Conduct Eigenvalue Decomposition as
          
          $$S^v u_i = \gamma_i u_i. \quad (4.12)$$
   
   5. Set the column vectors of matrix $U_k^t$ as the leading eigenvectors, namely, $U_k^t = [u_1, u_2, \ldots, u_{m'_k}]$.
   
   End
   
   If $\| U_k^t - U_{t-1}^k \| < \sqrt{m_k m'_k} \varepsilon, k = 1, 2, \ldots, n$ ($\varepsilon$ is set to $10^{-4}$ in this work), then break.
   
End

3: **Output the projection matrices** $U_k^k = U_k^t, k=1,2,\ldots,n$. 

Chapter 5

Maximum Unfolded Embedding

Summary

Graph is a powerful tool to model pairwise sample relations and the intrinsic manifold structures of sample distributions. In this chapter, we propose a spectral clustering algorithm employing graph to model the sample manifold structures and the clustering objective is formulated as a trace ratio, which is then optimized using the procedure we proposed in Chapter 3.

5.1 Introduction

With the rapid development of internet and storage hardware, larger and larger image repositories are being built. As a result, image search and retrieval are becoming more and more popular. Image clustering is a powerful high-level technique that helps analyze the image content of a large database [16].

The aim of image clustering is to find a mapping from the archive images to a set of label numbers. Traditional clustering
methods, such as K-means [21] and Gaussian Mixture Model [21], directly operate on the original feature space. They often cannot produce satisfactory results due to the underlying complex data distribution, and many researches show that the image data may lie on a nonlinearly distributed manifold [67].

The spectral clustering algorithms [5][19] utilize a graph to characterize the relationship of the input data. By relaxing the cluster labels to the real value domain, they apply the efficient eigen-decomposition method for data embedding. The most popular spectral clustering algorithm is normalized cut [30], which has been widely used in various applications [37][66]. An extension of the normalized cut was proposed by Ng et al. [2], which deals with the clustering problem in a more effective way.

The spectral embedding algorithms Laplacian Eigenmap (LE) [5] and Locality Preserving Projections (LPP) [27] have shown their effectiveness for image clustering [67]. The disadvantage of LE and LPP is that their objective functions for embedding cannot ensure the separability of those faraway point pairs in the derived low-dimensional feature space, which may greatly degrade the performance of the consequent clustering process.

To overcome the above disadvantage, we propose a novel spectral embedding algorithm called Maximum Unfolded Embedding (MUE), and consequently a new clustering algorithm, Maximum Unfolded Clustering (MUC), is presented by utilizing K-means on the derived embedding. The aim of MUE is two-fold: one is to preserve the local manifold structure, and the other is to maximize the distance between the faraway sample pairs. These dual objectives are integrated to form a Trace Ratio optimization problem. We then utilized the procedure proposed in the previous chapters to directly optimize the trace ratio, instead of transforming the Trace Ratio optimization problem into a Ratio Trace optimization problem which has the advantage of
a closed form solution yet sacrifices the potential clustering capability. Moreover, the linear extension of MUE is also presented so as to map the out-of-sample data into the low-dimensional feature space.

5.2 Maximum Unfolded Embedding

An image set is often considered lying on a compact nonlinear manifold embedded in a high-dimensional Euclidean space. Image clustering directly performed on the original image feature space may suffer from the curse of dimensionality, and it is desirable to conduct dimensionality reduction before the clustering process. In this work, we propose a novel dimensionality reduction algorithm called Maximum Unfolded Embedding (MUE), for image clustering by taking into account the underlying manifold structure of the image set.

For a given data set, we construct a weighted adjacency graph $G^a = (V, W^a)$ by connecting each point with its $k_n$ nearest neighboring points, and a weighted separation Graph $G^s = (V, W^s)$ that indicates the faraway point pairs measured in the original feature space and connects each point with its $k_f$ farthest contrary points. Without loss of generality, we assume these two graphs are both connected. The element of the weight matrix equals to 1 when there is an edge connecting the corresponding two vertices; otherwise 0.

The aim of MUE is to find a low-dimensional representation such that the connected vertices are close to one another and the faraway vertex pairs are faraway to each other. Let an $N \times m$ matrix $Y = [y_1, y_2, ..., y_m]$ denote such a low-dimensional representation ($N$ denotes the sample number and $m$ is the desired reduced feature dimension). The $i$-th row corresponds to the $i$-th dimension of the mapped data. To embrace the above two
objectives, the following optimization problem is posed in MUE:

\[
Y = \underset{Y^T Y = I}{\text{argmax}} \sum_{ij} \|y^{(i)} - y^{(j)}\|^2 W_{ij}^s/\sum_{ij} \|y^{(i)} - y^{(j)}\|^2 W_{ij}^a.
\] (5.1)

Here \(y^{(i)} = [y_1^{(i)}, ..., y_m^{(i)}]\) is the \(m\) dimensional representation of the \(i\)-th vertex. It is easy to have the following theorem:

**Theorem 1.** The problem defined in (5.1) is equal to a Trace Ratio optimization problem:

\[
Y = \underset{Y^T Y = I}{\text{argmax}} \frac{Tr(Y^T L^s Y)}{Tr(Y^T L^a Y)},
\] (5.2)

where \(L^a\) and \(L^s\) are the Laplacian matrices of the adjacency graph and separation graph respectively. That is, \(L^a = D^a - W^a\) and \(L^s = D^s - W^s\), where \(D^a\) and \(D^s\) are both diagonal and their diagonal elements are the row sums of the matrices \(W^a\) and \(W^s\) respectively.

### 5.3 Optimize Trace Ratio

Since the objective is a trace ratio, we can utilize the procedure proposed in Chapter 3 to derive a global optimal solution.

**Clustering:** Based on the derived low-dimensional representation, the traditional K-means algorithm can be used to conduct further clustering process in a much faster manner.

### 5.4 Another Justification: Maximum Variance Embedding

To preserve local information, LE computes a nonlinear embedding with spectral decomposition method, which has shown to be an approximation of the Laplace Beltrami operator on the manifold [5]. Merely minimizing the objective function \(y^T L^a y\) in
LE may encourage consistent output for the neighboring samples in the input space.

Compared with LE, the objective function of MUE exerts additional power on the faraway point pairs and thus prevents the mapping from collapsing into one point. A special case is that the separation matrix $W_s$ can be constructed by utilizing $k_f=N-1-k_n$ farthest points for each sample respectively, and then the objective function becomes

$$Y = \arg \max_{Y} \frac{\sum_{ij} \|y^{(i)} - y^{(j)}\|^2 \tilde{W}_{ij}^a}{\sum_{ij} \|y^{(i)} - y^{(j)}\|^2 W_{ij}^a},$$

(5.3)

with $\tilde{W}_{ij}^a = 1 - W_{ij}^a$.

Summing up the numerator and the denominator, we can rewrite the objective function (5.1) as:

$$\frac{\sum_{ij} \|y^{(i)} - y^{(j)}\|^2}{\sum_{ij} \|y^{(i)} - y^{(j)}\|^2 W_{ij}^a} = \frac{Tr(Y^T L_{cp} Y)}{Tr(Y^T L_{a} Y)},$$

(5.4)

where $L_{cp}$ is the Laplacian matrix of the Binary Complete Graph which connects all the samples with equal weights. It means that the objective function is to maximize the global variance (numerator) and preserve the local characteristic (denominator) simultaneously. In this case, the proposed algorithm has the property of maximum variance, and hence may be justified as a Maximum Variance Embedding.

### 5.5 Linear Extension: Maximum Unfolded Projection

To provide the low-dimensional representation for the out-of-sample data, a simple but effective way is to assume that there exists a linear relationship between the original data and the embedding, i.e., $Y^T = U^T X$, where the $i$-th column vector of
X is $x_i$. With the constraint of column orthogonality on $U$, the optimal embedding for the objective function (5.1) is then transformed into the following trace ratio optimization problem:

$$U = \max_{U^TU=I} \frac{Tr(U^T X L^s X^T U)}{Tr(U^T X L^a X^T U)}.$$  \hspace{1cm} (5.5)

It can also be solved with the iterative procedure discussed above.

5.6 Experiments

In this section we present a set of experiments to verify the effectiveness of our proposed algorithms for image embedding and clustering. For comparison, we also report the experimental results from K-means, PCA Kmeans, normalized cut (NCut) and Locality Preserving Clustering (LPC).

5.6.1 Data set

The algorithmic performance is evaluated on the general-purpose image database, a subset of COREL [1], which contains 10,000 photo images from 79 categories. The image number for each cluster varies from 100 to 300.

We make a combination of the Color Histogram and Color Texture Moment (CTM) [65] for image description. As an extension to color moments that characterize the color texture distribution of the image, CTM provides a 48 dimensional feature vector. For Color Histogram, we set the bin number in HSI (Hue, Saturation and Intensity) space as $4 \times 4 \times 4$. Thus for each image we obtain a 112 dimensional feature vector, which is normalized in the preprocessing step so that the norm of each vector is 1.
5.6.2 Evaluation Metric

We utilize the normalized mutual information as the evaluation metric for clustering accuracy. Mutual information, as a measure for the mutual dependence of two variables in information theory, is defined as:

\[
I(X, Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}.
\]

An advantage of the mutual information is that it stays invariant when there exist permutations between the clustering results and the ground truth labels. The normalized mutual information \(\bar{I}\) is defined as

\[
\bar{I}(X, Y) = \frac{I(X, Y)}{\max(H(X), H(Y))}.
\]

Where \(H(X)\) and \(H(Y)\) are the entropy of the set \(X\) and \(Y\) respectively. It is clear that the range of \(\bar{I}\) is \([0,1]\).
Table 5.1: Clustering accuracies of MUC, NCut, PCA-Kmeans and K-Means in the cases with different cluster numbers.

<table>
<thead>
<tr>
<th>k</th>
<th>MUC</th>
<th>NCut</th>
<th>LPC</th>
<th>PCA-Km.</th>
<th>K-Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.9022</td>
<td>0.8337</td>
<td>0.7943</td>
<td>0.7367</td>
<td>0.7367</td>
</tr>
<tr>
<td>3</td>
<td>0.5511</td>
<td>0.5407</td>
<td>0.4859</td>
<td>0.3815</td>
<td>0.3755</td>
</tr>
<tr>
<td>4</td>
<td>0.5535</td>
<td>0.4744</td>
<td>0.5026</td>
<td>0.3561</td>
<td>0.3481</td>
</tr>
<tr>
<td>5</td>
<td>0.4450</td>
<td>0.3960</td>
<td>0.4132</td>
<td>0.3315</td>
<td>0.3447</td>
</tr>
<tr>
<td>6</td>
<td>0.3730</td>
<td>0.3462</td>
<td>0.3447</td>
<td>0.3103</td>
<td>0.3075</td>
</tr>
<tr>
<td>7</td>
<td>0.3382</td>
<td>0.3016</td>
<td>0.2798</td>
<td>0.2760</td>
<td>0.2748</td>
</tr>
<tr>
<td>8</td>
<td>0.3286</td>
<td>0.3137</td>
<td>0.2676</td>
<td>0.2976</td>
<td>0.2739</td>
</tr>
<tr>
<td>9</td>
<td>0.3432</td>
<td>0.3310</td>
<td>0.2646</td>
<td>0.2784</td>
<td>0.2818</td>
</tr>
<tr>
<td>10</td>
<td>0.3381</td>
<td>0.3125</td>
<td>0.2851</td>
<td>0.2766</td>
<td>0.2620</td>
</tr>
</tbody>
</table>

5.6.3 Performance Comparison

We vary the cluster number $k$ from 2 to 10 and altogether five algorithms are compared and evaluated: MUC, NCut, LPC, PCA Kmeans and K-means. All these algorithms utilize the K-means method for final clustering, either on the feature domain or on the spectral domain. To overcome the influence of the local convergence, we perform the K-means process 20 times and report the best performance. For PCA-Kmeans, Principal Component Analysis (PCA) [38] is designed to retain 95% of the total energy and the algorithm proposed by Ng et al. [2] is adopted for NCut. We make an exhaustive search over the reduced feature dimensions and report the best performance. The comparison performances are demonstrated in Table 5.1. From these results, we can see that the proposed MUC algorithm consistently obtains the highest clustering accuracy, followed by NCut and LPC. The accuracy of PCA Kmeans is lower; yet is still higher than that of direct K-Means on original features in most cases, since certain amount of noise has been removed by PCA. We can also observe a clear gap between the spectral clustering algorithms and the PCA Kmeans. As the cluster number increases, the perfor-
mance of LPC approaches that of the K-Means, mainly because the linear separability decreases with the increase of the cluster number; while MUC does not have this limitation. For a better understanding of MUE, the embedding visualization is displayed in Figure 5.1 and some clustering results are shown in Figure 5.2.

Figure 5.2: Some clustering results from 6 clusters

5.6.4 Generalization Capability

As described in Section 5, MUE can be linearized to construct the Maximum Unfolded Projections (MUP), and correspondingly we call the new clustering algorithm as MUP Clustering (MUPC). One merit of MUP is that the linearization process facilitates us with the generalization capability from training set to testing set.

Under the assumption that the training set and the testing set share similar data distributions, the projection derived from the training set can be extended to the testing set, thus we need
not implement the Maximum Unfolded Embedding on the entire data set.

The second experiment is designed to demonstrate the generalization capability of the proposed algorithm. The data set contains 1000 images. We vary the percentage of training samples and evaluate the performance of MUPC on different training sets. For a comparison, the clustering result from MUC is also presented. Two configurations are adopted in this experiment. One is that we implement the algorithm on the entire data set and test the performance on the testing data set, which is denoted by MUPC (G) and MUC (G). The other configuration, MUPC (T), is as follows: first the projection is learned from the training set; then the embedding projection is performed on the testing set and K-means clustering algorithm is carried out; finally the normalized mutual information is calculated and compared among different configurations on the testing set. Figure 5.3 displayed the generalization results.

It is observed that as the percentage of training samples increases, the performance on the testing set is improved. We also notice that 30% of the total samples will be enough for the
MUPC to produce a satisfying clustering accuracy and MUPC (G) can achieve comparable results as the MUC (G) with a much lower computational cost.

5.7 Conclusion

To maximize the distance of faraway sample pairs and simultaneously preserve the locality of the image manifold, a novel spectral embedding algorithm, Maximum Unfolded Embedding, was formulated. Then, the previously proposed iterative procedure is utilized for the trace ratio optimization. Consequently, a new spectral clustering algorithm was finalized by integrating the derived embedding and the K-Means approach. Extensive experiments verified the effectiveness of the proposed algorithm and its linearized version.

☐ End of chapter.
Chapter 6

Regression on MultiClass Data

Summary

Besides the supervised discriminant analysis, unsupervised clustering, a recent trend in machine learning is the semisupervised framework, in which samples are only partially labeled. In this chapter, we demonstrate that the manifold structures of samples can also be explored to guide the regression in the semisupervised configuration. Specifically, we focus on the semisupervised regression problem on multiclass samples and labels of different sample classes are employed to pilot the regression.

6.1 Introduction

Large scale and high dimensional data are ubiquitous in real-world applications, yet the processing and analysis of these data are often difficult due to the curse of dimensionality as well as the high computational cost involved. Usually, these high dimensional data lie approximately on an underlying compact low dimensional manifold, which may turn the problem tractable. Substantive works have been devoted to unveiling the intrinsic
structure of the manifold data, among which the popular ones include ISOMAP [44], LLE [39], Laplacian Eigenmap [6], and MVU [53].

Besides these unsupervised algorithms that purely explore the manifold structure of the data, researchers have also well utilized the latent manifold structure information from both labeled and unlabeled samples to enhance learning algorithms with limited number of labeled samples. Great success has been achieved in various areas, such as classification [32] [7], manifold alignment [25] and regression [4] [68]. Recently increasing attention has been drawn to the semisupervised regression problem by considering the manifold structure. [14] proposed a transductive algorithm minimizing the leave-one-out error of the ridge regression on the joint set composed from both labeled and unlabeled data. To exploit the manifold structure, [4] adds a graph Laplacian regularization item to the regression objective, which imposes extra condition on the smoothness along the data manifold and it has proved to be quite useful in applications. [17] first roughly transduces the function values from the labeled data to the unlabeled ones utilizing local neighborhood relations, and then optimizes the global objective that best fits the labels of the training points as well as the estimated labels provided by the first step.

None of these state-of-the-art regression algorithms make use of the class information to guide the regression. In real-world applications, however, multi-class samples are ubiquitous and samples from different classes can be regarded as lying on multiple manifolds. For example, in age estimation, the personal aging process varies differently for different genders, but there still exists consistency between the aging processes of male and female. In the pose estimation from images of multiple persons, each person can be regarded as one class and the images vary similarly with poses from different persons. Moreover, we
may also encounter the multi-modality samples whose representations may not be consistent due to the lack of correspondence, while the inner manifold configuration for the label distribution across different modalities may still be similar. Besides, in the function learning stage of the semisupervised regression framework, the class or modality information is usually easy to obtain, thus it is desirable to utilize this information to further boost the regression accuracy. To fully exploit the relations among data manifolds, we develop in this Chapter a semisupervised regression algorithm called *TRIM* (Transductive Regression piloted by Inter-Manifold Relations). *TRIM* is based on the assumption that the data of different classes share similar configuration for label distributions. In our proposed algorithm, manifolds of different classes are aligned according to the landmark connections constructed from both the sample label distance and manifold structural similarity, and then the labels are transduced across different manifolds based on the alignment output. Besides the intra-manifold smoothness condition within each class, our method introduces an inter-manifold regularization item and employs the transduced labels from various data manifolds to pilot the trend of the function values over all the samples. In addition, the function to be learnt can be approximated by the functionals lying on the Reproducing Kernel Hilbert Space (RKHS), and the regression on the induced RKHS directly leads to an efficient solution for the label prediction of the out-of-sample data even without corresponding class information.

6.2 Background

6.2.1 Intuitive Motivations

Consider the problem of adult height estimation based on the parents’ heights. Dramatic difference exists between male and
female adults, while the general configuration of the height distribution within each gender may still be similar: the taller the parents are, the higher the children are supposed to be. While once we mix up the data from both genders, the height distribution may become complicated. Figure 7.1 displays a toy example for the obstacles encountered in the multi-class regression problem. We have two classes of samples and the function values within each class manifold share similar configurations, while the number of samples and labels may vary across classes. As is shown in the upper part of Figure 7.1, it is difficult to get a satisfactory regression if we put all the samples together due to the complex structures produced by the manifold intersections. On the other hand, if the regression is carried out separately on different classes, the accuracy could still be low due to the lack of sufficient labeled samples for each class. Also, for the incoming out-of-sample data, the prediction cannot be done without class information. Moreover, we may face some regression problems with multi-modality samples, e.g., the estimation of human ages from both photos and sketch images. It is meaningless to compose data of different modalities into one manifold since the semantics of the features from different modalities are essen-
tially different, while it is still possible that the intra-modality label configurations are similar. In this Chapter we focus on the multi-class regression problem, but our framework can also be easily extended to give predictions for the multi-modality data.

6.2.2 Related Work

Multi-view learning [10] technique can also be applied to the regression problem with multi-class data. Denote the labeled and unlabeled instances as $x^l \in \mathcal{X}^l \subseteq \mathcal{X}$ and $x^u \in \mathcal{X}^u \subseteq \mathcal{X}$ respectively, where $\mathcal{X}^l, \mathcal{X}^u, \mathcal{X}$ are the labeled, unlabeled and whole sample sets. State-of-the-art semisupervised multi-view learning frameworks employ multiple regression functions from the Hilber space $\mathcal{H}$ so that the estimation error on the training set and disagreement among the functions on the unlabeled data are minimized [12], i.e.,

$$\tilde{f}_{v \mid M_l} = \min_{f_v \in \mathcal{H}} \sum_{v=1}^{M_l} \left( \sum_{x \in \mathcal{X}^l} c(f_v(x^l), y(x^l)) + \gamma \|f_v\|^2 \right) + \lambda \sum_{v_1, v_2=1}^{M_l} \sum_{x \in \mathcal{X}^u} c(f_{v_1}(x^u) - f_{v_2}(x^u)), \quad (6.1)$$

where $y(x)$ is the sample label, $f_{v \mid M_l}$ are $M_l$ multiple learners and $c(\cdot)$ is a cost function.

We would like to highlight beforehand some properties of our framework compared to the multi-view learning algorithms:

1. There exists a clear correspondence between the multi-view data from the same instance in the multi-view learning framework, while our algorithm does not require the correspondence among different manifolds. The data of different modalities or classes may be obtained from different instances in our configuration, thus it is much more challenging to give an accurate regression.
2. The class information is utilized in two ways:
   a) Sample relations within each class are coded by intra-manifold graphs and a corresponding regularization item is introduced to ensure the within-class smoothness separately.
   b) A set of inter-manifold graphs are constructed from the cross manifold label propagation on the aligned manifolds and an inter-manifold regularization item is proposed to fully exploit the information conveyed among different classes.

3. The class information is used in the function learning phase but no class attributes are required for the out-of-sample data in the prediction stage.

6.3 Problem Formulation

6.3.1 Notations

Assume that the whole sample set $X$ consists of $N$ samples from $M$ classes, denoted as $X^1, X^2, ..., X^M$. For each class $X^k$, $N^k = l^k + u^k$ samples are given, i.e.,

$$X^k = \{(x^k_1, y^k_1), ..., (x^k_{l^k}, y^k_{l^k}), (x^k_{l^k+1}, y^k_{l^k+1}), \ldots, (x^k_{l^k+u^k}, y^k_{l^k+u^k})\},$$

where $Y^k_l = [y^k_1, y^k_2, ..., y^k_{l^k}]^T$ represents the function values with respect to the given labeled samples and $Y^k_u = [y^k_{l^k+1}, ..., y^k_{l^k+u^k}]^T$ corresponds to the function values of the remaining unlabeled data to be estimated in the $k^{th}$ class. In this paper, the 'label' refers to a real value to be regressed.

Let $G^k = (V^k, E^k)$ denote the intra-class graph with vertex set $V^k$ and edge set $E^k$ constructed within the data of the $k^{th}$ class. Here we focus on the undirected graph and it is easy to generalize our algorithm for directed graphs. The edges in $E^k$ reflect the neighborhood relations along the manifold data,
which can be defined in terms of $k$-nearest neighbors or an $\epsilon$-ball distance criterion in the sample feature space $\mathcal{F}$. One choice for those non-negative weights on the corresponding edges is to use the heat kernel [6] or the inverse of feature distances [17], i.e.,

$$w_{ij} = e^{-\frac{\|\Phi^k(x_i) - \Phi^k(x_j)\|^2}{t}}$$

or

$$w_{ij} = \|\Phi^k(x_i) - \Phi^k(x_j)\|^{-1},$$

where $t \in \mathbb{R}$ is the parameter for the heat kernel and $\Phi^k(\cdot)$ is a feature mapping from $\mathcal{X}$ to the normed feature vector space $\mathcal{F}$ for the sample of the $k^{th}$ class. For samples from different modalities/manifolds, the feature mappings may be different. The other one is to solve a least-square problem to minimize the reconstruction error and get the weights $w_{ij}$:

$$w_{ij} = \arg \min_{w_{ij}} \|x_i - \sum_j w_{ij}x_j\|^2,$$

$$\text{s.t. } \sum_j w_{ij} = 1, w_{ij} \geq 0$$

[39].

To encode the mutual relations among different sample classes, we also introduce the inter-manifold graph, denoted as a triplet $G^{k_i,k_j} = (V^{k_i}, V^{k_j}, E^{k_i,k_j})$. The inter-manifold graph $G^{k_i,k_j}$ is a bipartite graph with one vertex set from the $k_i^{th}$ class and the other from the $k_j^{th}$. The construction of $G^{k_i,k_j}$ will be discussed in the following subsections.

### 6.3.2 Regularization along Data Manifold

Now we are given the sample data of multiple classes, denoted as $\mathcal{X}$, including both labeled and unlabeled samples $\mathcal{X}^l$ and $\mathcal{X}^u$. The manifold structures within each class data are encoded by the intra-class graph. [4] introduced a manifold regularization item for the semisupervised regression framework, i.e., the graph
Laplacian, which is expected to impose smoothness conditions along the manifold on the possible solutions. The final formulation seeks a balance between the fitting item and the smoothness regularization, i.e.,

$$\tilde{f} = \arg\min \frac{1}{l} \sum_{i} (f_i - y_i)^2 + \gamma f^T L^p f,$$

(6.2)

where $p \in \mathbb{N}$. When $p = 1$, the regularization item is the graph Laplacian and for $p = 2$, the regularization turns out to be the 2-norm of the reconstruction error when the weights $w_{ij}$ are normalized, i.e.,

$$f^T L^T L f = \sum_{i} \| f_i - \sum_j w_{ij} f_j \|^2$$

w.r.t. $\sum_j w_{ij} = 1, w_{ij} \geq 0$.

### 6.3.3 Cross Manifold Label Propagation

In our configuration, there does not exist a clear correspondence among the manifolds of different classes or modalities, and even the representations for different modalities can be distinct. Thus it is rather difficult to construct sample relations directly from the similarity in the sample space $X$. Alternatively, the function labels still convey some correspondence information, which may be utilized to guide the inter-manifold relations. Moreover, the manifold structure also contains some indications about the correspondence. Thus we first seek a point-to-point correspondence for the labeled data combining the indications provided by both the sample labels and the manifold structures. This is done under two assumptions:

1. Samples with similar labels lie generally in similar relative positions on the corresponding manifold.
2. Corresponding sample sets tend to share similar graph structures on the respective manifold.

**Reinforced Landmark Correspondence**

First we search for a set of stable landmarks to guide the manifold alignment. Specifically, we use the \(\epsilon\)-ball distance criterion on the sample labels to initialize the inter-manifold graphs. To give a robust correspondence, we reinforce the inter-manifold connections by iteratively implementing

\[
W_{ijk}^k \leftarrow W_{ik}^k \times W_{jk}^k \times W_{kj}^k.
\]  

(6.3)

Similar to the similarity propagation process on the directed graph [9], (6.3) reinforces the similarity score of sample pairs by the similarity of its neighbor pairs, i.e.,

\[
w_{ij}^{k,i} \leftarrow \sum_{m,n} w_{im}^{k,i} w_{mn}^{k,j} w_{nj}^{k,j}.
\]  

(6.4)

The assumption here is that two nodes tend to be similar if they have similar neighbors. (6.4) utilizes the intra-manifold structure information to reinforce the inter-manifold similarity and thus can generate a more robust correspondence. The accuracy of the landmark correspondence is critical for our algorithm. To ensure a robust performance, only the correspondences with the top 20% largest similarity scores are selected as the landmarks and it is common that some classes may miss certain sample labels, so plenty of labeled samples remain unmatched.

**Manifold Alignment**

To propagate the sample labels to the unlabeled points across manifolds, we ‘stretch’ all the manifolds with respect to the landmark points obtained from the previous step and this can be realized by the semisupervised manifold alignment [25].
In the manifold alignment process, we seek an embedding that minimizes the correspondence error on the landmark points and at the same time keeps the intra-manifold structures, i.e.,

\[
\begin{align*}
f^{k_i | M}_{k_i=1} &= \arg \min \left( \sum_{k_i} C(f^{k_i | M}_{k_i=1}) \right), \\
D^{k_i} &= \text{the diagonal degree matrix, } L^{k_i} = \text{the graph Laplacian matrix for the } k_i^{th} \text{ class and } p \in \mathbb{N}. \text{ To ensure the inter-class adjacency, we add a global compactness regularization item } \beta f^T L^a f \text{ to the cost function } C, \text{ where } L^a \text{ is the Laplacian Matrix of } W^a \text{ with } \quad w^a_{ij} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ are of different classes} \\ 0 & \text{o.w.} \end{cases}
\end{align*}
\]

The labels are propagated across manifolds on the derived aligned manifolds using the nearest neighbor approach, i.e., we connect the labeled samples with the nearest points from other
Algorithm 3 Procedure to construct inter-manifold connections

1: Inter-manifold graph initialization

\[ w_{ij}^{k_i k_j} = \begin{cases} 
1 & \text{if } \|y_i^k - y_j^k\|^2 < \epsilon \\
0 & \text{o.w.} 
\end{cases} \]

2: Correspondence Reinforcement

for \( \text{Iter} = 1 : N_{\text{Iter}} \)

\[ W^{k_i k_j} = W^{k_i} \times W^{k_i k_j} \times W^{k_j} \]

end

3: Landmark Selection: Select the sample pairs with the top 20% largest similarity scores as correspondence. Set the corresponding elements in \( W^{k_i k_j} \) as 1 and others 0.

4: Manifold Alignment using the Inter-manifold Graphs \( W^{k_i k_j} \).

5: Find the corresponding points from different classes for the unmatched labeled samples using the nearest neighbor approach in the aligned space and update the inter-manifold graphs \( W^{k_i k_j} \).

classes on the aligned space. The derived inter-manifold graphs are concatenated to form

\[
W^r = \begin{pmatrix}
O & W^{12} & \cdots & W^{1M} \\
W^{21} & O & \cdots & W^{2M} \\
\vdots & \vdots & \ddots & \vdots \\
W^{M1} & W^{M2} & \cdots & O
\end{pmatrix}, \quad (6.7)
\]

which is then symmetrized and employed in the following Inter-Manifold regularization item.

6.3.4 Inter-Manifold Regularization

We rearrange the samples to place the data from the same class together and put the labeled points first for each class, that is,

\[
\mathcal{X} = \{x_1, x_2, \ldots, x_l, x_{l+1}, \ldots, x_{l+u_1}, x_1, x_2, \ldots, x_{l_2}, x_{l_2+1}, \\
\ldots, x_{l_2+u_2}, \ldots, x_1, x_2, \ldots, x_M, x_{M+1}, \ldots, x_{M+u_M} \}. 
\]
Denote the corresponding function values as $f$, which is a concatenation of $f^k = [f^k_{x^1_i}, ..., f^k_{x^L_i}]^T$ from all the classes. Our regression objective is defined as:

$$\tilde{f} = \arg \min_f \sum_k \frac{1}{l_k} \sum_{x^i_k \in \mathcal{X}} \|f^k_{x^i_k} - y^i_k\|^2$$

$$+ \beta \sum_k \frac{1}{(N_k)^2} (f^k)^T L^p_k f^k + \frac{\lambda}{N^2} f^T L^r f, \quad (6.8)$$

where $L^r$ is the Laplacian matrix of the symmetrized $W^r$.

The minimization of the objective is achieved when

$$\tilde{f} = R^{-1} \sum_k \frac{1}{l_k} (S^k_{lk} S^k) Y^k_l$$

(6.9)

where

$$S^k_{lk} = \begin{pmatrix} I_{l_k \times l_k} & O_{l_k \times u_k} \end{pmatrix},$$

$$S^k = \begin{pmatrix} O_{N_k \times \sum_{k_i=1}^{k-1} N_{k_i}} & I_{N_k \times N_k} & O_{N_k \times \sum_{k_i=k+1}^{M} N_{k_i}} \end{pmatrix}$$

are label and class selection matrices respectively and

$$R = \sum_k \frac{1}{l_k} (S^k_{lk} S^k)^T (S^k_{lk} S^k)$$

$$+ \beta \sum_k \frac{1}{(N_k)^2} S^k T L^p_k S^k + \frac{\lambda}{N^2} L^r.$$

### 6.4 Regression on Reproducing Kernel Hilbert Space (RKHS)

A series of algorithms such as SVM, Ridge regression and LapRLS [7] employ different regularization items and empirical cost measures to the objective and solve the optimization problem in an
appropriately chosen Reproducing Kernel Hilbert Space (RKHS). One merit for the regression on RKHS is the ability to predict out-of-sample labels.

Let $K$ denote a Mercer kernel: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $\mathcal{H}_K$ denote the induced RKHS of functions $\mathcal{X} \to \mathbb{R}$ with norm $\| \cdot \|_K$. The regression with inter-manifold regularization on the RKHS is defined as

$$
\tilde{f} = \arg \min_{f \in \mathcal{H}_K} \sum_k \frac{1}{l_k} \sum_{x_i^k \in \mathcal{X}^l} \| f_{x_i^k}^k - y_i^k \|^2 + \gamma \| f \|^2_K
$$

$$
+ \beta \sum_k \frac{1}{(N^k)^2} (f_k^k)^T L_k^p f_k^k + \frac{\lambda}{N^2} f^T L^v f. \quad (6.10)
$$
Similar to the Tikhonov regularization, we add an RKHS norm penalization item to the TRIM algorithm as a smoothness condition.

For the multi-class data under the same modality, we have the following theorem:

**Theorem-1.** The solution to the minimization problem (6.10) admits an expansion

\[ \tilde{f}(x) = \sum_{i=1}^{N=\sum_k(l^k+u^k)} \alpha_i K(x_i, x) \]  

(6.11)

**Theorem-1** is a special version of the **Generalized Representer Theorem** [42] and the proof is omitted here. It says that the minimizer of (6.10) can be expressed in terms of the linear expansion of \( K(x_i, x) \) on both labeled and unlabeled data over all the sample classes. Thus the minimization over Hilbert space boils down to minimizing the coefficient vector \( \alpha = [\alpha_1, \ldots, \alpha_{l_1}, \ldots, \alpha_{l_1+u_1}, \ldots, \alpha_{l_1}, \ldots, \alpha_{l_M}, \ldots, \alpha_{l_M+u_M}]^T \) over \( \mathbb{R}^N \) and the minimizer is given by:

\[ \tilde{\alpha} = J^{-1} \sum_k \frac{1}{l_k} (S^k_k S^k_k)^T Y^k, \]  

(6.12)

where

\[ J = \sum_k \frac{1}{l_k} (S^k_k S^k_k)^T (S^k_k S^k_k) K + \gamma I \]

\[ + \beta \sum_k \frac{1}{(N_k)^2} S^k_k L^p K + \frac{\lambda}{N^2} L^2 K. \]

and \( K \) is the \( N \times N \) Gram matrix of labeled and unlabeled points over all the sample classes.

For the out-of-sample data, the estimated labels can be obtained using:

\[ y_{\text{new}} = \sum_{i=1}^{N=\sum_k(l^k+u^k)} \tilde{\alpha}_i K(x_i, x_{\text{new}}). \]  

(6.13)
Note here in this framework the class information for the incoming sample is not required in the prediction stage.

6.5 Experiments

We performed experiments on two synthetic datasets and one real-world regression problem of human age estimation. Comparisons are made with the traditional graph Laplacian regularized semisupervised regression [6]. We also evaluate the generalization performance of the multi-class regression on RKHS. In the experiments, the intra-manifold graphs are constructed using 10 nearest neighbors and the inter-manifold graphs are constructed by following the procedure described in Section 3. For all the configurations, the parameter $\beta$ for the intra-manifold graph is empirically set as 0.001 and the $\lambda$ for the inter-manifold regularization is set as 0.1. For the kernelized algorithms, the coefficient for the RKHS norm $\gamma = 0.001$ and the gaussian kernel $K(x, y) = \exp\{-\|x - y\|^2/\delta_o^2\}$ with parameters $\delta_o = 2^{1/2.5}\delta$ is applied, where $\delta$ is the standard deviation of the sample data. We use the Mean Absolute Error (MAE) criterion to measure the regression accuracy and it is defined as an average of the absolute errors between the estimated labels and the ground truth labels.

6.5.1 Synthetic Data: Nonlinear Two Moons

The nonlinear two moons dataset is shown in Figure 6.3. The colors in the figure are associated with the function values and the variation of those sample labels along the manifold is not uniform. The labeled samples are marked by ‘+’ and their distributions are quite different across classes. As we can see, the sample labels for the class lying on the upper part of the figure are not enough to give an accurate guidance for the regres-
sion on the nonlinear label distribution. The traditional graph Laplacian regularized regression algorithm does not make use of the information conveyed by the inter-class similarity and the prediction result is not satisfactory, while in our algorithm the sample labels from different classes can be utilized to guide the regression and thus estimation accuracy is much higher.

Figure 6.4: Regression on Cyclone Dataset: (a) Original Function Values. (b) Traditional Graph Laplacian Regularized Regression (separate regressors for different classes). (c) Three Class TRIM. (d) Three Class TRIM on RKHS.

6.5.2 Synthetic Data: Three-class Cyclones

One merit of our algorithm is that the regression accuracy may be boosted as the class number increases. This is because the cross manifold information that could be utilized grows rapidly as the class number increases. The Cyclone dataset consists
of three classes of samples and the class distribution is demonstrated in Figure 6.5. The label distributions among different classes are quite similar, while the labeled samples scatter differently from class to class. As we may observe from Figure 6.4, without the inter-class regularization, the regression for certain class may fail due to the lack of sufficient labeled samples while our algorithm still gives a satisfying performance.

6.5.3 Human Age Estimation

In this experiment we consider as a regression problem the age estimation from facial images. In real applications, we often cannot obtain enough age labels though we may easily get plenty of facial images. Those unlabeled data can be used to guide our regression in the semisupervised framework. One challenge for this estimation is caused by the gender difference. Although the male and female may share some similar configurations in the age distribution, just as demonstrated in Figure 7.1, mixing the two may complicate the regression problem. On the other hand, the gender information is usually easy to obtain and it is thus desirable to use both the age labels and gender information to guide the regression.

The aging data set used in this experiment is the Yamaha database, which contains 8000 Japanese facial images of 1600 persons with ages ranging from 0 to 93. Each person has 5 images and the Yamaha database is divided into two parts with 4000 images from 800 males and another 4000 images for 800 females. The ages distribute evenly from 0 to 69 with 3500 male images and 3500 female images, and the rest are distributed in the age gap from 71 to 93. We randomly sampled 1000 photos from the male and female subset respectively and thus altogether 2000 images are used in our experiments. Before the regression, the input image data is preprocessed with Principal Component
Analysis and the first 20 dimensions are used for data projection. To fully evaluate the regression performance for both close set samples and those out-of-sample data, we design two experimental configurations for this dataset.

**Configuration 1: Close Set Evaluation.** In this configuration, we do not have the out-of-sample data and the close set performance of *TRIM* is evaluated in comparison with the traditional graph Laplacian regularized regression [4]. The close set contains altogether 2000 samples with 1000 images from males and 1000 images from females respectively. We vary the number of randomly selected labeled samples and examine the performance of different regression algorithms. The comparison results between *TRIM* and the traditional single class Laplacian Regularized regression are shown in Figure 6.6. We can observe that our algorithm generally gives a higher regression accuracy and the performance improvement is remarkable especially when the number of labeled samples is small. As the number of sample labels increases, the difference between the two algorithms becomes smaller. This may be caused by the fact that as the labels are sparse, the label guidance is far less enough and thus the class information and inter-manifold relations will have a greater influence on the regression accuracy while when the labels are abundant enough to guide the regression, the class information as well as those inter-manifold connections becomes less impor-
Configuration 2: Open Set Evaluation. Now we examine the out-of-sample prediction performance for the kernelized \textit{TRIM} compared with kernelized version of the traditional graph Laplacian regularized regression \cite{7}. In this configuration, the sample set is divided into two subsets: one for the regression function training and the other for the evaluation of aging estimation performance on those out-of-sample data. The training set contains randomly selected 800 male and 800 female images and the remaining ones are used for out-of-sample evaluation. In the testing phase we do not input any gender information. As is demonstrated in Figure 6.7, our algorithm achieves a lower MAE in both training and testing sets. Another observation similar as in the case for close set configuration is that the regression accuracy improvement grows as the sample labels become sparser.

6.6 Conclusions

In this paper, we have presented a novel algorithm dedicated for the regression problem on multi-class data. Manifolds constructed from different classes are regularized separately and to utilize the inter-manifold relations, we developed an efficient
cross manifold label propagation method and the labels from different classes can thus be employed to pilot the regression. Moreover, the regression function is further extended by the kernel trick to predict the labels of the out-of-sample data without class information. Both synthesized experiments and real world applications demonstrated the superiority of our proposed framework over the state-of-the-art semisupervised regression algorithms. To the best of our knowledge, this is the first work to discuss the semisupervised regression problem on multi-class data.

Figure 6.7: Open set evaluation for the kernelized regression on the YAMAHA database. (a) Regression on the training set. (b) Regression on out-of-sample data.

End of chapter.
Chapter 7
Correspondence Propagation

Summary

Besides the intrinsic distribution in the sample vector space, real-world samples in applications may share some additional common structures, such as geometric structure, which can also be used to boost the algorithm performance. In this chapter, we develop an image registration framework utilizing the feature geometric structures to perform correspondence propagation and a closed-form solution is derived. Integrating simple automatic procedures, our framework can easily switch between the semisupervised and unsupervised configurations.

7.1 Introduction

Feature matching of two objects is a fundamental problem for computer vision research, and a variety of computer vision tasks heavily rely on the feature matching results, such as object tracking and recognition, image warping and stitching, and 3D reconstruction. The feature matching accuracy may be affected
by various factors including feature descriptors, similarity measurements, and matching approaches.

Substantive works have been devoted to seeking the correspondences between features extracted from images. Grauman et al. [22] considers the image features as unordered elements in sets of different cardinalities and proposes a pyramid matching algorithm for pursuing inexact correspondences. On the other hand, the popular feature detectors, such as SIFT [34], salient region detector [31], as well as scale and affine invariant interest point detector [36], tend to output interest points or regions in a clustered way. Also, it is observed that the edge points and SIFT features extracted from those images with similar structures often share similar spatial distributions. Thus the feature location also conveys important information for feature matching. The works in [8] [54] and [41] presents approaches for utilizing structure information. They formulate the feature matching problem with integer quadratic programming (IQP) or Semidefinite Programming (SDP) techniques, and hence severely suffer from the high computational cost. Leordeanu et al. [33] proposes a spectral analysis method for promoting feature matching accuracy with the geometric structure information and designs an iterative procedure to eliminate the conflicts among the derived correspondences. [18] adds affine constraints to the spectral matching formulation and proposes a normalization procedure to improve the matching accuracy.

One common issue encountered by all above feature matching algorithms is that the top few matches with the largest similarities are often very accurate, but the matching accuracy falls rapidly when the match number increases, especially for data with noise. Another issue arising in real-world applications is that the unsupervised feature matching algorithms often cannot provide sufficiently accurate results for the subsequent applications such as image stitching and object recognition. A natural
question is how to incorporate extra clues for promoting feature matching performance. In this work, we present a solution for feature matching with the ‘reliable correspondence priors’, from either manual labeling or the top few reliable correspondences obtained by conventional feature matching algorithms.

First, the relative geometric relation of the feature pairs within an image is encoded as a spatial graph, and the matching assignments are considered as the vertices of the product graph constructed from two spatial graphs of the images to be matched. Then, based on the these spatial relations, the assignment neighborhoods are defined on the product graph and the point-to-point matchings are then propagated from those reliable correspondences to the remaining ones. Finally, we deduce an efficient closed-form solution for the feature matching problem by ensuring both spatial consistency and feature similarity agreements.

The works in [33], [8] and [41] also try to employ the feature location information for matching, while our work in this paper differs from them in several aspects. First, our algorithm makes full use of the information provided by those reliable correspondences, and thus our framework is easy to incorporate human interactions to guide the correspondence searching. Second, similar to the semi-supervised learning algorithms, we penalize the Graph Laplacian of the derived product graph to impose a spatial smoothness constraint. Finally, we design two configurations to incorporate human interactions: exact pairwise correspondence labeling and obscure correspondence guidance. An illustration of the whole framework for correspondence propagation from reliable correspondence priors is displayed in Figure 7.1.

Here, we would like to highlight some aspects of our proposed Reliable Correspondence Propagation (RCP) algorithm:

1. RCP makes full use of the prior information of the sparse
reliable correspondences, and is naturally applicable to incorporating the interactive manual labeling for further promoting feature matching accuracy in a semi-supervised way.

2. The algorithmic objective provides a unified formulation that employs both the categorical product graph constructed from two spatial graphs for characterizing spatial smoothness and the bipartite similarity graph for representing feature similarity agreements.

3. A closed-form solution is deduced with extremely low computational cost, and hence our algorithm is applicable to large-scale image registration problems.
7.2 Problem Formulation and Solution

7.2.1 Graph Construction

The two sets of features within two images to be matched are denoted as $\Phi_1 = \{\phi_1^1, \phi_2^1, ..., \phi_N^1\}$ and $\Phi_2 = \{\phi_1^2, \phi_2^2, ..., \phi_N^2\}$ with $\phi_i^k = \{f_i^k, x_i^k\}$, where $f_i^k$ is the feature vector and $x_i^k$ is the feature point location in the $k^{th}$ image ($k \in \{1, 2\}$).

Let $G^k = (V^k, E^k)$ be the undirected spatial graph with vertex set $V^k$ and edge set $E^k$ for the $k^{th}$ image. The edges in $E^k$ reflect the geometric neighboring relations among the features, and can be defined in terms of $k$-nearest-neighbor or an $\epsilon$-ball distance criteria in the feature position space. In addition, an adjacency/weight matrix $W^k$ is defined for the graph $G^k$. One way to compute the weight matrix is directly based on the edge information, namely

$$w_{ij}^k = \begin{cases} 
1 & \text{if } x_i^k \text{ and } x_j^k \text{ are connected}, \\
0 & \text{else.}
\end{cases}$$

There are also other ways for computing the similarity matrix, such as the heat kernel [6], i.e., $w_{ij}^k = e^{-\|x_i^k - x_j^k\|^2 / t}$, where $t \in \mathbb{R}$ is a parameter to define the heat kernel.

To encode the pairwise feature similarity between features from two distinct sets, we introduce the similarity graph, denoted as a triplet $G^{s12} = (V^{s1}, V^{s2}, E^{s12})$. The similarity graph $G^{s12}$ is a bipartite graph, and the weights of $G^{s12}$ are computed as the cosine distance of the feature pair measured in feature vector space.
7.2.2 Regularization on categorical Product Graph

The feature matching process can be considered as seeking a binary function over the product set of $\Phi^1$ and $\Phi^2$:

$$M : \Phi^1 \times \Phi^2 \rightarrow \{0, 1\},$$

where $\times$ denotes the set product and the function value 1 means matching and 0 for mismatching. To transduce the matching assignment from the reliable correspondence priors to the other feature pairs, we first give a neighborhood definition for the matching assignments.

**Definition** Suppose $\Phi^1 = \{\phi^1_{i_1}, \phi^1_{i_2}, \ldots, \phi^1_{i_{N_1}}\}$ and $\Phi^2 = \{\phi^2_{i_1}, \phi^2_{i_2}, \ldots, \phi^2_{i_{N_2}}\}$ are the vertices of graph $G^1$ and $G^2$ respectively. Two assignments $m_{i_1i_2} = \{\phi^1_{i_1}, \phi^2_{i_2}\}$ and $m_{j_1j_2} = \{\phi^1_{j_1}, \phi^2_{j_2}\}$ are neighbors iff both pairs $\{\phi^1_{i_1}, \phi^1_{j_1}\}$ and $\{\phi^2_{i_2}, \phi^2_{j_2}\}$ are neighbors in $G^1$ and $G^2$ respectively, namely,

$$m_{i_1i_2} \sim m_{j_1j_2} \text{ iff } \phi^1_{i_1} \sim \phi^1_{j_1} \text{ and } \phi^2_{i_2} \sim \phi^2_{j_2},$$

(7.1)

where $a \sim b$ means $a$ and $b$ are neighbors on the corresponding graph.

Suppose binary weights are utilized. According to the definition (7.1), the assignment graph $G^a$ is the categorical product graph of $G^1$ and $G^2$, i.e., $G^a = G^1 \times G^2$, and the adjacency of the assignments can be expressed as:

$$w^{12}_{m_{i_1i_2}m_{j_1j_2}} = w^1_{\phi^1_{i_1} \phi^1_{j_1}} w^2_{\phi^2_{i_2} \phi^2_{j_2}} .$$

(7.2)

An example of the categorical product graph is demonstrated in Figure 7.2.

Defined on the space of Cartesian product graph, the assignment
\( M \) can be regarded as a binary matrix of \( N_1 \) by \( N_2 \), i.e.,

\[
M = \begin{pmatrix}
m_{11} & m_{12} & \ldots & m_{1N_2} \\
m_{21} & m_{22} & \ldots & m_{2N_2} \\
\vdots & \vdots & \ddots & \vdots \\
m_{N_11} & m_{N_12} & \ldots & m_{N_1N_2}
\end{pmatrix}, \quad m_{ij} \in \{0, 1\}
\]

(7.3)

where the elements \( m_{ij} \) corresponds to the assignment of \( \phi_i^1 \) to \( \phi_j^2 \). To facilitate the solution, we arrange the columns of \( M \) consequently to construct a vector \( \vec{M} \), i.e.,

\[
\vec{M} = \text{vec}(M) = [m_{11}, m_{21}, \ldots, m_{N_11}, m_{12}, m_{22}, \ldots, m_{N_12}, m_{1N_2}, m_{2N_2}, \ldots, m_{N_1N_2}]^T,
\]

(7.4)

where \( \text{vec}(\cdot) \) is the vectorization operator.

Now the assignment is converted into a function on the \( N_1N_2 \) dimensional vector space and thus the adjacency matrix \( W^{12} \) of the assignments is an \( N_1N_2 \) by \( N_1N_2 \) matrix, i.e.,

\[
W^{12} = W^{2T} \otimes W^1,
\]

(7.5)

where \( \otimes \) is the Kronecker product operator and the corresponding graph \( G^{12} \) is the categorical product graph of \( G^1 \) and \( G^2 \). Note that the adjacency matrix of the categorical product graph can also be defined as \( W^{12} = W^1 \otimes W^2 \) if we rearrange the sequence of assignments while here we adopt the first definition.
so that the assignment arrangement is coherent with that of \( \vec{M} \). When the \( W^1 \) and \( W^2 \) are not binary, the adjacency matrix \( W^{12} \) calculated from 7.5 can still capture the relative geometric relations of the assignments.

To introduce a spatial consistency for the assignments, we make the assumption that the neighboring vertices on the categorical product graph share similar assignment values. This is quite natural in the representation of structural feature sets, since in real world applications, the feature points that constitutes certain kind of structures are often extracted together and thus the features are often matched ‘set’ by ‘set’. Emphasizing this assumption can also transduce the ‘reliable correspondences’ derived from manual labeling or automatic approaches to their neighboring assignments and then the assignments are propagated along the categorical product graph until a final balance is drawn.

According to the spectral graph theory [7], penalizing the Graph Laplacian will impose smoothness conditions on the possible solutions. The Graph Laplacian can be expressed as:

\[
\vec{M}^T L^{12} \vec{M}^T = \frac{1}{2} \sum_{ij} w_{ij}^{12} (m_i^v - m_j^v)^2,
\]

where \( m_i^v \) is the \( i^{th} \) element of \( \vec{M} \), \( L^{12} = D^{12} - W^{12} \) is the Laplacian matrix of the categorical product graph and \( D^{12} \) is a diagonal matrix with \( D_{ii}^{12} = \sum_j W_{ij}^{12} \). If \( m_i^v \) and \( m_j^v \) are adjacent in the graph, i.e., the \( w_{ij}^{12} \) is large, the solutions that give large distance on \( m_i^v \) and \( m_j^v \) will draw large penalty and thus minimizing the Graph Laplacian leads to a smooth solution over the graph.
7.2.3 Consistency in Feature Domain and Soft Constraints

Besides the geometric consistency, we also emphasize the coherence in the feature domain. The pairwise feature agreement is encoded by the $N_1$ by $N_2$ adjacency matrix $S$ of the similarity graph. The coherence of the feature similarity is then converted into the maximization of item,

$$|S \odot M|_1 = vec(S) \cdot vec(M) = \vec{S^T} \vec{M},$$

w.r.t. $m_i^v \in \{0, 1\}$, $s_i \geq 0$ \hspace{1cm} (7.6)

where $\odot$ is the matrix Hadamard product, $|\cdot|_1$ is the $L^1$ matrix norm and $\cdot$ is the vector inner product.

Finally, for those one-to-one correspondence configurations, a soft penalty is introduced, i.e.,

$$\sum_{i=1}^{N_1} (|A^i_1 \odot M|_1 - 1)^2 + \sum_{i=1}^{N_2} (|A^i_2 \odot M|_1 - 1)^2,$$ \hspace{1cm} (7.7)

where $A^i_1$ is an $N_1$ by $N_2$ coefficient matrix with 1 in the $i^{th}$ row and 0 for others; $A^i_2$ is an $N_1$ by $N_2$ coefficient matrix with 1 in the $i^{th}$ column and 0 for other elements.

Vectorizing the coefficient matrices $A^i_1$ and $A^i_2$ and arranging the derived column vectors, we construct the constraint coefficient matrices $\hat{A}_1$ and $\hat{A}_2$:

$$\hat{A}_1(:, i) = vec(A^i_1), \quad \hat{A}_2(:, i) = vec(A^i_2).$$

Then the item (7.7) can be expressed as:

$$Tr((\hat{A}^T_1 \vec{M} - e_{N_1})^T (\hat{A}^T_1 \vec{M} - e_{N_1}))$$

$$+ Tr((\hat{A}^T_2 \vec{M} - e_{N_2})^T (\hat{A}^T_2 \vec{M} - e_{N_2})),$$ \hspace{1cm} (7.8)

where $\hat{A}_1 = e_{N_2} \otimes I_{N_1}$ is an $N_1 N_2$ by $N_1$ matrix, $\hat{A}_2 = I_{N_2} \otimes e_{N_1}$ is an $N_1 N_2$ by $N_2$ matrix, $e_N$ is an $N$ dimensional column vector of 1 and $I_N$ is an $N$ by $N$ identity matrix.
Note that for the one-to-one correspondence, we can also impose hard constraints, i.e.,

\[ \hat{A}_1^T \tilde{M} = e_{N_1} \quad \text{or} \quad \hat{A}_2^T \tilde{M} = e_{N_2}, \]  

(7.9)

but these conditions may not be satisfied, since the feature extracted in one image may not have a correspondence in the other image due to the noise, occlusion or the inequality of the feature set cardinality. Thus we adopt a soft penalty in the objective and the affine constraints are consequently removed from the formulation.

### 7.2.4 Inhomogeneous Pair Labeling

Since the one-to-one matching is optimized on the product graph of the two input graphs, the number of variables can be extremely large and it grows rapidly with the increase of the input vertex number. The number of features extracted depends on various factors such as the feature extractors, the complexity of surroundings, the scales searched for maximum and the size of images. For example, in the object recognition database, we may easily extract 100-1000 SIFT features from the input image. Suppose we have two input feature sets of cardinality 300, then the matching variables in the optimization problem can be as many as \( 300 \times 300 = 90000 \), which makes the problem intractable. However, in most vision problems, the assignment variables are highly redundant. Substantive assignment variables are dispensable due to the low similarity, or, with large feature distances between the involved feature pairs. We call these assignments ‘inhomogeneous pairs’. Rather than simply remove them, in our framework the ‘mismatch’ information of those inhomogeneous pairs is also utilized. Specifically, they are assigned 0, which indicates that the corresponding point pairs
will not be matched, i.e.,
\[ M_{i,j} = \vec{M}_i + (j - 1) \times N_1 \leq 0 \text{ if } \{ \phi_i^1, \phi_j^2 \} \in \Psi, \]  
(7.10)
where \( \Psi \) is the set of inhomogeneous pairs. Then the ‘mis-match’ information of those inhomogeneous pairs is also utilized to guide the solution and transduced to the remaining ones.

### 7.2.5 Reliable Correspondence Propagation

In the following the known correspondences including some reliable correspondences and certain number of inhomogeneous pairs are called labeled assignments or labeled point pairs. We arrange the matching variables so that the labeled assignments are placed ahead, i.e.,
\[ \vec{M}^* = [\vec{M}^l; \vec{M}^u], \]  
(7.11)
where \( \vec{M}^l \) represents the assignments of the labeled point pairs, \( \vec{M}^u \) corresponds to the assignment values of the remaining unlabeled point pairs to be estimated. \( \vec{M}^* \) is the rearranged assignment vector.

Correspondingly, the constraint coefficient matrices \( \hat{A}_1, \hat{A}_2 \) and the vectorized adjacency matrix \( \vec{S} \) of the similarity graph are also rearranged, so that,
\[ \hat{A}_1^* = [\hat{A}_1^l; \hat{A}_1^u], \hat{A}_2^* = [\hat{A}_2^l; \hat{A}_2^u] \text{ and } \vec{S}^* = [\vec{S}^l; \vec{S}^u], \]  
(7.12)
where \( \hat{A}_1^l, \hat{A}_2^l \) and \( \vec{S}^l \) are the coefficients and vectorized adjacency sub-matrix of the similarity graph for the labeled assignments respectively; \( \hat{A}_1^u, \hat{A}_2^u \) and \( \vec{S}^u \) are the coefficients and vectorized adjacency sub-matrix for the unlabeled assignments; and \( \hat{A}_1^*, \hat{A}_2^* \) and \( \vec{S}^* \) are the rearranged coefficients and vectorized similarity graph adjacency matrix.

Due to the variable rearrangement, the vertex order in the categorical product graph is also modified. The rearranged adjacency matrix \( W^{12*} \) and the corresponding Laplacian matrix
Algorithm 4 Elicit $k$ correspondences. [Input: $M$]

1: Output the correspondence $m_{ij} = \{\phi_i, \phi_j\} = \arg\max_{\phi_i, \phi_j} M$.
2: Remove from $M$ all potential assignments in conflict with $m_{ij}$.
3: If column or row dimension of $M$ becomes 0 or if the output correspondence number reaches $k$, stop. Otherwise go back to step 1.

$L^{12*}$ are:

$$W^{12*} = \begin{pmatrix} W^{12ll} & W^{12lu} \\ W^{12ul} & W^{12uu} \end{pmatrix}, \quad L^{12*} = \begin{pmatrix} L^{12ll} & L^{12lu} \\ L^{12ul} & L^{12uu} \end{pmatrix}. \quad (7.13)$$

Integrating all factors and we get the final optimization formulation for our feature matching framework:

$$\min_{\tilde{M}^*} -\tilde{S}^T \tilde{M}^* + \lambda \tilde{M}^* L^{12*} \tilde{M}^* +$$

$$\gamma(Tr((\hat{A}_1^* \tilde{M}^* - e_{N_1})^T (\hat{A}_1^* \tilde{M}^* - e_{N_1}))$$

$$+ Tr((\hat{A}_2^* \tilde{M}^* - e_{N_2})^T (\hat{A}_2^* \tilde{M}^* - e_{N_2}))) \quad \text{w.r.t.} \quad m_i^* \in \{0, 1\}, \quad i \in \{1, 2, \ldots, N_1 N_2\} \quad (7.14)$$

where $m_i^*$ is the $i^{th}$ element of $\tilde{M}^*$.

We relax the binary integer optimization problem to real values by discarding the constraints in (7.14) and the formulation is converted to an unconstrained quadratic optimization. Take the derivative w.r.t. $\tilde{M}^*$ and substitute the equation 7.11, we obtain a closed-form relation between the labeled and unlabeled assignments:

$$\tilde{M}^u = C^{-1}_{uu} (B_u - C_{ul} \tilde{M}^l), \quad (7.15)$$

where

$$C = \begin{pmatrix} C^{ll} & C^{lu} \\ C^{ul} & C^{uu} \end{pmatrix}$$

$$= \gamma (\hat{A}_1^* \hat{A}_1^T + \hat{A}_2^* \hat{A}_2^T) + \lambda L^{12*} \quad (7.16)$$
and

\[
B = \begin{pmatrix}
B^l \\
B^u
\end{pmatrix} = \gamma (\hat{A}_1^* e_{N_1} + \hat{A}_2^* e_{N_2}) + \frac{1}{2} \tilde{S}^*
\]  \tag{7.17}

### 7.2.6 Rearrangement and Discretizing

To get the original assignment \( \mathcal{M} \), we first take the inverse process of the element arrangement described above and convert \( \tilde{M}^* \) to \( \tilde{M} \), then reshape the derived assignment vector into the \( N_1 \times N_2 \) matrix \( M \). Since the assignment variables have been relaxed, we tried two discretization strategies: thresholding and eliciting. Setting a threshold for discretization is natural and it can determine the correspondence number automatically. This strategy is also suitable for the cases in which the correspondences are not required to be one-to-one. On the other hand, in case a fixed number of one-to-one correspondences are needed, we design an iterative correspondence eliciting procedure, which is displayed in Algorithm 1. Finally the whole algorithmic process is listed in Algorithm 2.

### 7.3 Algorithmic Analysis

#### 7.3.1 Selection of Reliable Correspondences

The accuracy of those reliable correspondences are critical for final performance. One way to obtain these reliable correspondences in the automatic matching configuration is simply to pick up a few pairs with the highest similarity scores while the correspondences derived in this way may be clustered together and their guidance for the correspondence searching is thus limited. The work [13] proposes an Adaptive Non-Maximal Suppression
### Algorithm 5 Reliable Correspondence Propagation

1. **Graph Construction**: Conuct the spatial graphs $G^1$ and $G^2$ from the feature locations and calculate the adjacency matrix for the categorical product graph using $W^{12} = W^{2T} \otimes W^1$. Construct the bipartite similarity graph $G^{s12}$ according pairwise feature similarity.

2. **Constraint Coefficient Matrix Initialization**: Initialize the constraint coefficient matrices $\hat{A}_1$ and $\hat{A}_2$ according to the cardinality of input feature sets.

3. **Assignment Labeling**: Initialize the corresponding assignments for those reliable pairs as 1 and set the assignment variables as 0 for those inhomogeneous pairs with low similarity values.

4. **Correspondence Propagation**: Rearrange the assignment variables, the adjacency matrices, the constraint coefficient matrices so that the labeled assignments are placed in front of the unlabeled variables and calculate the closed-form solution in (7.15).

5. **Rearrangesment**: Take the inverse process of the arrangement in step 4 and get the correspondences using the strategies described in section 2.6.

(ANMS) strategy to elicit a fixed number of interest points and at the same time keep the the selected interest points spatially well distributed. In this paper, we adopt the Correspondence Elicit Procedure described in Algorithm 1 and the first several correspondences produced are regarded to be reliable in the automatic matching configuration.

The transductive property of our algorithm makes it easy to incorporate human interactions for the correspondence searching and a semi-supervised matching framework is naturally derived. In this work, two configurations of human interactions are used:

**Exact Pairwise Correspondence Labeling**: In this configuration, the users are asked to give exact correspondence labeling for the guidance of matching, and the assignments labeled by human are used as reliable correspondence priors in the feature matching process.

**Obscure Correspondence Guidance**: To facilitate the user labeling, we also provide an obscure matching scheme in
which the user only has to describe a rough correspondence of image parts. The procedures used in the automatic matching configuration are then employed to extract reliable correspondences within the indicated corresponding areas.

### 7.3.2 Computational Complexity

The complexity of the inverse operation for an $n$ by $n$ matrix is $O(n^3)$, which is greater than the spectral algorithms ($O(n^2)$). However, the matrix $C_{uu}$ in our algorithm is sparse and exploiting this sparsity, the computational cost can be greatly reduced. Also, efficient parallel algorithms exist for the gaussian elimination procedure in the computation of the sparse matrix inversion problem and thus the computation time can be further shortened. Another factor affecting the computation cost is the candidate matching variable number, which determines the dimension of the matrix $C_{uu}$. In our experiments, 6000 assignments with the largest similarity scores are fetched as matching candidates and the variable number can be adjusted according to the requirement of the applications. Our algorithm is orders of magnitude faster than the QP and SDP based algorithms and is applicable for the large scale real-world applications.

### 7.4 Applications and Experiments

In this section, our algorithm is systematically evaluated in two settings: semi-supervised and unsupervised. In the semi-supervised setting, the reliable correspondence priors are labeled manually; while in the unsupervised setting, those reliable correspondences are derived automatically. In all the experiments, the SIFT [34] descriptor is used for salient point detection and representation; the spatial graph is constructed using 10-nearest neighbors and the weights for the spatial graphs are calculated.
using heat kernels $K(x, y) = \exp\{-\|x - y\|^2 / \delta_o^2\}$ with parameters $\delta_o = 2^{1/2.5}\delta$ applied, where $\delta$ is the standard deviation of the feature locations. For the similarity graph, 16 nearest neighbors are used and the cosine distance is directly used as the graph weight. The coefficient $\lambda$ is empirically set as 0.4 and $\gamma$ is set as 0.05. In the inhomogeneous pair labeling process, we keep 6000 pairs with the top similarities as candidate matchings and others are labeled 0. The performance of our algorithm is systematically compared with the state-of-the-art feature matching algorithms.

7.4.1 Matching Demonstration on Object Recognition Databases

In this subsection, we evaluate our algorithm on the Caltech 101 Object Recognition database \(^1\) and ETH-80 database \(^2\). Four categories of images are used in this demonstration, i.e., the 'motorbikes' and 'car side' images from Caltech 101 database as well as the 'dog' and 'horse' images from the ETH-80 database. Since for the objects of different types, the correspondences may not be one-to-one, a threshold of 0.01 is used in the discretization process and thus the correspondence number is determined automatically. For comparison, the matchings with the largest $k$ cosine distances are also plotted as baseline, where $k$ is the number of correspondences determined by our algorithm. The matching results are demonstrated by Figure 7.4, in which the reliable correspondences drawn by hand are marked by red stars, the obscure guidance indicated by human interaction is described by rectangles of different colors and the automatically derived reliable correspondences are plotted by small crosses. From the results we can observe that the matching accuracy is boosted with the guidance of the manually labeled correspondences, and

\(^1\)http://www.vision.caltech.edu/Image Datasets/Caltech101/
\(^2\)http://www.vision.ethz.ch/projects/categorization/
Figure 7.3: The number of correct matches v.s. the number of automatically selected reliable correspondences on the first two images of Graffiti database.

the unsupervised version of our algorithm is also superior over the baseline algorithm.

7.4.2 Automatic Feature Matching on Oxford Image Transformation Database

In this subsection, the unsupervised version of our algorithm is evaluated on the Oxford real image transformation database \(^3\). The Oxford database is a benchmark database for the feature descriptor evaluation. It contains eight subsets for six different geometric and photometric real image transformations, including zoom, rotation, viewpoint change, image blur, JPEG compression, and light variation. Two different scene types are involved for the case of rotation, viewpoint change, and blur: one contains homogeneous regions with distinctive edge boundaries and the other contains repeated textures of different forms, which facilitates us to analyze the effect of changing the image conditions and the scene type separately. The image width and height are resized to 1/5 of the original ones and for each image, 100-500 SIFT descriptors are extracted. Since the homographies between the reference image and other images in each particular subset are given, we can derive the ground truth matches for the

---

\(^3\)http://www.robots.ox.ac.uk/~vgg/research/affine.
40-180 assignments are extracted as the reliable correspondences using Algorithm 1 in the evaluation. The matching score is calculated as the ratio between the number of correct matches and the smaller value of detected feature numbers from the image pair. In this section, our algorithm is systematically compared with the state-of-the-art matching algorithms, such as the spectral correspondence technique (SC) [33] and the matching algorithm used in [34] (SM), which compares the distance of the closest neighbor to that of the second-closest neighbor. We take the $N_1$ by $N_2$ pairwise similarity matrix as the input $M$ for the Correspondence Eliciting Procedure (CE) and the matching scores are also reported. The QP and SDP based algorithms are inapplicable for comparison due to the large number of points involved. For the adjacency matrix $M$ in the spectral correspondence technique [33], we assign a score that is linearly increasing with the cosine distance between the feature and its candidate corresponding feature to the diagonal element. Since the adjacency matrix of the categorical product graph in our algorithm represents the geometrical relations of assignments, the non-diagonal elements of $M$ is set using the corresponding elements in $W^{12}$. The detailed results are demonstrated in Figure 7.5. It is observed that our algorithm generally reaches a higher accuracy compared with the state-of-the-art techniques and the algorithmic performance is stable over all the subsets. Although in some situation such as the JPEG compression the spectral technique shows an excellent performance, it is not so stable in most cases.

Though the spectral based technique also employs geometric information as well as feature similarity in the matching process, our algorithm generally produces a better performance. The main reason is that our algorithm essentially puts different weights on the correspondences and the reliable correspondences
are emphasized, while this information is ignored in other state-of-the-art matching algorithms.

### 7.4.3 Influence of Reliable Correspondence Number

In the unsupervised configuration, the performance of our algorithm relies on the accuracy of the reliable correspondences, which also deteriorates as the correspondence number increases. It is interesting and necessary to evaluate the performance of our algorithm with respect to the number of automatically selected ‘reliable’ correspondences. Figure 7.3 shows the correct matching number versus the number of reliable correspondences automatically derived. We can observe that the correct match number increases along with the increase of the reliable correspondence number within a reasonable range, and then the accuracy falls when the reliable correspondence number becomes too large to give an accurate guidance.

### 7.5 Conclusion and Future Works

In this paper, we proposed an efficient feature matching framework that transduces certain number of reliable correspondences to the remaining ones by utilizing both geometric smoothness constraints and feature agreements. Furthermore, the framework is naturally extended to incorporate human interactions for promoting feature matching performance. Experimental results showed that our algorithm, both the semi-supervised and the unsupervised versions, achieves a higher matching accuracy compared to the state-of-the-art techniques. We are planning to further investigate our algorithm with other feature descriptors and explore the combination with the ANMS strategy for reliable correspondence selection.
End of chapter.
Figure 7.4: Feature matching results on Caltech 101 and ETH-80 example images. The first row of each figure is the performance of (a) semi-supervised matching by manual pairwise correspondence labeling, (b) semi-supervised matching of obscure correspondence guidance, and (c) automatic version of our algorithm; and the second row demonstrates the baseline results. The correspondence number of the two figures within the same column is the same.
Figure 7.5: Automatic feature matching score on the Oxford real image transformation dataset. The transformations include viewpoint change ((a) Graffiti and (b) Wall sequence), image blur ((c) bikes and (d) trees sequence), zoom and rotation ((e) bark and (f) boat sequence), illumination variation ((g) leuven) and JPEG compression ((h) UBC).
Chapter 8

Conclusion and Future Work

In this thesis, we proposed a series of algorithms to explore the intrinsic sample structures for a better performance. The samples are expressed as either tensors or vectors, and pairwise sample relations are generally represented by graphs. Thus the intrinsic manifold distribution is captured by the weighted graph adjacency matrix. The analysis is then carried out based on the constructed graphs.

Aiming at enlarging the separability among classes, discriminant analysis tries to maximize the between class distance and at the same time minimize the within class scatter. Recent graph embedding framework constructs intrinsic graphs and penalty graphs for within class and between class samples respectively. Consequently graphs are also incorporated into the trace quotient formulation. Moreover, the state-of-the-art manifold embedding algorithms, including LLE, ISOMAP, Laplacian Eigenmap and LPP, can also be formulated as trace quotient optimization problem. Since no closed form solution exists for the optimization of the trace ratio objective, traditional algorithms turns to solve a ratio trace problem instead. In this thesis, efficient iterative procedures are proposed to directly optimize the trace ratio objective and the algorithmic convergency is also proved. We also proposed an unsupervised embedding algorithm that employs the trace ratio optimization process. The distances
of faraway sample pairs are maximized and meanwhile the local manifold structures are preserved, which essentially results in a Trace Ratio optimization problem.

In real-world applications, sample labels are generally rare and the unlabeled samples are comparably easier to obtain. Semisupervised framework tries to explore the information provided by both labeled and unlabeled samples to improve the system performance. The intrinsic manifold structures are also exploited much by the semisupervised learning algorithms. We proposed a novel regression algorithm on multi-class data manifolds. Samples of different classes are considered as lying on different sub-manifolds. Intra-manifold regularization and inter-manifold regularization are then introduced to boost the performance.

Finally, we built a general framework for feature correspondence searching that utilizes both geometrical structures and pairwise feature agreements for a robust registration. The correspondence is propagated from sparse reliable correspondences to the remaining. Combining automatic approaches for the reliable correspondence extraction, our framework can also work in an unsupervised manner.

Although great progress has been made during the past decades in the pursuit of intrinsic sample structures, it is still far from enough. Some possible future works we would like to point out include:

1. Regularization and generalization for the trace ratio based discriminant analysis. Discriminant analysis algorithms may get overfit when the training data set is too small. Consequently, in real applications, one has to employ large data set in the training process to reassure the overfit problem. [23] introduces a regularized discriminant analysis algorithm to fix the singularity problem met in the small training set. Still, how to regularize the classifier in the
training process remains an open problem in computer vision.

2. Automatic parameter estimation. The performance of subspace analysis algorithms is influenced by parameters, such as the preprocessing PCA dimension and LDA dimension; and the optimal parameter may be different for different data set. Thus it is desirable to perform parameter estimation to further improve the algorithm robustness.

3. From point-to-point correspondence to set-to-set correspondence.

4. Multi-scale correspondence searching.

5. Combine the object segmentation and correspondence searching process.

End of chapter.
Bibliography


[38] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Ed)*.


