

密级: 绝密



中国科学院大学
University of Chinese Academy of Sciences

硕士学位论文

面向图像集合分类的黎曼流形判别学习方法研究

作者姓名: 李显求

指导教师: 陈熙霖 研究员

中国科学院计算技术研究所

学位类别: 工学硕士

学科专业: 计算机应用技术

研究 所: 中国科学院计算技术研究所

二〇一六年四月

Discriminant Learning **Method** on Riemannian Manifold
for Image Set Classification

A Thesis Submitted to
The University of Chinese Academy of Sciences
in Partial Fulfillment of the Requirement
for the Degree of
Master of Science
in
Computer Science and Technology

by
Li Xianqiu
Thesis Supervisor: Professor Chen Xilin

Institute of Computing Technology
Chinese Academy of Sciences
April, 2016

声 明

我声明本论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，本论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

作者签名：

日期：

论文版权使用授权书

本人授权中国科学院计算技术研究所可以保留并向国家有关部门或机构送交本论文的复印件和电子文档，允许本论文被查阅和借阅，可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编本论文。

(保密论文在解密后适用本授权书)

作者签名：

导师签名：

日期：

摘要

视觉作为人类的主要的感知机能之一，对人类感知世界的重要性不言而喻。计算机视觉的任务就是为计算机赋予接近甚至超过人类视觉的感知能力。图像作为计算机视觉任务的主要输入，与其它数据形式（如文本，语音等）相比蕴含了更多的信息。

另一方面尽管图像本身蕴含了丰富的信息但是如何运用这些信息，以及图像本身的一些问题（如视角变化大、光照变化剧烈、分辨率低等）也给视觉任务带来不小的挑战。与此同时，越来越多现实生活中的数据以集合的形式出现：视频监控数据、用户上传视频、主题相册、物体的多视角数据以及动作描述视频等在近年来都呈现出爆发式的增长；图像集合分类问题也在这样的背景下应运而生，针对集合数据呈现出的量大但质未必优的特点，图像集合分类问题的核心任务之一便是利用数据量大的特点以克服质低的问题。经过 10 多年的发展，根据图像集合的表示方式的不同，图像集合分类相关方法逐渐形成了以下的一些类别：1、子空间以及流形建模的方法；2、仿射包建模的方法；3、统计建模的方法；4、深度学习的方法；5、其它（稀疏编码，协同表示等）。

在众多方法中，统计建模的方法以其优越表现逐渐成为图像集合分类问题的主要方法之一，本文将以黎曼流形为工具对统计建模图像集合问题进行研究分析。本文的主要工作包含：(1) 研究了矩阵函数与流形上的优化理论与方法，在对流形、矩阵函数等概念进行介绍的基础上，**针对矩阵流形上，针对矩阵流形上的优化问题进行讨论与探究，并结合学位论文课题中提炼出的相关实例对矩阵流形优化进行介绍，一方面帮助读者理解并复现本文所提出的方法和结论，另一方面为解决类似流形优化问题提供借鉴。**(2) 提出了黎曼流形上的偏最小二乘回归方法，通过在流形上单一切空间构建子流形的方式将欧氏空间中的偏最小二乘回归 (Partial Least Square Regression, PLS) 扩展到黎曼流形空间；考虑到黎曼流形与欧氏空间的几何结构差别以及图像集合数据稀疏的问题，进一步设计了流形上多切空间构建子流形的方法，采用逐步回归的方案整合多个切空间中的结果；本文以非奇异协方差矩阵即对称正定矩阵 (Symmetric Positive Definite, SPD) 黎曼流形为实例，在集合数据分类问题上进行了实验验证，取得了与当前最优方法可比甚至更好的结果。(3) 提出了低秩对称半正定矩阵 (Low-Rank symmetric Positive Semi-Definite, PSD) 建模图像集合的方法，解决采用样本协方差矩阵建模图像集合时由于数据稀疏带来的矩阵奇异（不满秩）、由于噪声带来的矩阵估计不准、以及对称正定矩阵表示存储高计算量大等问题，**并采用 Graph Embedding 的方法将判别信息内嵌到的低秩对称半正定矩阵表示中，最后在 KDA[7] 的框架下研究了该表示下的判别学习方法问题，最终的实验验证了低秩对称半正定矩阵表示的有效性。**

关键词：图像集合；统计建模；黎曼流形；判别学习

Discriminant Learning Method on Riemannian Manifold for Image Set Classification

Li Xianqiu (Computer Science and Technology)

Directed by Chen Xilin

Vision works as one of the main abilities for human to perception the real world, the importance goes without saying. The mission of CV (Computer Vision) is to endow computers with close to or even greater ability than human to perception the real world.

As the main input, images contains much more information than text, audio and so on which is good for CV task, while how to make full use of the information becomes a problem. The variations of images bring great challenges to CV tasks. At the same time, data occurs more frequently in the form of image set, such as surveillance video, multi-view image sets and so on. Under these background, image set classification comes into being. Image sets usually contain large amount of images in poor quality. So one core task of image set classification is overcome the quality disadvantage with advantage of quantity.

With more than ten years of development, a lot of methods have been propose for this task. According to how to model an image set they can be divided into following categories:
1. Subspace/Manifold base methods, 2. Affine hull base methods, 3. Statistics model based methods, 4. Deep Learning based methods. 5. Others, like Dictionary/Sparse coding based method, Collaborative representation method, etc.

Among the categories listed above, Statistics model based methods attract a lot attention with its excellent performance. This article takes Riemannian manifold as tool and try to explore statistics model based methods. Following contents are included: 1) Studied matrix function and manifold optimization theory and methods. Based on the concept introduction to manifold and matrix function, jointing with the real problems extract form the research topics, optimization on the manifold have been studied in this part. On the one hand it will help readers understand and implement methods proposed in this article, on the other hand it can also provide reference for similar problems' solving. 2) Proposed Partial Least Square Regression methods on Riemannian manifold with submanifold construct from one tangent space (usually is the tangent space of samples' Karcher mean). Then in order to overcome the structure difference between Euclidean space and Riemannian manifold as well as the sparse drawback of samples multi-tangent space Partial Least Square Regression method was designed. On the Symmetric

Positive Definite (SPD) matrices manifold, image set classification experiment were designed to test the proposed method and find out that this method reached or even outperforms the state-of-art performance on the commonly used databases. 3) Proposed Low-Rank PSD matrices model image set's method to overcome the rank-deficient and high dimension problems of sample covariance models as well as lack of scale information (eigenvalue) drawback in the subspace models. With Graph Embedding framework we encoded label information into Low-Rank PSD representations of image sets then designed the discriminant learning methods with Kernel Discriminant Analysis framework. The final experiments on the commonly used databases support our proposition.

Keywords: Image set; Statistics model; Riemannian manifold; Discriminant learning

目 录

摘要	I
目录	V
图目录	IX
表目录	XI
第一章 绪 论	1
1.1 符号说明	1
1.2 问题的背景与意义	2
1.3 国内外研究现状	2
1.3.1 子空间以及流形建模的方法	3
1.3.2 仿射包建模的方法	6
1.3.3 统计建模图像集合的方法	7
1.3.4 深度学习的方法	8
1.3.5 国内外研究现状小结	10
1.4 数据介绍	11
1.5 本文的组织结构	13
第二章 矩阵函数的导数计算与矩阵流形上的基本优化方法	15
2.1 黎曼流形简介	15
2.1.1 黎曼流形	15
2.1.2 对称正定矩阵（SPD）流形	17
2.2 优化问题与梯度	18
2.2.1 Lagrange 对偶问题	18
2.2.2 梯度计算问题	19
2.2.3 梯度下降和共轭梯度	20
2.3 矩阵函数的导数计算	22
2.3.1 矩阵函数求导的一般形式	22
2.3.2 矩阵包含 0 特征值的问题	24

2.3.3 矩阵函数的偏导数计算示例	25
2.4 矩阵流形上的基本优化问题	28
2.5 总结	31
第三章 黎曼流形上的 PLS 回归	33
3.1 偏最小二乘方法	33
3.2 黎曼流形上的投影问题	35
3.2.1 一般化的投影	36
3.2.2 SPD 矩阵流形上的均值	36
3.2.3 黎曼流形上的子流形空间投影	37
3.3 黎曼流形上的 PLS 回归问题	38
3.3.1 黎曼流形上 PLS 回归问题的一般形式	39
3.3.2 面向图像集合分类的黎曼流形上的 PLS 回归	40
3.4 实验验证	45
3.4.1 原始特征构造	46
3.4.2 SPD 矩阵表示的构造	46
3.4.3 实验结果与分析	46
3.5 总结与下一步工作	48
第四章 Low-rank PSD 矩阵判别学习方法	49
4.1 Stiefel 流形和 Grassmann 流形	50
4.2 Fixed-Rank PSD 流形	53
4.3 Fixed-Rank PSD 流形研究概况	54
4.3.1 Fixed-Rank PSD 表示图像集合	55
4.3.2 Fixed-Rank PSD 流形用于图像集合分类	55
4.4 Low-Rank PSD 流形判别学习方法	56
4.4.1 融入判别信息的 Low-Rank PSD 矩阵的构造	58
4.4.2 Low-Rank PSD 矩阵集合上的判别学习方法	61
4.5 实验结果与分析	61
4.6 总结与下一步工作	63

第五章 结束语	65
5.1 本文工作总结	65
5.2 反思与讨论	66
参考文献	69
致 谢	i
作者简介	iii

图目录

图 1.1 几个图像集合的例子	3
图 1.2 子空间之间的主夹角示意图	4
图 1.3 流形的局部线性近似示意图	5
图 1.4 MDA 方法示意图	5
图 1.5 仿射包建模图像集合方法关系图	7
图 1.6 2×2 对称正定矩阵的外边界在 3 维空间中的结构	8
图 1.7 多统计模型建模图像集合方法	8
图 1.8 统计建模图像集合的方法间的关系	9
图 1.9 深度学习建模图像集合的代表性网络结构	9
图 1.10 数据库示例	11

表目录

表 1.1 符号说明	1
表 1.2 数据库列表.....	11
表 3.1 黎曼流形上的 PLS 回归算法实验结果	47
表 4.1 Fixed-Rank PSD 流形中的核	56
表 4.2 SPD 矩阵流形上的距离度量	57
表 4.3 Fixed Rank PSD 流形判别学习算法实验结果	62
表 4.4 Power Metric 相关的核.....	63

第一章 绪 论

宋代诗人苏东坡说过：“博观而约取，厚积而薄发”。意思是说，只有广见博识才能择其精者而取之。研究如此，研究生生涯亦是如此。研究生生涯作为人生的一部分，从长远来看是一个厚积的过程，这个时期积累的对待问题的态度，见识的众多同行的思想碰撞以及研究过程中的失败与成功等，都会成为今后生活的财富；而短期内，也就是落实到研究中，只有充分调研了问题的背景，了解了国内外了情况并取其精华去其糟粕，准备了充分的数据才能在自己的实际工作中得心应手，做出期望的成果。

1.1 符号说明

在进入本文的主要内容之前，由于本文涉及较多数学符号，并为了节约篇幅，这里利用表1.1统一对本文中的主要符号进行说明。并且本文约定：如无特别说明将使用小写字母（如： a, b ）表示常量，小写加粗（如： \mathbf{x} ）表示向量，大写的字母（如： X ）表示矩阵，子空间或集合（具体可根据上下文确定），大写字母加粗（如： \mathbf{X} ）表示张量。

表 1.1 符号说明

符号	说明
\mathbb{R}^n	n 维向量空间，特别地 \mathbb{R} 表示实数空间
M	此符号专用于表示流形 (Manifold)
(S, g)	表示黎曼流形 (集合 S 以及其上的黎曼度量 g 的组合)，通常为了简单起见也用 S 代替该流形 (如：有时也用 \mathbb{S}_d^+ 对称正定矩阵流形)，因此 S 的具体意义需要根据上下文确定
\mathbb{S}_d^+	$d \times d$ 的对称正定矩阵集合 (SPD 矩阵)
$\mathbb{S}_d^+(k)$	秩为 k 的 $d \times d$ 半正定矩阵集合 (Fixed-Rank PSD 矩阵)
\mathbb{S}_d	$d \times d$ 的对称矩阵构成的集合
$\text{St}(n, k)$	$n \times k$ 列满秩矩阵的集合，也表示 non-compact Stiefel 流形
$\text{St}^*(n, k)$	$n \times k$ 列正交矩阵的集合，也表示 compact Stiefel 流形
$\text{Gr}(n, k)$	\mathbb{R}^n 空间中 k 维子空间构成的集合，也表示 Grassmann 流形
$\log(\cdot)$	不做特别说明的话本文中表示的是矩阵的 log 函数
$\exp(\cdot)$	不做特别说明的话本文中表示的是矩阵的 exp 函数
Log	流形上的 Log 变换
Exp	流形上的 Exp 变换
R	流形上的 Retraction 变换
T	流形上的 Vector Transport 变换
$T_x M$	流形 M 上 x 处的切空间 (tangent space)。特别地， M 上的所有切空间记为 TM 称为 M 上的切空间束

1.2 问题的背景与意义

计算机视觉的任务就是希望给机器赋予等同于人类甚至是超过人类视觉系统对于周围环境的处理能力。图片作为计算机视觉的主要输入，为计算机理解提供丰富信息的同时也给计算机视觉任务带来了挑战：首先，图片是三维空间向二维空间的投影，大量的信息在这个过程中丢失；其次，由于拍摄的角度变化，光照变化以及低分辨率，遮挡等问题导致了使用单一的图片进行识别、理解等任务变得十分困难；另一方面，由于近年来监控视频，主题相册，用户上传视频，**multi-view** 的数据等都以图像集合的形式呈现并呈现出爆发式的增长。**图像集合的分类问题在这样的大背景下应运而生。**

图像集合分类问题中的数据的主要呈现出两个特点：一是图像的量大，二是图像的质却未必有（大 variation）。因而图像集合分类问题的主要任务就是利用量大的特点克服 variation 大的问题。**由于以上的原因，加之数据本身的特点（以集合的形式呈现）都为图像集合分类问题的研究赋予了重要的实践和理论意义。**

计算机视觉的任务的大多都是来源于实际问题的，图像集合的分类也不例外，**前面提到的视屏监控中就是一个很好也非常有用的例子，视频监控中的分类识别问题对于警方的网络追逃，海关的出入境管理等的重要性不言而喻；此外，动作识别（动作的描述往往是一段视频输入）对于暴力事件的甄别，预防犯罪也有重要的意义；另一方面，在众多的用户上传中的视频数据中不管是做基本的视频检索还是做更深层次的用户行为的分析理解等，图像集合分类问题的研究同样具有重要的意义。**

在理论上图像集合分类问题的意义主要体现在：首先，**由于图像数据是以集合的形式出现，而在机器学习领域的研究中相较于单点（向量）作为输入形式的研究，集合作为输入的研究却不是那么充分；所以图像集合的分类问题的研究对于机器学习中的集合对象的研究有着一定的推动意义；其次，由于数据的独特性，其数学表示（往往是子空间，对称正定矩阵，分布函数，流形等等）也比较特殊，而这些非线性结构的表示的研究的另一个意义也将促进机器学习中非线性数据表示的研究。**

1.3 国内外研究现状

在介绍图像集合分类问题的国内外研究现状前，了解图像集合是什么将会对后面的理解决很有裨益。图像集合，顾名思义指的就是多张图片构成的集合，图像集合已经被用于多个领域（视频人脸识别，物体识别，动作识别，表情识别等等），图1.1给出了几个图像集合的例子。

图像集合的分类问题的研究和发展已经走过了 10 多年的时间；在这 10 多年中，图像集合分类问题从最初被引入 CV 领域，逐渐成为计算机视觉中的一个研究热点；这个过程中，学者们不断推陈出新，发展出了一系列的方法和路线，为图像集合分类问题的研究做出了重要的探索。



(a) EXAMP 01: 一段录像 (图片来自 YTC[36] 数据库)



(b) EXAMP 02: 一个物体的 Multi-view (图片来自 ETH80[37] 数据库)



(c) EXAMP 03: 一个动作描述 (图片来自 CMU MoBo[16] 数据库)

图 1.1 几个图像集合的例子

首先，从问题层面可以将图像集合分类问题分为两个大类：图像集合对图像集合的分类问题（Probe 和 Gallery 都是图像集合），图像集合对静态图像的分类问题（Probe 和 Gallery 中一边是静态图像另一边是图像集合）。其中前者是目前图像集合分类问题研究的主流方向，而后者则是一个新的方向，拥有着广泛的运用前景，在该方向上目前的一些主要工作有：文献 [28] (Huang and *et al.* CVPR' 14) 探究了静态图像到图像集合的分类问题；文献 [38] (Li and *et al.* CVPR' 15) 则把静态图像与图像集合的匹配的问题开创性的运用到了视频/图像检索领域；而文献 [67] (Zhu *et al.* ICCV' 13) 借助 Affine Hull 的图像集合的表示在 Metric Learning 的框架下，比较全面的讨论了 point-to-set 以及 set-to-set 的问题。

图像集合到图像集合的分类问题一直以来是图像集合分类问题的主流的方向，这个问题上按照方法这里进一步的可把图像集合分类问题归纳为如下的几类：1、子空间以及流形建模的方法 [18,56,58,63]；2、仿射包建模的方法 [11,26,60,65]；3、统计建模的方法 [20,22,29,42,54,57,59]；4、深度学习的方法 [24,41]；5、其它（稀疏编码 [68]，协同表示 [12] 等）。接下来的内容将简要对它们进行介绍。

1.3.1 子空间以及流形建模的方法

这一类方法出现在图像集合问题研究的早期，为图像集合问题的形成奠定了基础，并且为该问题给出了早期的解决方案。

1.3.1.1 子空间建模的方法

工作 [63], [18] 是使用子空间建模图像集合的代表，工作 [63] 提出了使用图像集合来克服图像大 variation 的问题（以量取胜），并使用子空间建模图像集合，然后使用主夹角来进行距离度量，工作 [18] 进一步的研究了子空间的方法，并且将其统一到 Grassmann

流形下进行解释。子空间建模图像集合的算法流程可以大致概括如下（参考 [18]）：

- 设 $\{\mathbf{x}_{ij} \in \mathbb{R}^l\}_{j=1}^{n_i}$ 表示第 i 个图像集合，其中 n_i 表示的是集合中的样本数
- 计算样本均值： $\bar{\mathbf{x}}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{x}_{ij}$ ，样本协方差： $C_i = \frac{1}{n_i-1} \sum_{j=1}^{n_i} (\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)^T$
- 对样本协方差做 svd 分解获得： $C_i = U_i \Lambda_i U_i^T$ ，指定子空间维数 $m(m < l)$ ，这里假设 svd 分解的结果是按特征值由大到小排序的
- 获得集合的子空间表示： $Y_i = U_i(:, 1:m)$ ，其中 $U_i(:, 1:m)$ 表示取 U_i 的前 m 列
- 定义两个子空间之间的距离，用于度量 $\{Y_j\}_{j=1}^n$ 两两之间的距离
- 在子空间的度量中，主夹角是最主要的概念：

$$\begin{aligned} \cos \theta_k &= \max_{\mathbf{u}_k \in \text{span}(Y_i)} \max_{\mathbf{v}_k \in \text{span}(Y_j)} \mathbf{u}_k^T \mathbf{v}_k \\ \text{s.t } \mathbf{u}_k^T \mathbf{u}_k &= 1, \mathbf{v}_k^T \mathbf{v}_k = 1 \\ \mathbf{u}_k^T \mathbf{u}_i &= 0, \mathbf{v}_k^T \mathbf{v}_i = 0, (i = 1, 2, \dots, k-1) \end{aligned} \quad (1-1)$$

其物理意义图 1.2 所示：

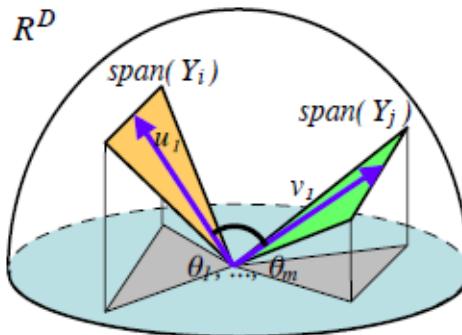


图 1.2 子空间之间的主夹角示意图（图片来自文献 [18]）

- 有了主夹角的定义即可定义子空间之间的度量：

$$\text{Projection metric: } d_p(Y_i, Y_j) = \left(\sum_{i=1}^m \sin^2 \theta_i \right)^{\frac{1}{2}}$$

$$\text{Max correlation: } d_{Max}(Y_i, Y_j) = (1 - \cos^2 \theta_1)^{\frac{1}{2}}$$

$$\text{Min correlation: } d_{Min}(Y_i, Y_j) = (1 - \cos^2 \theta_m)^{\frac{1}{2}}$$

$$\text{Procrustes metric: } d_{CF}(Y_i, Y_j) = 2 \left(\sum_{i=1}^m \sin^2(\theta_i/2) \right)^{\frac{1}{2}}$$

工作 [18] 进一步在此基础上利用 Kernel LDA 的框架进行了判别学习，然后在核空间进行图像集合的分类。

1.3.1.2 流形建模的方法

流形建模的方法 [58], [56] 假设图像集合中的图像位于流形上（并不充满整个空间），使用多个局部线性空间建模图像集合来估计流形结构，然后定义距离进行图像集合分类，如图 1.3 所示。其中 M 表示的是原始的流形结构，文献 [58] 根据数据构建局部线性子空间 $S_1, S_2, S_3, S_4, \dots$ 来近似表示该流形 M ，然后度量通过点到点的距离定义点

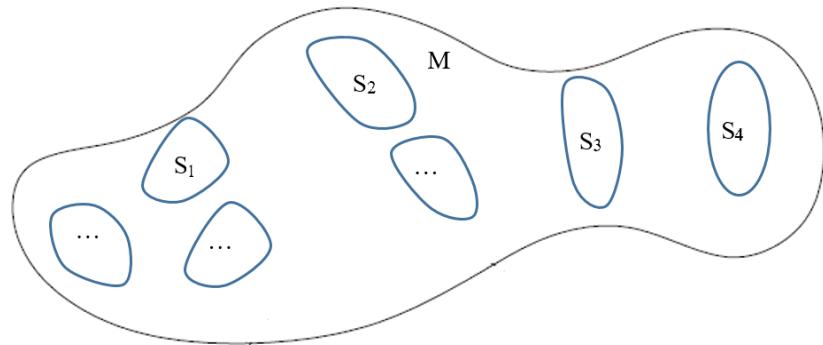


图 1.3 流形的局部线性近似示意图

到子空间的距离然后定义子空间到子空间的距离，最后定义流形到流形的距离，并用此距离来度量两个流形的距离，从而进行图像集合的分类（下述定义中的 S_i, C_j 表示的是子空间而不是矩阵）。

- Point to point distance: $d_{ppd}(x, y) = \|x - y\|$
- Point to subspace distance: $d_{psd}(x, S) = \min_{x' \in S} \|x - x'\|$
- Subspace to subspace distance: $d_{ssd}(S_1, S_2) = \text{any valid space metric}$
- Point to manifold distance: $d_{pmd}(x, M) = \min_{C_i \in M} d_{psd}(x, C_i)$
- Subspace to manifold distance: $d_{smd}(S, M) = \min_{C_i \in M} d_{ssd}(S, C_i)$
- Manifold to manifold distance: $d_{mmd}(M_1, M_2) = \min_{C_i \in M_1} d_{smd}(C_i, M_2)$

文章 [58] 利用上面的 Manifold to manifold distance 来度量流形之间的距离，然后在此距离上进行图像集合的分类问题，另一篇相关的工作 [56] 则是在 [58] 的基础上增加了判别信息得到了 MDA(Manifold Discriminat Analysis) 方法：在构建子空间 $S_1, S_2, S_3, S_4, \dots$ 的时候要求类内散度尽量小而类间散度尽量大，如图 1.4 所示。

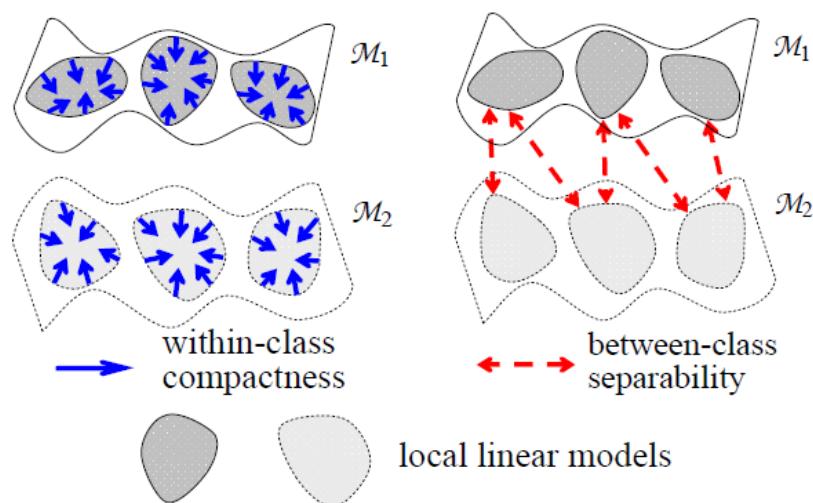


图 1.4 MDA 方法示意图（图片来自 [56]）

1.3.2 仿射包建模的方法

仿射包建模的方法以其简单有效的特点在图像集合分类问题中被研究者所关注，仿射包建模图像集合的方法的代表作有 [11,26,60,65]，而仿射包建模的图像集合的方法最核心的内容就是仿射包（Affine Hull）（由文献 [11] 引入到图像集合分类问题中），其定义如公式1-2所示。

$$H_k = \left\{ \mathbf{x} | \mathbf{x} = \sum_{i=1}^n \alpha_{ki} \mathbf{x}_{ki}, \sum_{i=1}^n \alpha_{ki} = 1 \right\} \quad (1-2)$$

其中 k 是图像集合的下标，此外借助空间中的基向量（这里用 U_k 表示基矩阵）的概念还可以定义如下的形式：

$$H_k = \left\{ \mathbf{x} | \mathbf{x} = \boldsymbol{\mu}_k + U_k \mathbf{v}_k, \mathbf{v}_k \in \mathbb{R}^l \right\} \quad (1-3)$$

两个仿射包之间的距离定义如公式1-4所示。

$$D(H_1, H_2) = \max_{\mathbf{x} \in H_1} \max_{\mathbf{y} \in H_2} \|\mathbf{x} - \mathbf{y}\| \quad (1-4)$$

但是如果直接使用仿射包进行分类的话容易出现两个仿射包相交的情况，也就是对噪声不够鲁棒，所以针对这个问题 [11] 在文献中提出了使用 Convex Hull：

$$H_k^c = \left\{ \mathbf{x} | \mathbf{x} = \sum_{i=1}^n \alpha_{ki} \mathbf{x}_{ki}, \sum_{i=1}^n \alpha_{ki} = 1, L < \alpha_{ki} < U \right\} \quad (1-5)$$

来建模表示的方案，其中 L, U 分别表示上界和下界（标量），这样做的目的是将 α_{ki} 限制在了一定的范围内来提高了模型对噪声的鲁棒性。

针对 Affine Hull 对样本不鲁棒的问题，[26] 提出了使用稀疏表示的方案来解决：

$$\begin{cases} F_{\mathbf{v}_i, \mathbf{v}_j} = \|(\boldsymbol{\mu}_i + U_i \mathbf{v}_i) - (\boldsymbol{\mu}_j + U_j \mathbf{v}_j)\|_2^2 \\ G_{\mathbf{v}_i, \boldsymbol{\alpha}} = \|(\boldsymbol{\mu}_i + U_i \mathbf{v}_i) - X_i \boldsymbol{\alpha}\|_2^2 \\ Q_{\mathbf{v}_j, \boldsymbol{\beta}} = \|(\boldsymbol{\mu}_j + U_j \mathbf{v}_j) - X_j \boldsymbol{\beta}\|_2^2 \\ \min_{\mathbf{v}_i, \mathbf{v}_j, \boldsymbol{\alpha}, \boldsymbol{\beta}} (F_{\mathbf{v}_i, \mathbf{v}_j} + \gamma(G_{\mathbf{v}_i, \boldsymbol{\alpha}} + Q_{\mathbf{v}_j, \boldsymbol{\beta}}) + \lambda_1 \|\boldsymbol{\alpha}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_1) \end{cases} \quad (1-6)$$

这样做虽然使得模型对噪声更鲁棒，但是也带来计算复杂度太高的问题；所以 [65] 提出了使用 l_p 范数（通常 $p = 2$ ）来代替 l_1 范数：

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} (\|X \boldsymbol{\alpha} - Y \boldsymbol{\beta}\|_2^2 + \lambda_1 \|\boldsymbol{\alpha}\|_{l_p} + \lambda_2 \|\boldsymbol{\beta}\|_{l_p}), s.t. \sum_k \alpha_k = 1, \sum_k \beta_k = 1 \quad (1-7)$$

而文章 [60] 则使用了高斯模型来增强模型的鲁棒性，使得在最小的误差情况下还要求样本属于该类的概率最大。

最后用图1.5总结一下几者之间的关系：

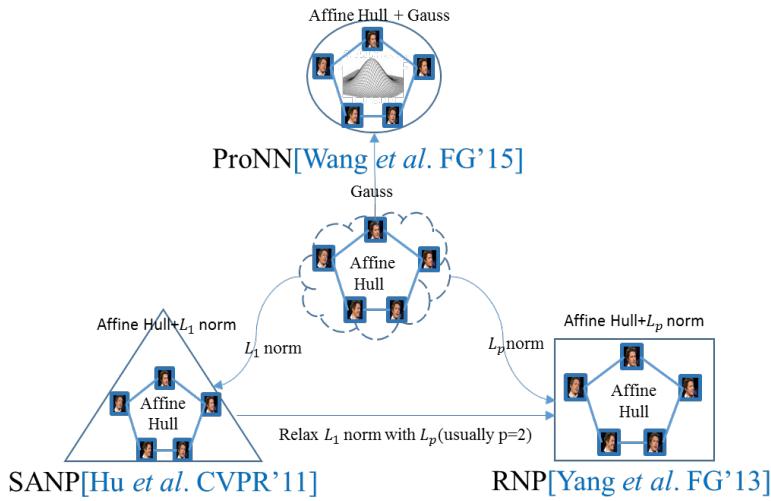


图 1.5 仿射包建模图像集合方法关系图

用文字总结起来就是：仿射包方法使用仿射包建模图像集合，为了克服直接使用仿射包分类对噪声不鲁棒问题，不同的限制被添加从而衍生出了不同的方法。

1.3.3 统计建模图像集合的方法

统计量建模的方法是近年来研究图像集合分类问题主流的方法之一，它以其优越的表现受到越来越多的关注；此外由于统计建模时，数据表示的特殊性（对称正定矩阵（SPD 矩阵），分布函数等），黎曼流形成为了主要的研究工具，这也促进了非线性数据表示的研究。

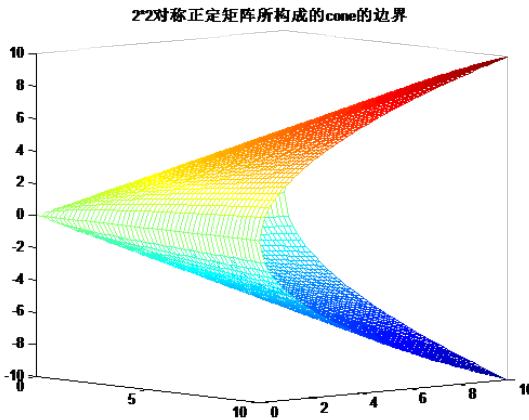
统计建模的方法又可以细分为：单一统计量建模的方法，多统计模型融合的方法以及基于分布函数的方法：在单统计量表示图像集合的方法中，协方差矩阵被认为是丰富而有效的特征表示，样本协方差矩阵： $C = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})$ ，其中 $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ ，为样本均值（协方差矩阵都要求是正定的 $C > 0$ ，由于该特殊性，其并不构成欧式空间），图 1.6 给出了一个简单的关于对称正定的例子（ 2×2 对称正定矩阵）^①。

协方差建模图像集合的代表作 [57] (Wang and et al. CVPR'12)：利用核函数 $\phi = \log(\cdot)$ 将协方差矩阵流形空间映射到 RKHS 空间，在新的 RKHS 空间中，使用 KPLS[49] 回归以及 KDA[7] 对数据进行分类。

文献 [22] 则在对称正定矩阵 (SPD) 集合中，利用黎曼度量进行协方差降维及判别学习，算法的优化的空间（投影矩阵所在的空间）为 Grassmann 流形空间。

多统计模型融合的方法主要的代表作有：[42] 和 [29]，两者的思想比较近似，主要思想是：不同的统计模型会刻画目标的不同侧面，包含了不同的信息，融合它们可以得到目标的更全面的表示，具体的实现则是利用核函数将不同的统计模型映射到统一的 RKHS(Reproducing Kernel Hilbert Space) 空间中，然后再在新的 RKHS 中利用融合的特

^① 需要注意的是： 2×2 对称正定矩阵组成的集合本身并不包含这些外边界（因为它是开集），而是该边界包住的整个锥的内部

图 1.6 2×2 对称正定矩阵的外边界在 3 维空间中的结构

征进行分类，其核心内容可以用图形1.7描述。

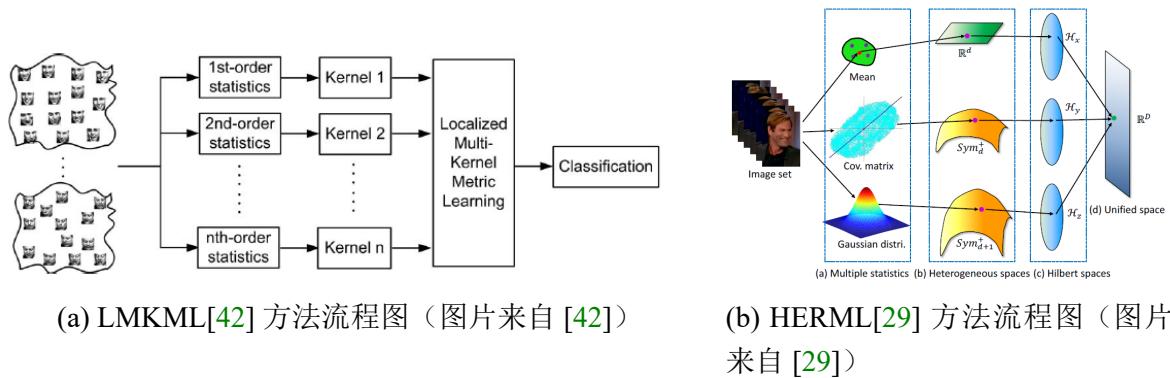


图 1.7 多统计模型建模图像集合方法

最后是基于分布函数建模的方法：为了更好的挖掘图像集合原始分布的信息，分布函数建模的方法被提出：文献 [59] 使用高斯混合模型（GMM）表示图像集合（逼近原始分布），利用核映射将 GMM 的各个 component 映射到 RKHS 中，并在 **KDA(Kernel Discriminant Analysis)**[7] 下学习投影矩阵，最后在投影空间中分类；而文献 [20] 则使用 **KDE(Kernel Density Estimation)** 表示图像集合（逼近原始分布），并设计距离/散度来度量两个图像集合的 KDE 表示的距离，为了使得 KDE 估计可靠，文章中还为数据学习一个具有判别性的降维矩阵 W 来辅助估计。

统计建模图像集合的方法小结：1) 统计建模的方法从最初的单统计量模型出发，经过发展逐步形成多统计模型融合以及分布函数建模图像集合等一系列方法；2) 由于表示的特殊性，统计模型的数学表示往往与黎曼流形相关联，黎曼流形成了研究它们的一个重要工具。图1.8描述了已有的统计建模图像集合的方法的一些关系：

1.3.4 深度学习的方法

深度学习在众多的领域都取得了不小的成果，所以也有学者将深度学习（**DL/Deep Learning**）的方法用于图像集合分类，目前这种尝试的主流做法是为每类学一个网络，并

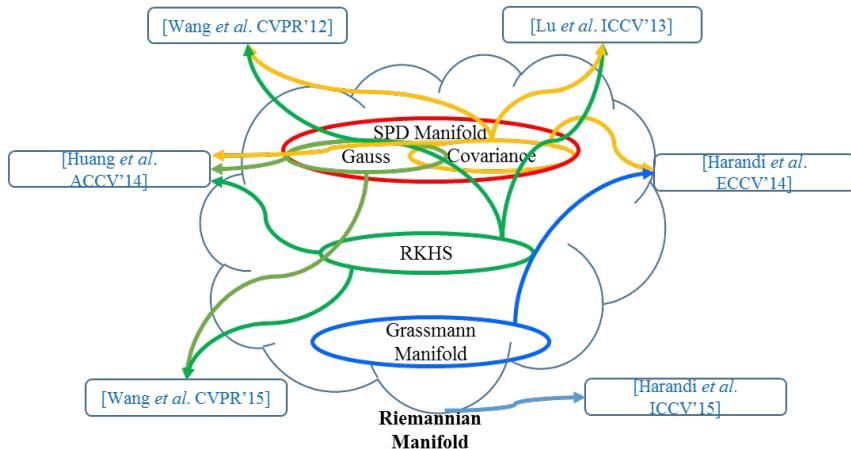


图 1.8 统计建模图像集合的方法间的关系

期望深度网络能够学到原始流形的 geometry 的结构。两个代表性的工作是：文献 [24] 为每类学一个 AE-Like(Unsupervised) 网络（网络结构如图 1.9(a) 所示），自动挖掘 Manifold 的 Geometry 结构，最终利用重建误差以及 voting 进行分类。

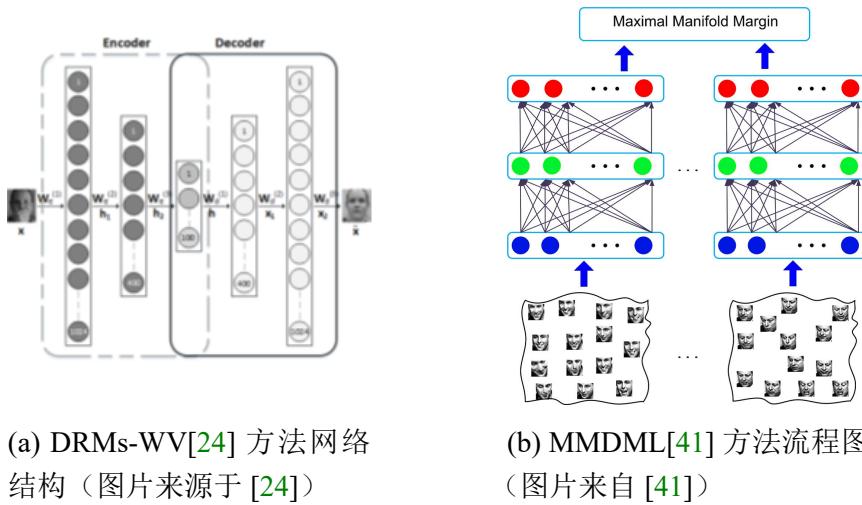


图 1.9 深度学习建模图像集合的代表性网络结构

文献 [41] 为每类学一个 DNN-Like(Supervised) 网络，使得在输出层不同类的 margin 尽量大；其网络结构如图 1.9(b) 所示。

这里以 MMDML[41] 为例对该类方法做一个介绍：在 $L+1$ 层的 DNN 网络中，对于第 c 个图像集合的第 i 张图片其顶层输出为：

$$\mathbf{h}_{ci}^L = s(\mathbf{W}_c^L \mathbf{h}_{ci}^{L-1} + \mathbf{b}_c^L)$$

其中 \mathbf{W}_c^L 是投影矩阵， \mathbf{b}_c^L 是偏置向量， \mathbf{h}_{ci}^{L-1} 是上一层输出， $s(\cdot)$ 为非线性激活函数；MMDML[41] 方法在网络顶层最大化不同 manifold 之间的 margin；为此，工作 [41] 为每

类的每个样本（如第 c 类的第 i 个样本），**定义公式1-8**中的近邻关系描述量：

$$\begin{cases} D_1(\mathbf{h}_{ci}^L) = \frac{1}{K_1} \sum_{p=1}^{K_1} \|\mathbf{h}_{ci}^L - \mathbf{h}_{cip}^L\|_2^2 \\ D_2(\mathbf{h}_{ci}^L) = \frac{1}{K_1} \sum_{q=1}^{(K_2)} \|\mathbf{h}_{ci}^L - \mathbf{h}_{ciq}^L\|_2^2 \end{cases} \quad (1-8)$$

其中 \mathbf{h}_{cip}^L 表示的是第 p 个同类的近邻在网络顶层的输出， \mathbf{h}_{ciq}^L 表示的是第 q 个不同类的近邻在网络顶层的输出， K_1, K_2 是两个设置近邻个数的参数，所以上述式子定义了第 c 类的第 i 个样本在网络顶层与 K_1 同类近邻， K_2 个不同类近邻的关系。

接下来目标函数使得在顶层最大化不同 manifold 之间的 margin：

首先将网络参数表示为： $f_c = [W_c^1, W_c^2, \dots, W_c^L, b_c^1, b_c^2, \dots, b_c^L]$ ，后定义：

$$\begin{cases} H_1 = \sum_{c=1}^C \sum_{i=1}^{N_c} g(D_1(\mathbf{h}_{ci}^L) - D_2(\mathbf{h}_{ci}^L)) \\ H_2 = \sum_{c=1}^C \sum_{l=1}^L (\|W_c^l\|_F^2 + \|b_c^l\|_2^2) \end{cases}$$

最终得到**MMDL**的优化目标函数：

$$\min_{f_1, f_2, \dots, f_C} H = H_1 + \frac{1}{2} H_2$$

文献 [41] 中使用了随机次梯度下降算法进行了**优化**获得网络参数，在最后在测试阶段，测试样本的分类结果由公式**1-9**获得。

$$L_q = \arg \min_c d(X_q, X_c), 1 \leq c \leq C \quad (1-9)$$

其中 $X_q = [\mathbf{x}_1^q, \mathbf{x}_2^q, \dots, \mathbf{x}_{N_q}^q]$ 表示的是测试数据 X_c 表示的是训练数据，而 $d(X_q, X_c)$ 的计算过程如下：1) 使用网络将 \mathbf{x}_j^q 映射到新的空间 $\mathbf{h}_c(\mathbf{x}_j^q)$ ；2) 计算 $\mathbf{h}_c(\mathbf{x}_j^q)$ 与 $\mathbf{h}_{ci}^L, i = 1, 2, \dots, N_c$ 的欧式距离，并将最小的距离作为 \mathbf{x}_j^q 与第 c 个 manifold 的距离，最后对所有样本 $\mathbf{x}_1^q, \mathbf{x}_2^q, \dots, \mathbf{x}_{N_q}^q$ 求平均作为 $d(X_q, X_c)$ 。

1.3.5 国内外研究现状小结

总结国内外对图像集合分类问题的研究，**对图像集合分类问题可做如下描述**：1) 为图像集合设计一种表示（子空间、流形、仿射包、统计模型及深度网络等）；2) 为这种表示（模型）设计/定义一种距离度量，并用此进行判别学习；3) (可选) 针对已有模型问题（不鲁棒、维度太高……）做进一步改进。

另一方面，在图像集合分类问题的**中**，由于数据的特殊性，往往需要对非线性的数据进行研究，在子空间、统计建模的方法中，黎曼流形作为成熟的数学工具在其中发挥了重要的作用。

最后，还需要注意到除了前面介绍的一些方法，还有稀疏编码和协同表示的方法也为图像集合的分类问题的探索做出了重要贡献。

1.4 数据介绍

图像集合分类问题来源于实际问题，最终还是要回到实际问题中；所以只有理论还是不够，还需要数据的支撑；本小节就是对图像集合分类问题中的一些常用的数据集进行介绍；此外，多样化的数据也更能说明算法的有效性，并且由于本文对黎曼流形问题的研究过程中并没有限制在图像集合问题上，所以这里还会介绍一两个可用黎曼流形建模的数据集合，表格1.2中列出了本节将要介绍的数据库。

表 1.2 数据库列表

数据库	描述	备注
YouTube Faces DB[62]	1595 个人，3425 段视频，低分辨高压缩率	人脸数据库
YouTube Celebrity[36]	47 个人，1910 段视频，低分辨高压缩率	人脸数据库
UIUC[39]	4 个大类，18 个子类，每类 12 张图片	材料 (material) 数据库
ETH-80[37]	共 8 个子类每类 4 个图像集合每个集合 41 张图片	物体识别数据库
CMU MoBo[16]	25 个人，4 个运动（行走）类，150 段视频数据	为步态研究而搜集

图1.10给出这些数据集的一些示例图片。**YouTube Faces DB** 数据库 [62] 最初是为视频人



(a) YouTube Faces DB[62] 数据库示例



(b) YouTube Celebrity[36] 数据库实例



(c) UIUC[39] 数据库示例



(d) ETH80[37] 数据库示例



(e) CMU MoBo[16] 数据库示例

图 1.10 数据库示例

脸验证任务收集的，其收集过程中根据 LFW(Labeled Faces in the Wild)[27] 数据集的样本进行，数据库包含了 1595 个人的 3425 个视频段，是一个较大的视频数据库，这些数据均从 Youtube 上获得，最短的只有 48 帧，最长的则有 6070 帧；由于数据是从网络中收集的所以存在脸部姿态变化，表情变化等一系列问题。

Youtube Celebrity 数据集 [36] 也是从 YouTube 上收集而来，最初是为人脸跟踪和识别任务而收集的，数据集包含 47 个人的 1910 个视频段，属于一个较大的视频数据库。其数据呈现出低分辨率和高压缩率的特点，此外同样存在面部姿态变化，表情变化等问题。数据库的状况比较接近于真实情况，识别任务有不小的挑战。

UIUC 数据库 [39] 是材料数据库 (material database)，数据库的组织结构大致为：顶层是四个大类，分别是树皮、织物、建筑材料和动物的皮毛，下面分出 18 个子类，每个子类包含了 12 张图像，该数据集并不属于图像集合的数据集，但是若使用 Region Covariance 来表示的话，该数据集上的分类问题研究也将与 SPD 矩阵黎曼流形相关。

ETH80 数据库 [37] 主要用于物体识别任务，其中的数据从概念上属于 4 个大类：水果蔬菜类、动物类、(小型的) 人造的类别及 (大型的) 人造的类，具体采集了 8 个类别：苹果、牛、杯子、狗、马、梨、西红柿、和汽车，整个数据集包含 80 个图像集合 (每个类别 10 个集合)，每个集合包含 41 张图片，所以数据集总大小为 3280 张图片。

CMU MoBo 数据库 [16] 包含 25 个人在室内环境下，跑步机上的 6 个 view 行走姿态的 150 段视频数据，数据库中的数据主要有 4 中行走姿态 (摄像机帧率 30FPS)：慢速行走，快速行走，斜面行走以及带球行走。

在本节的最后简单的介绍一下测试协议的问题，其中由于 CMU MoBo[16] 提出的时间相对比较早，所以这里不再介绍它的测试协议，此外该数据集也不会在实际的实验中使用。而 YouTube Faces DB 数据库由于做的是人脸验证任务，与分类识别任务有所区别，在本文的实际实验中也没有列入，但是考虑到今后可能会用到这个数据库 (毕竟它是相当大的视频人脸数据库) 所以这里也会简单介绍一下它的测试协议；在所有的数据集上均采用了 10 折交叉验证，最后报告的结果均是 10 则交叉验证的平均结果，其中在每一次验证过程中我们将训练集称为 Gallery 测试集称为 Probe，在各个数据集合上对数的划分情况如下：在 YTF (YouTube Faces DB) 上根据文献 [62] 中的方式，将数据库提供的 5000 对视频对平均分为 10 份 (每份 500 对，其中 250 对的每对是同一个人，另外 250 对的每对不是同一个人)。在 YTC (YouTube Celebrity) [36] 上数据的划分则是参考了 [57] 以及 [54] 的数据划分方式，在 YTC 数据集合上对每个人随机选取 3 个视频段作为训练 (Gallery) 6 个视频段作为测试 (Probe)，然后将这个过程重复进行 10 次来获得数据集的随机划分。在 UIUC 数据库 [39] 上的数据划分比较简单，我们随机从每个子类中选取一半的样本做为训练集 (Gallery) 剩下的一半的数据作为测试集 (Probe)，然后也重复这个过程 10 次得到 10 次验证的数据。ETH80 数据上的数据划分也参考了 [57]，其上的数据划分是在每个类别中随机选取 5 个图像集合作为训练集 (Gallery) 剩下的 5 个作为测试集 (Probe)，并重复这个过程 10 次作为 10 次验证的数据。以上便是本文中使用的数据库的数据划分方式和测试协议。

1.5 本文的组织结构

在本章的最后，我们来介绍一下本文的组织结构：第一章是绪论，这一章主要介绍问题的背景意义以及国内外的研究现状；作为本文的第一章主要目的是引领读者对图像集合分类问题有一个宏观的了解，也为后续自己工作的介绍做准备。

第二章将介绍矩阵函数与黎曼流形上的优化问题，这部分的内容既包含黎曼流形，矩阵函数等这样的基本概念也对矩阵函数和流形优化等一般化的问题进行了初步的探究。并结合本文其它两个研究内容中的一些实际问题对其进行展开，目的是方便读者理解和实现本文中的提到的方法和概念的同时，并希望读者在遇到类似问题的时候能够从中获得启示。

第三章介绍黎曼流形上的偏最小二乘问题，这一章的内容会从欧式空间的偏最小二乘问题开始介绍，然后借助投影的一般形式将其扩展到黎曼流形空间中得到黎曼流形上偏最小二乘问题的基础版本的，紧接着是结合流形的特点以及考虑到数据的稀疏性问题，从基础版本的黎曼流形上的偏最小二乘问题出发发展出了多切空间逐步回归的偏最小二乘学习方法，最后实验验证了该方法。

第三章简单回顾了使用子空间和协方差矩阵表示图像集合的方法的缺点后，考虑使用半正定（PSD）矩阵表示图像集合，在对已有工作 [43] 做简单回顾与探讨后，从 Fixed-Rank PSD 矩阵表示出发研究了使用 Low-Rank PSD 矩阵表示图像集合的方法，最终的实验结果支持了最初的设想。

第五章总结和讨论前几章的内容，对现有研究做了回顾与不足之处的分析，并对下一步可能的方向做了讨论和展望。

第二章 矩阵函数的导数计算与矩阵流形上的基本优化方法

《论语》有云：“工欲善其事，必先利其器”。本文把黎曼流形作为主要的研究工具（对象），则必然涉及到其上的优化问题，为此需要对该问题进行探索，同时本章的内容也是一个一般化的问题而并不仅限于本文中的应用，最后这部分内容也是对研究课题中流形上的优化问题的归纳和总结。当然这里不会像 [9] 或 [3] 中那样详细的介绍黎曼流形上的优化问题。这里首先会介绍黎曼流形这个基础的概念，然后探究矩阵函数和它们的导数计算问题，最后结合研究课题的主要工作以及实际中遇到的一些问题和例子进行说明，目的是方便读者理解作者硕士期间研究课题的同时也更容易举一反三地在遇到类似问题的时候能够从中提炼出思路或解决方案；当然，对于流形上很细节的问题读者还是到文献 [3,9] 中寻找答案会更合适。此外，本章的内容不会对算法的收敛性等这样专业的问题作讨论，因为这既非作者所长也不是这里的写作目的，并且这可能会让内容过于专业化而变得枯燥乏味。

2.1 黎曼流形简介

黎曼流形作为本工作的主要研究对象之一，在这一节将对其进行简要介绍。本节的内容主要包括基本流形和黎曼流形的基本定义和性质；最后将就黎曼流形中的 SPD(Symmetric Positive Define) 矩阵流形做进一步的介绍，并着重介绍 SPD 流形上的两个重要的度量 (Affine Invariant Metric 和 log-Euclidean Metric)（本节的一些内容尤其是一些流形上的基本定义的介绍参考了 [2] 和 [1]）。

2.1.1 黎曼流形

流形是数学上的一个抽象的概念，而它的定义则依赖于另一个更抽象的概念——拓扑空间：

定义 2.1（拓扑空间） 设 S 表示一个集合， τ 也是一个集合且 τ 中的元素满足：

1. $\emptyset, S \in \tau;$
2. τ 中有限个元素的交仍然属于 τ
3. τ 中任意多个元素的并仍然属于 τ

在数学上这样的 τ 称为 S 上的拓扑结构，并且称 τ 中的元素为开集；拓扑的研究中，点集拓扑是一个重要的内容，其中需要理解的有两个概念（系统的相关内容可以参看 [2]），第一个概念（空间是 A_2 的）：如果一个拓扑空间具有可数拓扑基则这样的拓扑空间^① 称

^① 具有这样性质的空间也叫做第二可数的

为 A_2 的；第二个概念（空间是 T_2 的）：如果一个拓扑空间具有 Hausdorff 性质则这样的拓扑空间称为 T_2 的。Hausdorff 性质：假设 S 是拓扑空间，设 x 和 y 是 S 中的点，我们称 x 和 y 可以“由邻域分离”，如果存在 x 的邻域 U 和 y 的邻域 V 使得 U 和 V 是不相交的 $U \cap V = \emptyset$ ，且任何两个 S 中不同的点都可以有这样的邻域分离，那么称 X 是豪斯多夫 (Hausdorff) 空间，因此豪斯多夫空间又叫做分离空间。**介绍完拓扑空间之后，为了介绍流形，接下来会集中定义一些基本概念为介绍流形做准备。**

定义 2.2 (r 阶连续) 若一函数是连续的，则称其为 C^0 函数；若函数存在连续导函数，即**连续可导**，则被称为 C^1 函数；若一函数 r 阶可导，并且其 r 阶导函数连续，则为 C^r 函数 $r \geq 1$ 。而光滑函数是对所有 r 都有 r 阶的连续导数，并记为 C^∞ 函数。

定义 2.3 (同胚) 两个拓扑空间 $\{X, \tau_X\}$ 和 $\{Y, \tau_Y\}$ 之间的函数 $f : X \rightarrow Y$ 称为同胚，如果它具有下列性质：

1. f 是双射（单射和满射）；
2. f 是连续的；
3. 反函数 f^{-1} 也是连续的（ f 是开映射）。

关于同胚，此处定义参考维基百科^① 以及文献 [2]。

定义 2.4 (C^r 流形) 设 M 是具有 A_2, T_2 性质的拓扑空间，如果存在 M 的开覆盖 $\{U_\alpha\}, \alpha \in \Gamma$ 以及相应的连续映射族 $\varphi_\alpha : U_\alpha \rightarrow \varphi_\alpha(U_\alpha)$ ；使得：

1. $\varphi_\alpha : U_\alpha \rightarrow \varphi_\alpha(U_\alpha) \subset \mathbb{R}^n$ 为从 U_α 到欧式空间开集 $\varphi_\alpha(U_\alpha)$ 上的同胚^②；
2. 当 $U_\alpha \cap U_\beta \neq \emptyset$ 时，若如下的转换映射：

$$\varphi_\beta \circ \varphi_\alpha^{-1} : \varphi_\alpha(U_\alpha \cap U_\beta) \rightarrow \varphi_\beta(U_\alpha \cap U_\beta)$$

为 $C^r (r \geq 1)$ 映射，则称 M 为 C^r 流形；

特别地，若 $r = 0$ 则称 M 为拓扑流形，又若 $r \geq 1$ 则称 M 为 C^r 微分流形，进一步令 $\mathcal{D} = \{(U_{\alpha \in \Gamma}, \varphi_\alpha)\}$ ；若 \mathcal{D} 是最大的，即：若 M 的坐标卡 (U, φ) 与 \mathcal{D} 中的每一个坐标卡都是 C^r 相容^③ 的，则有 (U, φ) 属于 \mathcal{D} ，这样的 \mathcal{D} 称为拓扑流形 M 的一个 C^r 微分构造或微分结构（该部分总结自参考文献 [2] 第一章的定义 1.1.1）。

定义 2.5 (a) (切向量与切空间) 记 $C^\infty(M)$ 为微分流形 M 上光滑函数的全体组成的向量空间。设 $p \in M$ ，如果线性映射 $X_p : C^\infty(M) \rightarrow \mathbb{R}$ 满足以下条件：

$$X_p(f \circ g) = X_p(g)f(p) + g(p)X_p(f), \forall f, g \in C^\infty(M)$$

① <https://zh.wikipedia.org/wiki/%E5%90%8C%E8%83%9A>

② 这里的 n 称为流形 M 的维度，记为 $\dim(M) = n$

③ 设 U 为 M 上的开集， $\varphi : U \rightarrow \mathbb{R}^n$ 为连续映射，且 φ 的像为开集， φ 到其像上是同胚。如果 φ 和 φ_α 之间的转换映射均为 C^r 的，则称 (U, φ) 和局部坐标覆盖 $(U_\alpha, \varphi_\alpha)$ 是 C^r 相容的（摘自 [2]）

则称 X_p 为 p 处的切向量。切向量的全体组成的向量空间记为 p 处的切空间 $T_p M$ 。

上述的定比较比较晦涩，下面是关于切向量的另一个更加直观的定义形式：

定义 2.5 (b) (切向量与切空间) 设 $p \in M$ ，是流形 M 上的一点，经过 p 的光滑曲线 $\sigma : (-a, a) \rightarrow M$, 使得 $\sigma(0) = p$ 。定义 $\sigma'(0) \in T_p M$:

$$\sigma'(0)f = \frac{d}{dt}|_{t=0}[f \circ \sigma(t)], \forall f \in C^\infty(M).$$

容易验证 $\sigma'(0)$ 为 p 处的切向量，称为 σ 的初始切向量，也记为 $\dot{\sigma}(0)$ 。

定义 2.6 (黎曼流形) 对任意 $p \in M$, 如果映射 $g_p : T_p M \times T_p M \rightarrow \mathbb{R}$ 满足条件:

1. $\forall \mathbf{x}_p \in T_p M, g_p(\mathbf{x}_p, \mathbf{x}_p) \geq 0$, 等号成立当且仅当 $\mathbf{x}_p = 0$;
2. $\forall \mathbf{x}_p, \mathbf{y}_p \in T_p M$, 均有 $g_p(\mathbf{x}_p, \mathbf{y}_p) = g_p(\mathbf{y}_p, \mathbf{x}_p)$ 。

即 g_p 定义了切空间 $T_p M$ 上的内积，称 g 为 M 上的黎曼度量， (M, g) 称为黎曼流形。

定义 2.7 (曲线长度与距离) 沿用前面的定义，设 $\sigma(t) : [a, b] \rightarrow M$ 表示黎曼流形 (M, g) 上的一条链接 p, q 的 C^1 曲线，其中自变量 $t \in [a, b]$ ；定义曲线 σ 的长度为：

$$L(\sigma) = \int_a^b \|\dot{\sigma}(t)\| dt \quad (2-1)$$

其中 $\dot{\sigma}(t)$ 与定义 2.5 中的意义相同且 $\|\dot{\sigma}(t)\| = g(\dot{\sigma}(t), \dot{\sigma}(t))^{1/2}$ ；最后利用公式 2-1 定义距离如公式 2-2 所示。

$$d(p, q) = \inf_{\sigma} L(\sigma) \quad (2-2)$$

此外，这里记录另一条对于计算距离的有用性质：在等距同构的映射下，新空间中最短的测地线仍然是原来空间中的最短的测地线的长度。

2.1.2 对称正定矩阵 (SPD) 流形

对称正定矩阵流形是本文研究的主要对象之一，在前面的 1.3.3 小节也有提到：统计建模图像集合的时候，其最终的数学形式往往以对称正定矩阵（Symmetric Positive Definite Matrices）的形式存在。而已经有充分的数学理论支撑，SPD 矩阵空间在适合的度量定义下构成黎曼流形，其中最有名也最常用的是 AIM[46] (Affine-Invariant Metric) 和 LEM[5] (Log-Euclidean metric)。下面是对称正定矩阵的定义。

定义 2.8 (对称正定矩阵集合) 由 $d \times d$ 的对称正定矩阵构成的集合

$$\mathbb{S}_d^+ = \left\{ A | A \in \mathbb{R}^{d \times d}, \mathbf{x}^T A \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^d \text{ and } \mathbf{x}^T A \mathbf{x} = 0 \text{ if } \mathbf{x} = 0 \right\} \quad (2-3)$$

当赋上适当的度量 g (如 AIM[46] 或 LEM[5]) 之后即构成黎曼流形空间 (\mathbb{S}_d^+, g) 。

在 PSD 矩阵流形上, AIM 度量 [46] 和 LEM 度量 [5] 各自的对数 Log 和指数 Exp 函数及距离分别由公式2-4和2-5描述。

$$AIM : \begin{cases} \exp_{X_1}(H) = X_1^{\frac{1}{2}} \exp(X_1^{-\frac{1}{2}} H X_1^{-\frac{1}{2}}) X_1^{\frac{1}{2}} \\ \log_{X_1}(X_2) = X_1^{\frac{1}{2}} \log(X_1^{-\frac{1}{2}} X_2 X_1^{-\frac{1}{2}}) X_1^{\frac{1}{2}} \\ \delta_A(X_1, X_2) = \langle \log_{X_1}(X_2), \log_{X_1}(X_2) \rangle_{X_1} \\ \quad = \|\log(X_1^{-\frac{1}{2}} X_2 X_1^{-\frac{1}{2}})\|_F \end{cases} \quad (2-4)$$

其中 H 是 X_1 处的切空间上的切向量, $\langle \cdot, \cdot \rangle_{X_1}$ 表示 X_1 的切空间上的黎曼度量 (内积)

$$LEM : \begin{cases} \exp_{X_1}(T_2) = \exp(\log(X_1) + D_{X_1} \log(T_2)) \\ \log_{X_1}(X_2) = D_{\log(X_1)} \exp . (\log(X_2) - \log(X_1)) \\ \delta_l(X_1, X_2) = \langle \log_{X_1}(X_2), \log_{X_1}(X_2) \rangle_{X_1} \\ \quad = \|\log(X_1) - \log(X_2)\|_F \end{cases} \quad (2-5)$$

其中 T_2 是 X_1 处的切空间上的切向量, $\langle \cdot, \cdot \rangle_{X_1}$ 表示的是 X_1 的切空间上的黎曼度量 (内积), 其中 $D_{\log(X)} \exp . = (D_X \log .)^{-1}$ 是由等式 $\log \circ \exp = I$ (I 是单位矩阵) 得出, 更多详细信息可参看文献 [5]。

2.2 优化问题与梯度

本节会从一般的优化问题开始, 以梯度相关的问题结束。第一部分是优化问题的介绍; 由于优化问题多种多样, 所以这里选取具有代表性的一类问题: Lagrange 对偶问题, 接着第二部分会以梯度为主线介绍梯度在优化问题中运用, 最后再介绍梯度下降与共轭梯度算法。

2.2.1 Lagrange 对偶问题

Lagrange 对偶问题的转化是求解带约束问题重要方法, 在优化问题中有有着重要的地位, 其转化过程可以描述如下: 首先, 假设有如下形式的原问题:

$$\begin{aligned} & \min f_0(\mathbf{x}) \\ & s.t \quad f_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m \\ & \quad h_i(\mathbf{x}) = 0, i = 1, 2, \dots, p \end{aligned} \quad (2-6)$$

对于原问题2-6, 其对应的 Lagrange 函数定义如式2-7所示:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\mathbf{x}) + \sum_{i=1}^m \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^p \nu_i h_i(\mathbf{x}) \quad (2-7)$$

由 Lagrange 函数²⁻⁷定义原问题的 Lagrange 对偶函数^①:

$$g(\lambda, v) = \inf_x \mathcal{L}(\mathbf{x}, \lambda, v) = \inf_x \left(f_0(\mathbf{x}) + \sum_{i=1}^m \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^p v_i h_i(\mathbf{x}) \right) \quad (2-8)$$

其中 \inf 是下确界的意思，其意义类似于最小值，但是稍有不同的是：例如对于开区间 $(0, 1)$ 它的最小值是不存在的，但是它的下确界却是存在的 $0 = \inf\{(0, 1)\}$ 。关于对偶问题²⁻⁸需要了解的是：对偶问题²⁻⁸对任意的 $\lambda \geq \mathbf{0}$ 以及 v 都是原问题²⁻⁶的下界，此外对于 $\lambda < \mathbf{0}$ 的情形这将导致 $g(\lambda, v)$ 失去实际意义。

既然 (λ, v) 对于任意的 $\lambda \geq \mathbf{0}$ 以及 v 是原问题²⁻⁶的下界，那么什么样的 λ, v 才是好的，对偶问题考虑：

$$\max g(\lambda, v), \text{s.t } \lambda \geq \mathbf{0} \quad (2-9)$$

所以实际上的 Lagrange 对偶问题求解的是问题²⁻¹⁰。

$$\max_{\lambda, v} \left(\min_x \mathcal{L}(\mathbf{x}, \lambda, v) \right) \quad (2-10)$$

最后，对偶问题与原问题的解的关系可由 KKT (公式²⁻¹¹) 条件刻画：

$$\begin{cases} \nabla f_0(\mathbf{x}) + \sum_{i=1}^m \lambda_i \nabla f_i(\mathbf{x}) + \sum_{i=1}^p v_i \nabla h_i(\mathbf{x}) = \mathbf{0} \\ f_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m \\ h_i(\mathbf{x}) = 0, i = 1, 2, \dots, p \\ \lambda_i \geq 0, i = 1, 2, \dots, m \\ \lambda_i f_i(\mathbf{x}) = 0, i = 1, 2, \dots, m \end{cases} \quad (2-11)$$

至此 Lagrange 对偶问题的介绍基本结束，接下来的部分是以梯度为主要介绍优化问题相关内容。

2.2.2 梯度计算问题

在^{2.2.1}部分给出了目标函数，接下来是目标函数的优化问题；这里为了方便理解将所有参数都归结到一起并用 \mathbf{x} 表示，并且这里只考虑最小化的问题 $\min_{\mathbf{x}} f(\mathbf{x})$ ，对于有约束的问题大部分可以利用^{2.2.1}部分的内容进行转化，还要一部分会与实际问题有关，本章后续的内容会涉及部分（如对称正定约束），这里不做过多介绍。

首先，优化问题中导数计算的重要性不言而喻，在一些比较特殊的的情况下通过导数甚至可以得到问题的解析解；这也是这里花篇幅介绍导数的原因，同时也是为后续矩阵函数的导数计算做铺垫。公式²⁻¹²给出方向导数的定义（这里假设 \mathbf{x} 是一个向量，因

^① Lagrange 对偶函数是一族仿射函数的下界，所以它是凹函数，具体细节请参看《Convex optimization》

为这往往是最普遍的情形，但是不仅限于向量）：

$$Df(\mathbf{x})[\mathbf{d}] = \lim_{h \rightarrow 0} \frac{f(\mathbf{x} + h\mathbf{d}) - f(\mathbf{x})}{h} \quad (2-12)$$

其中 $Df(\mathbf{x})[\mathbf{d}]$ 表示的是 $f(\mathbf{x})$ 沿 \mathbf{d} 方向的方向导数，方向导数的定义自然的包含了偏导数的定义 $\frac{\partial}{\partial x_i} f(\mathbf{x})$ ，这里不在赘述；此外公式2-12另一个重要的用途是 gradient check，当把 h 取得很小的时候（一般取 10^{-3} 或者更小），将公式2-12计算得到的值 $g(h) = \frac{f(\mathbf{x} + h\mathbf{d}) - f(\mathbf{x})}{h}$ 与利用求导公式计算得到的导数值进行比较，当小于一定误差限的时候认为计算导数的公式是正确的。此外，当 $f(\mathbf{x})$ 的函数形式特别复杂使得导数难以计算的时候，利用公式2-12还可以计算 $f(\mathbf{x})$ 的数值导数来代替导数作为算法的输入。

下面用几个关于求均值的例子对问题进行简要的说明，首先注意到关于均值的计算基本上可以统一的如下的优化问题的框架中：

$$\mu = \min_{\mathbf{x}} \frac{1}{n} \sum_{i=1}^n \text{dist}^2(\mathbf{x}, \mathbf{x}_i) \quad (2-13)$$

公式2-13又叫做 Fréchet Varaince，Fréchet Mean 则是唯一使得上式达到最小的点，而上式的局部极小点则一般称为 Karcher Mean。对于不同的 $\text{dist}(\cdot, \cdot)$ 的定义这里会得到不同的 Fréchet Mean；例如当 $\text{dist}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$ 的时候，其最优解在 $\mu = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ 处到达，也就是上述优化问题的解析解；又或者当 $X_i \in \mathbb{S}_d^+$ 的时候在 Log-Euclidean Distance[5] 的距离计算框架下，问题2-13也有解析解 $\mu = \exp\left(\frac{1}{n} \sum_{i=1}^n \log(X_i)\right)^{\textcircled{1}}$ 。

但是对于 $X_i \in \mathbb{S}_d^+$ 且 $n > 2$ 的时候，在 Affine-Invariant Distance[46] 的距离计算框架下，问题2-13却没有解析解，甚至于不能保证问题的唯一极小值点的存在，此时最优化的方法就发挥作用了^②。关于 \mathbb{S}_d^+ 在 AID[46] 距离计算框架下的均值的计算，读者可在本章的后续部分看到详细的过程。

2.2.3 梯度下降和共轭梯度

本节会简单的介绍一下梯度下降和共轭梯度算法，这里的目的除了保持本节完整性外还是为后续章节介绍黎曼流形上的这两种方法做准备。

2.2.3.1 梯度下降算法

关于梯度下降算法这里首先从泰勒展开说起：

定义 2.9 设 $f(x)$ 是实数域上的无穷可微函数，那么它在 x_0 点泰勒展开式表示为：

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots \quad (2-14)$$

① 这里使用 μ 表示对称正定矩阵的均值，主要是出于约定俗成的考虑

② 当然对于有解析解的问题优化算法也是适用的，不过几乎没有这样做的必要

同样的对于无穷可微向量函数 $f(\mathbf{x})$ 以及初始点 \mathbf{x}_0 有^①：

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \left(\frac{\partial f(\mathbf{x}_0)}{\partial \mathbf{x}} \right)^T (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2!} (\mathbf{x} - \mathbf{x}_0)^T \mathbf{H} (\mathbf{x} - \mathbf{x}_0) + \dots \quad (2-15)$$

关于梯度下降，首先需要注意到的是，梯度下降的两个参数：方向 \mathbf{g} 和步长 α ，为了方便起见这里不妨假设 $\|\mathbf{g}\| = 1$ （虽然在实际中并不一定做归一化）。

梯度下降算法的目的是使得 $f(\mathbf{x}_0 + \alpha \mathbf{g})$ 最小，于是利用一阶泰勒展开近似 $f(\mathbf{x})$ 得到：

$$f(\mathbf{x}_0 + \alpha \mathbf{g}) \approx f(\mathbf{x}_0) + \alpha \left(\frac{\partial f(\mathbf{x}_0)}{\partial \mathbf{x}} \right)^T \mathbf{g} \quad (2-16)$$

上式右边当 $\mathbf{g} = -\left(\frac{\partial f(\mathbf{x}_0)}{\partial \mathbf{x}} \right) / \|\frac{\partial f(\mathbf{x}_0)}{\partial \mathbf{x}}\|$ 的时候是 $f(\mathbf{x}_0 + \alpha \mathbf{g})$ 较 $f(\mathbf{x}_0)$ 下降最快的方向，所以梯度下法也叫最速下降法。

至于步长 α 的选择则是一个一维的优化问题，这个问题比较简单（二分法，0.618 法等都可以解决），当然也可以是预先定义的。

对于凸函数由于有： $f(\mathbf{x}) \geq f(\mathbf{x}_0) + \left(\frac{\partial f(\mathbf{x}_0)}{\partial \mathbf{x}} \right)^T (\mathbf{x} - \mathbf{x}_0)$ ，所以梯度下降算法可以保证目标函数值是不增加的。

2.2.3.2 共轭梯度算法

共轭梯度算法的提出克服了梯度下降慢以及牛顿法存储要求高和 Hessian 阵难以计算的问题。该算法最初由 Hestenes 和 Stiefel 于 1952 年为求解线性方程组而提出的，后来，人们把这种方法用于求解无约束最优化问题，使之成为一种重要的最优化方法。共轭梯度算法有很多种模式，其中 Fletcher-Reeves 共轭梯度法 [15]（简称 FR 法）是其中的一种，接下来就以该模式的算法为例进行说明。

关于 Fletcher-Reeves 共轭梯度法需要注意的是：最开始的方向需要由最速下降法（梯度下降法）获得 $\mathbf{d}_0 = \nabla f(\mathbf{x}_0)$ ，一般地对于第 $k+1$ 次迭代，已知 $\mathbf{x}_k, \mathbf{d}_k$ ，则 $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$ ，其中 λ_k 满足：

$$\lambda_k = \min_{\lambda} f(\mathbf{x}_k + \lambda \mathbf{d}_k) \quad (2-17)$$

然后计算 $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$ ，并利用如下的更新公式更新搜索的方向 [15]：

$$\begin{cases} \mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k \\ \beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k} \end{cases} \quad (2-18)$$

关于算法的细节读者可以参看原文 [15] 中的相关内容；最后关于 Fletcher-Reeves 共轭梯度法需要注意的是对于高于二次的目标函数，目标函数可能存在局部极小点并破坏二次

^① 其中 \mathbf{H} 矩阵就是通常意义的 Hessian 矩阵

截止性（对于二次及以下的函数共轭梯度算法会在 d 次迭代内找到精确解），此时需要重启算法以完成极值点的搜索。

2.3 矩阵函数的导数计算

本部分的内容主要是针对矩阵函数的导数计算的内容以及对称正定矩阵流形上的导数计算；矩阵函数的计算由于自变量的特殊性（一般为矩阵），有其特别的地方。在“The matrix cookbook”[47] 中对矩阵求导计算中的大多数情况做了说明，但是对于一些比较特殊的情况仍然不能很好的解决（如 SPD 矩阵中在 AID 距离计算框架下最小化 Fréchet Variance 的最小化问题），需要更一般的解决方案；接下来的内容主要参考文献 [9] 和 [3]。

2.3.1 矩阵函数求导的一般形式

对于任意的矩阵 $A \in \mathbb{S}_d$ ($d \times d$ 的实对称矩阵组成的集合) 并假设 A 有 svd 分解 $A = U\Lambda U^T$ ，以及光滑实值函数 $f(x)$ ，这里设 $f(x)$ 的 Taylor 展开式如2-19所示。

$$f(x) = \sum_{k=0}^{\infty} \alpha_k x^k \quad (2-19)$$

利用公式2-19矩阵函数 $f(A)$ 可以由下式定义：

$$f(A) = \sum_{k=0}^{\infty} \alpha_k A^k = U \text{diag}(f(\lambda_1), f(\lambda_2), \dots, f(\lambda_d)) U^T \quad (2-20)$$

为了计算 $\nabla_A f(A)$ ，参照 [9] 中的内容，这里先从方向导数开始2-12，并先给出两条计算方向导数的规律 [9]：

$$\begin{cases} \text{rule 1 : } D(f \circ g)(X)[H] = Df(g(X))[Dg(X)(X)[H]] \\ \text{rule 2 : } D \langle f(X), g(X) \rangle (X)[H] = \langle Df(X)(X)[H], g(X) \rangle + \langle f(X), Dg(X)(X)[H] \rangle \end{cases} \quad (2-21)$$

在矩阵优化问题中关于方向导数公式2-12，需要注意的是此时的自变量为矩阵而公式4-32中的内积定义 $\langle \cdot, \cdot \rangle$ 最常见的是矩阵的内积： $\langle A, B \rangle = \text{tr}(AB^T)$ 。

根据定义首先来看多项式 A^k 的方向导数（本章以后总假设 A 的特征分解为 $U\Lambda U^T$ ，其中 $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_d)$ 是特征值构成的对角矩阵）：

$$\begin{aligned} DA^k(A)[H] &= \lim_{h \rightarrow 0} \frac{(A + hH)^k - A^k}{h} = \sum_{l=1}^k A^{l-1} H A^{k-l} \\ &= U \left(\sum_{l=1}^k \Lambda^{l-1} [U^T H U] \Lambda^{k-l} \right) U^T \end{aligned} \quad (2-22)$$

利用公式2-20和公式2-22可得到：

$$\begin{aligned}
 Df(A)(A)[H] &= \sum_{k=0}^{\infty} \alpha_k DA^k(A)[H] \\
 &= \sum_{k=0}^{\infty} \alpha_k U \left(\sum_{l=1}^k \Lambda^{l-1} [U^T H U] \Lambda^{k-l} \right) U^T \\
 &= U \left(\sum_{k=0}^{\infty} \alpha_k \sum_{l=1}^k \Lambda^{l-1} [U^T H U] \Lambda^{k-l} \right) U^T \\
 &= UDf(\Lambda)(\Lambda)[U^T H U] U^T
 \end{aligned} \tag{2-23}$$

其中，若定义 $\tilde{H} = U^T H U$, $M \triangleq Df(\Lambda)(\Lambda)[\tilde{H}]$, 则有 (假设 $\lambda_i \neq 0$):

$$\begin{aligned}
 M_{ij} &= \sum_{k=1}^{\infty} \alpha_k \sum_{l=1}^k (\Lambda^{l-1} \tilde{H} \Lambda^{k-l})_{ij} \\
 &= \sum_{k=1}^{\infty} \alpha_k \sum_{l=1}^k \lambda_i^{l-1} \lambda_j^{k-l} \tilde{H}_{ij} \\
 &= \tilde{H}_{ij} \sum_{k=1}^{\infty} \alpha_k \frac{\lambda_j^k}{\lambda_i} \sum_{l=1}^k \left(\frac{\lambda_i}{\lambda_j} \right)^l
 \end{aligned} \tag{2-24}$$

利用公式 $\sum_{l=1}^k x^k = x \frac{1-x^k}{1-x}$, $x \neq 1$, 可以得到 (当 $\lambda_i \neq \lambda_j$ and $\lambda_i \neq 0$ 时):

$$\frac{\lambda_j^k}{\lambda_i} \sum_{l=1}^k \left(\frac{\lambda_i}{\lambda_j} \right)^l = \frac{\lambda_j^k}{\lambda_i} \frac{\lambda_i}{\lambda_j} \frac{1 - \left(\frac{\lambda_i}{\lambda_j} \right)^k}{1 - \frac{\lambda_i}{\lambda_j}} = \frac{\lambda_j^k - \lambda_i^k}{\lambda_j - \lambda_i} \tag{2-25}$$

当 $\lambda_i = \lambda_j, \lambda_i \neq 0$ 的时候有:

$$\frac{\lambda_j^k}{\lambda_i} \sum_{l=1}^k \left(\frac{\lambda_i}{\lambda_j} \right)^l = k \lambda_j^{k-1} \tag{2-26}$$

而当 $\lambda_i = 0$ 的时候, 由于 0^0 未定义, 结果不能由公式2-24的结果确认, 关于这部分的讨论放在了2.3.2中。

最后, 注意到并不一定需要 $A \in \mathbb{S}_d$, 只要 $A = U \Lambda U^{-1}$ 可对角化就可以了。这里将计算矩阵函数的方向导数 (假设方向为 H) 的步骤归纳如下:

- 对角化矩阵 A : $A = U \Lambda U^{-1}$
- 计算矩阵 H : $\tilde{H} = U^{-1} H U$
- 计算矩阵 F :

$$F_{ij} = \begin{cases} \frac{f(\lambda_i) - f(\lambda_j)}{\lambda_i - \lambda_j}, & \text{if } \lambda_i \neq \lambda_j \\ f'(\lambda_i), & \text{otherwise} \end{cases} \tag{2-27}$$

- 计算 $Df(A)(A)[H] = U(F \odot \tilde{H})U^T$

其中符号 \odot 表示矩阵的哈达玛积, 也就是矩阵的对应元素相乘。

2.3.2 矩阵包含 0 特征值的问题

前面已经介绍在，由于 0^0 未定义，所以 2.3.1 部分介绍的方法不能适用，**为此在在这一节对其进行讨论。**

$$\text{let } B = A + \mu I, \text{ then } A = \lim_{\mu \rightarrow 0} B \quad (2-28)$$

由于 A 是有限维的，那么一定存在一个 $\mu_0 > 0$ 使得 $0 < \mu < \mu_0$ 的时候 $\det(B) \neq 0$ 。并且若 $A = U\Lambda U^T$ 是 A 的 svd 分解的话， $U(\Lambda + \mu I)U^T$ 是 B 的特征分解（为方便起见记 $D = \Lambda + \mu I = \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_d)$ ），利用公式 2-23 的话可得到公式 2-29 的形式。

$$\begin{aligned} Df(B)(B)[H] &= \sum_{k=0}^{\infty} \alpha_k DB^k(B)[H] \\ &= \sum_{k=0}^{\infty} \alpha_k U \left(\sum_{l=1}^k D^{l-1}[U^T H U] D^{k-l} \right) U^T \\ &= U \left(\sum_{k=0}^{\infty} \alpha_k \sum_{l=1}^k D^{l-1}[U^T H U] D^{k-l} \right) U^T \\ &= UDf(D)(D)[U^T H U] U^T \end{aligned} \quad (2-29)$$

其中，若定义 $\tilde{H} = U^T H U, M' \triangleq Df(D)(D)[\tilde{H}]$ ，则有公式 2-30 的结。

$$\begin{aligned} M'_{ij} &= \sum_{k=1}^{\infty} \alpha_k \sum_{l=1}^k (\Lambda^{l-1} \tilde{H} \Lambda^{k-l})_{ij} \\ &= \sum_{k=1}^{\infty} \alpha_k \sum_{l=1}^k \gamma_i^{l-1} \gamma_j^{k-l} \tilde{H}_{ij} \end{aligned} \quad (2-30)$$

下面针对 λ_i, λ_j 的情况进行讨论，首先是 $\lambda_i = 0, \lambda_j \neq 0$ 的情况，此时：

$$\begin{aligned} M'_{ij} &= \sum_{k=1}^{\infty} \alpha_k \sum_{l=1}^k \mu^{l-1} \gamma_j^{k-l} \tilde{H}_{ij} \\ &= \tilde{H}_{ij} \sum_{k=1}^{\infty} \alpha_k \gamma_j^{k-1} \sum_{l=0}^{k-1} \left(\frac{\mu}{\gamma_j} \right)^l \\ &= \tilde{H}_{ij} \sum_{k=1}^{\infty} \alpha_k \gamma_j^{k-1} \frac{1 - \left(\frac{\mu}{\gamma_j} \right)^k}{1 - \frac{\mu}{\gamma_j}} \end{aligned} \quad (2-31)$$

由于: $\lim_{x \rightarrow 0} \frac{1-x^k}{1-x} = 1$ 于是有

$$\begin{aligned} M_{ij} &= \lim_{\mu \rightarrow 0} M'_{ij} = \lim_{\mu \rightarrow 0} \tilde{H}_{ij} \sum_{k=1}^{\infty} \alpha_k \gamma_j^{k-1} \\ &= \frac{1}{\lambda_j} \tilde{H}_{ij} \sum_{k=1}^{\infty} \alpha_k \lambda_j^k \\ &= \frac{\tilde{H}_{ij}}{\lambda_j} (f(\lambda_j) - f(0)) \end{aligned} \quad (2-32)$$

同理, 当 $\lambda_i \neq 0, \lambda_j = 0$ 时 $M_{ij} = \frac{\tilde{H}_{ij}}{\lambda_i} (f(\lambda_i) - f(0))$; 最后是当 $\lambda_i = \lambda_j = 0$ 的时候 $M_{ij} = \lim_{\mu \rightarrow 0} \sum_{k=1}^{\infty} \alpha_k \sum_{l=1}^k k \mu^{k-1} \tilde{H}_{ij} = \tilde{H}_{ij} f'(0)$ 。最后总结起来, 不难发现 $\lambda_i \lambda_j = 0$ 的时候也可以归结到公式2-25和2-26的形式。

2.3.3 矩阵函数的偏导数计算示例

本节的内容利用两个例子和前面2.3.1小结的结果, 对矩阵函数的导数进行计算, 首先第一个例子稍微复杂一些, 第二个例子相对简单一些, 但是在 SPD 矩阵流形的优化问题中却又着重要的意义, 同时它也是 [9] 中的例子。

A. 第一个例子

在第一个例子中我们定义矩阵函数 $f(Z)$ 如公式2-33所示:

$$f(Z) = \text{tr} \left(\left((C_1 + Z)(C_1 + Z)^T \right)^{\frac{1}{n}} \left((C_2 + Z)(C_2 + Z)^T \right)^{\frac{T}{n}} \right), C \geq 0, Z > 0, n \geq 1 \quad (2-33)$$

该函数的形式来自于本文第四章中关于 Fixed-Rank PSD 中的 Power Metric 的交叉项的讨论 (做了一些简化, 并要求 $Z > 0$ 使得优化空间为 \mathbb{S}_d^+); 于是关于 $f(Z)$ 的导数计算如下:

为了方便起见, 定义 $g_1(Z) = (C_1 + Z)(C_1 + Z)^T, g_2(Z) = (C_2 + Z)(C_2 + Z)^T$, 后利用4-32中的定律, 可得到:

$$\begin{aligned} Df(Z)(Z)[H] &= D \left((g_1(Z))^{\frac{1}{n}}, (g_2(Z))^{\frac{1}{n}} \right) (Z)[H] \\ &= \left\langle D(g_1(Z))^{\frac{1}{n}} (Z)[H], (g_2(Z))^{\frac{1}{n}} \right\rangle + \left\langle (g_1(Z))^{\frac{1}{n}}, D(g_2(Z))^{\frac{1}{n}} (Z)[H] \right\rangle \end{aligned} \quad (2-34)$$

注意到公式2-34右边的两部分中, 如果可以计算其中一部分那么另一部分也可以方便的

计算, 故接下来仅以其前一部分作为研究对象。

$$\begin{aligned}
 \left\langle D(g_1(Z))^{\frac{1}{n}}(Z)[H], (g_2(Z))^{\frac{1}{n}} \right\rangle &= \left\langle D(g_1(Z))^{\frac{1}{n}}(g_1(Z))[Dg_1(Z)(Z)[H]], (g_2(Z))^{\frac{1}{n}} \right\rangle \\
 Dg_1(Z)(Z)[H] &= D(C_1^{\frac{1}{2}} + Z)(C_1^{\frac{T}{2}} + Z^T)(Z)[H] \\
 &= D(C_1^{\frac{1}{2}} C_1^{\frac{T}{2}} + C_1^{\frac{1}{2}} Z^T + Z C_1^{\frac{T}{2}} + Z Z^T)(Z)[H] \\
 &= C_1^{\frac{1}{2}} H^T + H C_1^{\frac{T}{2}} + H Z^T + Z H^T \\
 &= (C_1^{\frac{1}{2}} + Z) H^T + H (C_1^{\frac{T}{2}} + Z^T)
 \end{aligned} \tag{2-35}$$

根据2-35以及2.3.1中的内容, 对2-35做公式2-36中的变换, 其中为了方便起见, 记 $B = Dg_1(Z)[H]$, $g_1(Z) = U_1 \Lambda_1 U_1^T$, 于是有公式2-36的计算过程。

$$\begin{aligned}
 \text{Compute : } \tilde{H} &= U_i^T B U_i \\
 \text{Compute : } \tilde{F}, \text{where } \tilde{F}_{ij} &= \begin{cases} \frac{\lambda_i^{\frac{1}{n}} - \lambda_j^{\frac{1}{n}}}{\lambda_i - \lambda_j}, & \text{if } \lambda_i \neq \lambda_j \\ \frac{1}{n} \lambda_i^{\frac{1}{n}-1}, & \lambda_i = \lambda_j \end{cases} \\
 \text{Compute : } M &= \tilde{H} \odot \tilde{F} \\
 \text{Compute : } D(g_1(Z))^{\frac{1}{n}}(Z)[B] &= U_i M U_i^T
 \end{aligned} \tag{2-36}$$

需要注意的是公式2-36中当 $n = 1$ 的时候, F 的计算比较特殊, 其结果为全 1 的矩阵; 接下来对公式2-35做进一步的化简得到:

$$\begin{aligned}
 \left\langle D(g_1(Z))^{\frac{1}{n}}(Z)[B], (g_2(Z))^{\frac{1}{n}} \right\rangle &= \left\langle U_i M U_i^T, (g_2(Z)) \right\rangle \\
 &= \left\langle \tilde{H} \odot \tilde{F}, (U_i^T (g_2(Z)) U_i) \right\rangle \\
 &= \left\langle U_i^T B U_i, (U_i^T (g_2(Z))^{\frac{1}{n}} U_i) \odot \tilde{F} \right\rangle, \text{where } \tilde{H} = U_i^T B U_i \\
 &= \left\langle B, U_i (U_i^T (g_2(Z))^{\frac{1}{n}} U_i) \odot \tilde{F} U_i^T \right\rangle \\
 &= \left\langle (C_1^{\frac{1}{n}} + Z) H^T, U_i (U_i^T (g_2(Z))^{\frac{1}{n}} U_i) \odot \tilde{F} U_i^T \right\rangle \\
 &\quad + \left\langle H (C_1^{\frac{T}{n}} + Z^T), U_i (U_i^T (g_2(Z))^{\frac{1}{n}} U_i) \odot \tilde{F} U_i^T \right\rangle \\
 &= \text{tr} \left((C_1^{\frac{1}{n}} + Z) H^T \left(U_i (U_i^T (g_2(Z))^{\frac{1}{n}} U_i) \odot \tilde{F} U_i^T \right)^T \right) \\
 &\quad + \text{tr} \left((C_1^{\frac{1}{n}} + Z) H^T U_i (U_i^T (g_2(Z))^{\frac{1}{n}} U_i) \odot \tilde{F} U_i^T \right) \\
 &= 2 \text{tr} \left(H^T \text{symm} \left(U_i (U_i^T (g_2(Z))^{\frac{1}{n}} U_i) \odot \tilde{F} U_i^T \right) (C_1^{\frac{1}{n}} + Z) \right) \\
 &= 2 \left\langle H, \text{symm} \left(U_i (U_i^T (g_2(Z))^{\frac{1}{n}} U_i) \odot \tilde{F} U_i^T \right) (C_1^{\frac{1}{n}} + Z) \right\rangle \\
 &= 2 \left\langle H, Z \text{symm} \left(D(g_1(Z))^{\frac{1}{n}}(Z)[(g_2(Z))^{\frac{1}{n}}] \right) (C_1^{\frac{1}{n}} + Z) Z \right\rangle_Z
 \end{aligned} \tag{2-37}$$

其中 $\text{symm}(X) = 0.5(X + X^T)$, $\langle \cdot, \cdot \rangle_Z$ 表示的是 \mathbb{S}_d^+ 的 Z 的切空间中的黎曼度量(内积);由此可得到 $\nabla_Z \left\langle D(g_1(Z))^{\frac{1}{n}}(Z)[B], (g_2(Z))^{\frac{1}{n}} \right\rangle = 2Z\text{symm}\left(D(g_1(Z))^{\frac{1}{n}}(Z)[(g_2(Z))^{\frac{1}{n}}]\right)(C_1^{\frac{1}{n}} + Z)Z$, 最后综合2-34~2-37的内容得到:

$$\begin{aligned} \nabla_Z \left\langle (g_1(Z))^{\frac{1}{n}}, (g_2(Z))^{\frac{1}{n}} \right\rangle &= \left\langle \nabla_Z (g_1(Z))^{\frac{1}{n}}, (g_2(Z))^{\frac{1}{n}} \right\rangle + \left\langle (g_1(Z))^{\frac{1}{n}}, \nabla_Z (g_2(Z))^{\frac{1}{n}} \right\rangle \\ &= 2Z\text{symm}\left(D(g_1(Z))^{\frac{1}{n}}(Z)[(g_2(Z))^{\frac{1}{n}}]\right)(C_1^{\frac{1}{n}} + Z)Z \\ &\quad + 2Z\text{symm}\left(D(g_2(Z))^{\frac{1}{n}}(Z)[(g_1(Z))^{\frac{1}{n}}]\right)(C_2^{\frac{1}{n}} + Z)Z \end{aligned} \quad (2-38)$$

B. 第二个例子

第二个例子在 SPD 矩阵流形上的优化问题中非常常见, 也相对于第一个例子更容易理解得多, 首先其矩阵函数的定义形式如公式2-39所示。

$$f(X|A) = \frac{1}{2} \langle \log_A(X), \log_A(X) \rangle_A, A, X \in \mathbb{S}_d^+ \quad (2-39)$$

其中 $\langle \cdot, \cdot \rangle_A$ 表示的是 SPD 矩阵流形上 A 点切空间 $T_A M$ 中的黎曼度量(内积) $\langle H_1, H_2 \rangle_A = \langle A^{-\frac{1}{2}} H_1 A^{-\frac{1}{2}}, A^{-\frac{1}{2}} H_2 A^{-\frac{1}{2}} \rangle$, 而 $\log_A(X) = A^{\frac{1}{2}} \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) A^{\frac{1}{2}}$; 所以 $f(X|A)$ 又可以写成公式2-40的形式:

$$f(X|A) = \frac{1}{2} \langle \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}), \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) \rangle \quad (2-40)$$

同样的, 这里首先计算 $f(X|A)$ 的方向导数 $Df(X|A)(X)[H]$:

$$\begin{aligned} Df(X|A)(X)[H] &= D \frac{1}{2} \langle \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}), \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) \rangle (X)[H] \\ &= \langle D \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}})(X)[H], \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) \rangle \\ &= \langle D \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}})(A^{-\frac{1}{2}} X A^{-\frac{1}{2}})[D(A^{-\frac{1}{2}} X A^{-\frac{1}{2}})(X)[H]], \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) \rangle \\ &= \langle D \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}})(A^{-\frac{1}{2}} X A^{-\frac{1}{2}})[A^{-\frac{1}{2}} H A^{-\frac{1}{2}}], \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) \rangle \end{aligned} \quad (2-41)$$

根据2.3.1部分的内容, 首先将 $A^{-\frac{1}{2}} X A^{-\frac{1}{2}}$ 对角化: $A^{-\frac{1}{2}} X A^{-\frac{1}{2}} = U \Lambda U^T$, 然后依次计算:

$$\begin{aligned} \tilde{H} &= U^T A^{-\frac{1}{2}} X A^{-\frac{1}{2}} U; F = \{F_{ij}\}_{n \times n} \\ F_{ij} &= \begin{cases} \frac{\log(\lambda_i) - \log(\lambda_j)}{\lambda_i - \lambda_j}, & \text{if } \lambda_i \neq \lambda_j \\ \frac{1}{\lambda_i}, & \text{if } \lambda_i = \lambda \end{cases} \end{aligned} \quad (2-42)$$

利用公2-42的结果可以将公式2-41的结果进一步的写成：

$$\begin{aligned}
 & \left\langle D \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}})(A^{-\frac{1}{2}} X A^{-\frac{1}{2}})[A^{-\frac{1}{2}} H A^{-\frac{1}{2}}], \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) \right\rangle \\
 &= \left\langle U(F \odot \tilde{H})U^T, \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) \right\rangle \\
 &= \left\langle F \odot \tilde{H}, U \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}})U^T \right\rangle \\
 &= \left\langle \tilde{H}, F \odot \text{diag}(\log(\lambda_1), \log(\lambda_2), \dots, \log(\lambda_d)) \right\rangle \\
 &= \left\langle U^T A^{-\frac{1}{2}} H A^{-\frac{1}{2}} U, U^T U \Lambda^{-1} U^T U \text{diag}\left(\frac{\log(\lambda_1)}{\lambda_1}, \frac{\log(\lambda_2)}{\lambda_2}, \dots, \frac{\log(\lambda_d)}{\lambda_d}\right) U^T U \right\rangle \quad (2-43) \\
 &= \left\langle U^T A^{-\frac{1}{2}} H A^{-\frac{1}{2}} U, U^T (A^{-\frac{1}{2}} X A^{-\frac{1}{2}})^{-1} \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) U \right\rangle \\
 &= \left\langle A^{-\frac{1}{2}} H A^{-\frac{1}{2}}, (A^{-\frac{1}{2}} X A^{-\frac{1}{2}})^{-1} \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) \right\rangle \\
 &= \left\langle H, X^{-1} \log(XA^{-1}) \right\rangle \\
 &= \left\langle H, \log(XA^{-1})X \right\rangle_X
 \end{aligned}$$

根据公式2-43最后两行内容可得到 $\nabla_X f(X|A)$ 分别在普通欧式空间与 X 的切空间 $T_X M$ ^① 中的结果；此外，公式2-43的推导过程运用了矩阵对称性：如果 $A \in \mathbb{S}_d$ 那么 $\langle A, B \rangle = \langle A^T, B \rangle$ 以及 $\log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) = \log(A^{-\frac{1}{2}} X A^{-1} A^{\frac{1}{2}}) = A^{-\frac{1}{2}} \log(XA^{-1}) A^{\frac{1}{2}}$ 的结果。

2.4 矩阵流形上的基本优化问题

在本章的前几节中介绍了一般优化问题和矩阵函数的导数计算问题，为本节即将介绍的矩阵流形上的优化问题做了准备；本节将基于前面几节的内容介绍矩阵流形上的优化问题，并介绍一般欧式空间中的梯度下降和共轭梯度算法在矩阵流形上的算法形式化。**为了方便理解这里还是以一个实例为研究对象进行介绍。**

在2.2.2小节中，抛出了 SPD 矩阵流形上的最小化 Fréchet Variance 的问题（由于该问题一般存在局部最小，所以最后的结果一般认为求得的是 Karcher Mean 的结果）但是并没有深入求解，所以这里以它为例介绍 SPD 矩阵流形上的优化问题。

在欧式空间中迭代算法的更新公式一般为 $x_{k+1} = x_k + \alpha_k d_k$ ，其物理意义相当于以 x_k 为起点，沿着 d_k 方向走一步，且步长为 α_k ，但是这在流形上是行不通的，因为这一步极有可能导致 x_{k+1} 跑出原来的流形空间；一般地，流形上的 $\log_A(H)$, $H \in T_A M$ 描述了在流形上的 A 处朝着切空间 $T_A M$ 的 H 方向在流形上移动，这是欧式空间中 $A + H$ 概念的泛化；而流形上的 EXP 变换正好对应于上述的物理意义；但是流形上的 EXP 操作的一个问题是计算量较大，所以为了简化计算，**另一个变换**来完成类似的操作被定义：Retraction（简写作 $R(\cdot)$ ，这里没有找到很好的中文翻译故使用英文表示）变换。

定义 2.10 (Retraction) 流形 M 上的 Retraction 变换是流形空间中的切空间束 (Tangent Bundle) TM 到流形空间本身 M 的具有如下性质的连续映射；这里令 R_X 表示切空间

^① 这个结果会在流形上优化的问题中用到，具体会在接下来的部分进行介绍

$T_X M$ 到 M 的 Retraction 变换。

- $R_X(0) = X$, 其中 0 表示的就是 $T_X M$ 中的 0 元素。
- R_X 在 0 处的微分 $(DR_X)_0 : T_0(T_X M) \equiv T_X M \rightarrow T_X M$ 是 $T_X M$ 中的恒等变换: $(DR_X)_0 = Id$ (局部刚性的)。

定义2.10给出了流形上 Retraction 变换的定义; 特别地, 流形上的 EXP 变换也是一个 Retraction 变换。在 SPD 矩阵流形上 $\exp_X(\cdot)$ 是最常用的 Retraction 变换。有了 Retraction 变换, 那么将欧式空间中的梯度下降算法泛化到流形空间也相对直接。这里类似于 [9] 中的做法, 这里先总结2.2.2小节中的梯度下降算法到算法1中。

Algorithm 1 梯度下降算法

Require: 目标函数 $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, 目标函数的梯度 $\nabla_x f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$, 初始值 x_0 。

Ensure: 算法的搜索系列 x_0, x_1, \dots 以及最后停止迭代的点 x^*

- 1: 初始化迭代次数 $k \leftarrow 0$
 - 2: **while** not converge **do**
 - 3: 计算负梯度方向: $d_k = -\nabla_x f(x_k)$
 - 4: 计算迭代步长: $\alpha_k = \min_\alpha f(x_k + \alpha_k d_k)$
 - 5: 更新结果: $x_{k+1} = x_k + \alpha_k d_k$
 - 6: 更新迭代次数: $k \leftarrow k + 1$
 - 7: **end while**
 - 8: **return** $\{x_i\}_{i=0}^k, x^* = x_k$
-

算法1中的收敛条件不尽相同, 最理想的情况是 $\|\nabla_x f(x_k)\| = 0$, 但是一般很难达到, 所以一般是要求 $\|\nabla_x f(x_k)\| < \epsilon$, 其中 ϵ 是很小的数 (如: $10^{-6}, 10^{-8}$ 等), 还有一类条件是要求 $k < maxiter$, 其中 $maxiter$ 是预先设置的最大迭代次数。

前面已经介绍 Retraction 变换是欧式空间中 $X + H$ 在流形空间中的泛化, 利用它代替1中的第五步可以得到流形空间中的梯度下降算法2。

算法2的收敛条件的设置与算法1类似, 这里不再赘述, 此外由于前面已经介绍 EXP 也是 Retraction 变换的一种; 为了方便理解, 下面以 SPD 矩阵流形的 Karcher Mean 的计算为例对算法2进行进一步的探讨。

在2.2.2一节中介绍了 SPD 矩阵流形上 $f(X|A) = \frac{1}{2}\text{dist}^2(A, X)$ 关于 X 的导数计算: $\nabla_X f(X|A) = \log(XA^{-1})X \in T_X M$, 此外, 注意到 $f(X|A) = \frac{1}{2}\text{dist}^2(A, X) = \frac{1}{2}\text{dist}^2(A, X) = f(A|X)$ 可以得到:

$$\nabla_A f(X|A) = \nabla_A f(A|X) = \log(AX^{-1})A = A^{\frac{1}{2}} \log(A^{\frac{1}{2}} X^{-1} A^{\frac{1}{2}})A^{\frac{1}{2}} \quad (2-44)$$

利用公式2-44的结果, SPD 矩阵流形空间中 Fréchet Var: $C(\mu) = \frac{1}{n} \sum_{i=1}^n \text{dist}^2(X_i, \mu)$ 的导

Algorithm 2 流形空间梯度下降算法

Require: 目标函数 $f(X) : M \rightarrow \mathbb{R}$, 目标函数的梯度 $\nabla_X f(X) : M \rightarrow T_X M$, 初始值 $X_0 \in M$ 。

Ensure: 算法的搜索系列 X_0, X_1, \dots 以及最后停止迭代的点 X^*

- 1: 初始化迭代次数 $k \leftarrow 0$
- 2: **while** not converge **do**
- 3: 计算负梯度方向: $H_k = -\nabla_{x_k} f(X_k) \in T_{X_k} M$
- 4: 计算迭代步长: $\alpha_k = \min_\alpha f(R_{x_k}(\alpha_k H_k))$
- 5: 更新结果: $X_{k+1} = R_{X_k}(\alpha_k H_k)$
- 6: 更新迭代次数: $k \leftarrow k + 1$
- 7: **end while**
- 8: **return** $\{x_i\}_{i=0}^k, X^* = X_k$

数如公式2-45所示：

$$\begin{aligned}
 \nabla_\mu C(\mu) &= \frac{1}{n} \sum_{i=1}^n \nabla_\mu \text{dist}^2(X_i, \mu) \\
 &= \frac{2}{n} \sum_{i=1}^n \mu^{\frac{1}{2}} \log(\mu^{\frac{1}{2}} X^{-1} \mu^{\frac{1}{2}}) \mu^{\frac{1}{2}} \\
 &= -\frac{2}{n} \sum_{i=1}^n \mu^{\frac{1}{2}} \log(\mu^{-\frac{1}{2}} X \mu^{-\frac{1}{2}}) \mu^{\frac{1}{2}} \\
 &= -\frac{2}{n} \sum_{i=1}^n \log_\mu(X_i)
 \end{aligned} \tag{2-45}$$

最后将上述结果带入到算法2中，可以得到 Karcher Mean 的更新公式：

$$\mu_{k+1} = \exp_{\mu_k} \left(\frac{\alpha_k}{n} \sum_{i=1}^n \log_\mu(X_i) \right) = \mu_k^{\frac{1}{2}} \exp \left(\frac{\alpha_k}{n} \sum_{i=1}^n \log(\mu_k^{-\frac{1}{2}} X_i \mu_k^{-\frac{1}{2}}) \right) \mu_k^{\frac{1}{2}} \tag{2-46}$$

接下来将对另一个常用的算法——共轭梯度算法在矩阵流形上的泛化做介绍。类似地，这里首先将2.2.2中介绍的欧式空间中的共轭梯度算法归纳到算法3中。

算法3中描述的方法并不能保证算法是收敛的这与 α_k, β_k 的选择密切相关，但是关于收敛性的证明不是本文讨论的重点，感兴趣的读者可以参考 [3] 及其它相关的文章；不过一般认为的是梯度算法的收敛（若收敛的话）速度比梯度下降要快得多。

将共轭梯度算法泛化到矩阵流形空间中需要解决的问题有两个：1) $x_{k+1} = x_k + \alpha_k d_k$ 中的加法问题，这个已经有 Retraction 变换解决；2) $d_{k+1} = -g_{k+1} + \beta_k d_k$ 中的加法问题，这里的主要问题是流形上 $d_{k+1}, g_{k+1} \in T_{x_{k+1}} M, d_k \in T_{x_k} M$ 是不同切空间中的向量，加法未定义，所以这里引入流形上的另一个概念 vector transport^① 并用 T 表示。

^① vector transport 是流形上 parallel translation 的近似，关于 parallel translation 感兴趣的读者可以参看 [3,9] 以及文章 [35] 的补充材料，这里不再展开

Algorithm 3 流形上的共轭梯度算法

Require: 目标函数 $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$, 目标函数的梯度 $\nabla_{\mathbf{x}} f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$, 初始值 \mathbf{x}_0 。

Ensure: 算法的搜索系列 $\mathbf{x}_0, \mathbf{x}_1, \dots$ 以及最后停止迭代的点 \mathbf{x}^*

- 1: 初始化迭代次数 $k \leftarrow 0$, 初始化迭代方向 $\mathbf{d}_0 = -\nabla_{\mathbf{x}} f(\mathbf{x}_0)$
- 2: **while** not converge **do**
- 3: 计算迭代步长: $\alpha_k = \min_{\alpha} f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)$
- 4: 更新结果: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
- 5: 计算梯度方向: $\mathbf{g}_{k+1} = \nabla_{\mathbf{x}} f(\mathbf{x}_{k+1})$
- 6: 计算参数 β_k : $\beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k}$
- 7: 更新搜索方向: $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k$
- 8: 更新迭代次数: $k \leftarrow k + 1$
- 9: **if** $k \bmod n = 0$ **and** not converge **then**
- 10: 重启共轭梯度算法
- 11: **end if**
- 12: **end while**
- 13: **return** $\{\mathbf{x}_i\}_{i=0}^k, \mathbf{x}^* = \mathbf{x}_k$

定义 2.11 (Vector Transport) Vector Transport 是流形 M 的切空间束 TM 上的光滑变换:

$$TM \oplus TM \rightarrow TM : (\xi, \eta) \rightarrow T_{\eta}(\xi) \in TM$$

并且满足如下的几条性质:

- (Retraction 关联) 如果一个 Retraction (记为 R) 满足 $T_{\eta}(\xi) \in T_{R_x(\eta)} M$, 则称 R 与 T 关联
- (一致性) $T_0(\xi) = \xi, \forall \xi \in T_x M$
- (线性性) $T_{\eta}(a\xi + b\zeta) = aT_{\eta}(\xi) + bT_{\eta}(\zeta); a, b \in \mathbb{R}$

利用 Retraction 以及 Vector Transport 变换以及算法3, 这里将流形空间中的共轭梯度算法归纳到算法4中: 关于流形上的共轭梯度算法更多实现相关的细节可以参看 Manopt[10] 的实现。

2.5 总结

本章的主要内容与工作 [9] 和 [3] 相关, 探讨了矩阵函数与黎曼流形上的优化问题, 并针对一些特殊的情况探讨和分析, 结合着学位论文课题中提炼出的相关实例进行介绍, 一方面帮助读者理解并复现接下来本文所提出的方法, 另一方面也为解决类似流形优化问题提供借鉴。

Algorithm 4 流形上的共轭梯度算法

Require: 目标函数 $f(X) : M \rightarrow \mathbb{R}$, 目标函数的梯度 $\nabla_X f(X) : M \rightarrow \mathbb{R}^n$, 初始值 $X_0 \in M$.

Ensure: 算法的搜索系列 X_0, X_1, \dots 以及最后停止迭代的点 X^*

- 1: 初始化迭代次数 $k \leftarrow 0$, 初始化迭代方向 $H_0 = -\nabla_X f(X_0) \in T_{X_0} M$
 - 2: **while** not converge **do**
 - 3: 计算迭代步长: $\alpha_k = \min_\alpha f(R_{X_k}(\alpha_k H_k))$
 - 4: 更新结果: $X_{k+1} = R_{X_k}(\alpha_k H_k)$
 - 5: 计算梯度方向: $G_{k+1} = \nabla_X f(X_{k+1}) \in T_{X_{k+1}} M$
 - 6: 计算参数 β_k : $\beta_k = \frac{\langle G_{k+1}, G_{k+1} \rangle_{G_{k+1}}}{\langle G_k, G_k \rangle_{G_k}}$
 - 7: 更新搜索方向: $H_{k+1} = -G_{k+1} + \beta_k T_{\alpha_k G_k}(H_k)$
 - 8: 更新迭代次数: $k \leftarrow k + 1$
 - 9: **if** $k \bmod n = 0$ **and** not converge **then**
 - 10: 重启共轭梯度算法
 - 11: **end if**
 - 12: **end while**
 - 13: **return** $\{X_i\}_{i=0}^k, X^* = X_k$
-

第三章 黎曼流形上的 PLS 回归

在 1.3 中已经提到的，统计建模图像集合的方法在图像集合分类问题中的优异表现使得该方法逐渐成为研究图像集合的分类问题的主流方法之一；而在使用统计模型建模图像集合的时候往往涉及到一种特殊的数据结构——对称正定矩阵（Symmetric Positive Definite Matrices）：在单统计量建模图像集合的工作中（如 [57], [22]）均使用样本协方差（Covariance）建模图像集合；在多统计模型建模图像集合的工作中（如 [42], [29]）使用的二阶统计量以及高斯分布等都与对称正定矩阵相关（根据信息集合的内容 [4]，高斯分布在适当的 metric 定义下构成黎曼流形，且该流形空间与 SPD 矩阵流形空间关系十分密切，详细的内容读者可以阅读文献 [4] 做进一步的了解）；在分布函数建模图像集合的工作 [59] 正是利用了 GMM（混合高斯模型）中的高斯分布与 SPD 矩阵流形的关系对图像集合建模的。由此可见 SPD 矩阵流形的研究，甚至是黎曼流形的研究对于图像集合统计建模的重要性。

另一方面，关于 SPD 流形的研究更早于图像集合的问题而被提出，在计算机视觉领域较早且影响深远的是医学上的 DTI（Diffusion Tensor Image）的研究（如工作 [46], [?], [14], [35] 等），它们为后续的用 SPD 矩阵流形研究图像集合奠定了基础；同时工作 [14] 和 [35] 则更进一步的将欧式空间中的两个有效数据分析方法：PCA（Principle Component Analysis/主成分分析）和 CCA（Canonical Correlation Analysis/典型相关分析）扩展到了黎曼流形上（自然也包括 SPD 矩阵流形）；这也启示我们将与 PCA 和 CCA 关系十分密切的 PLS（Partial Least Square Regression/偏最小二乘回归）扩展到黎曼流形上，并针对图像集合问题的特点（相较于 DTI 图像，图像集合问题中的样本数要少很多但是维度却要高很多）对其进行改进，使其适配到图像集合分类问题上。

为此，我们将接下来的内容安排如下：首先花一些篇幅介绍一下 PLS（Partial Least Square Regression/偏最小二乘回归），然后是黎曼流形空间中的投影的概念，接着是流形空间中计算投影的数学形式，然后借助投影和子流形的概念定义黎曼流形上的 PLS 问题，紧接着是针对图像集合问题的黎曼流形上的 PLS 方法的问题适配和改进，最后给出实验验证和未来可能的方向讨论。

3.1 偏最小二乘方法

这一节将对偏最小二乘方法做介绍，其中还会介绍 NIPALS(Nonlinear Iterative Partial Least Squares algorithm)[31] 算法，它至今仍然是求解偏最小二乘的有力工具，此外还会对 PLS 用于分类问题的场景做简要介绍，为后续的分类问题做准备，在本节的最后会讨论 CCA(Canonical Correlation Analysis) 与 PLS 的关系。

偏最小二乘法 (partial least squares method) 于 1983 年由伍德 (S.Wold) 和阿巴诺 (C.Albano) 等人首次提出，并在之后的几十年间得到了长足的发展，文献 [6,25,48,50,61] 就是其中具有代表性的工作。PLS 是对多元线性回归模型的一种扩展（也被称为第二代回归方法），偏最小二乘方法在建模的过程中集中了主成分分析，典型相关分析以及回归分析的特点，因而相较于其它的多元数据分析方法可以给出更加合理的模型，特别地，PLS 可以比较好的处理回归分析中自变量多重共线性的问题，且当解释变量的数量超过观测变量的时候或者两者之间存在严重的共线性的关系的时候 PLS 仍然可以对数据进行很好的建模，下面对 PLS 的数学形式进行简要的描述。

假设两个数据集合 $X \subset \mathbb{R}^N$ 和 $Y \subset \mathbb{R}^M$ ，并且假设有分别来自两个样本集合的 n 个样本，记为： $X \in \mathbb{R}^{n \times N}$, $Y \in \mathbb{R}^{n \times M}$ (这里不失一般性的假设数据集 X, Y 都是 0 均值的)，PLS 通过得分向量来关联两个数据集合：

$$\begin{aligned} X &= TP^T + E \\ Y &= UQ^T + F \end{aligned} \quad (3-1)$$

其中 T, U 是 $n \times p$ 的矩阵，也就是 p 个得分向量所构成的矩阵， $N \times p$ 矩阵 P 和 $M \times p$ 矩阵 Q 称为载荷矩阵，末尾的 $n \times N$ 矩阵 E 和 $n \times M$ 矩阵 F 表示的是残差矩阵。

上述模型中，最主要的问题是得分向量以及载荷矩阵的计算，在众多的求解算法中 NIPALS(Nonlinear Iterative Partial Least Squares algorithm)[31] 是最具代表性的算法之一，至今仍然是 PLS 中最常用和核心的算法：该算法通过寻找向量 w, c 使得： $[cov(Xw, Yc)]^2$ 最大化，即：

$$[cov(u, t)]^2 = [cov(Xw, Yc)]^2 = \max_{\|r\|=\|s\|=1} [cov(Xr, Ys)]^2 \quad (3-2)$$

NIPALS 的计算过程如公式3-3所示：首先是随机初始化向量 u ，然后依次执行如下的步骤直到收敛为止：

$$\begin{aligned} w &= X^T u / (u^T u) \quad c = Y^T t / (t^T t) \\ \|w\| &\rightarrow 1 \quad \|c\| \rightarrow 1 \\ t &= Xw \quad u = Yc \end{aligned} \quad (3-3)$$

对于 NIPALS 算法这里还剩下两个问题：1) 如何计算多个得分向量；2) 当用于分类问题时 PLS 的标签数据集是怎样表示的。这两个问题在 [48] 中有具体细致的介绍，这里就不再赘述了，只简单的给出结果方便后续引用：对于第 1 个问题，通常的做法是在每次计算新的得分向量的时候我们都会减去前一得分向量的影响，如公式3-4所示。

$$\begin{aligned} p &= X^T t / (t^T t) \\ X &= X - tp^T \\ Y &= Y - tt^T Y / (t^T t) = Y - tc^T \end{aligned} \quad (3-4)$$

对于第二个问题，若假设分类问题有 C 个类，并且 $y_i \in \{1, 2, \dots, C\}$, $i = 1, 2, \dots, n$ 表示类别标签，则对于每一个 y_i 可以将其映射到一个 C 维向量 $y_i \rightarrow p^{y_i}$ 使得：

$$p_k^{y_i} = \begin{cases} 1 & \text{if } k = y_i \\ 0 & \text{else} \end{cases}$$

因此，若标签是按照类别排序的，那么原本的标签向量就转换为如下的标签矩阵 Y ：

$$Y = \begin{bmatrix} p^{y_1} \\ p^{y_2} \\ \vdots \\ p^{y_n} \end{bmatrix} = \begin{bmatrix} 1_{n_1} & 0_{n_2} & \cdots & 0_{n_C} \\ 0_{n_1} & 1_{n_2} & \cdots & 0_{n_C} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{n_1} & 0_{n_2} & \cdots & 1_{n_C} \end{bmatrix} \quad (3-5)$$

其中 n_1, n_2, \dots, n_C 表示的是各个类别的样本数，且有 $n = \sum_{i=1}^C n_i$ 。

最后，这里简单介绍 PLS 与 CCA 的关系，更多的关于 PLS, PCA, CCA 之间的关系可以从 [48] 中获得；CCA(Canonical Correlation Analysis) 与 PLS 作为多元统计分析中的有力工具，广泛应用于统计分析，机器学习，计算机视觉等领域；两者虽然名字差别很大但是形式上两者是很相近的，以至于在一些情况下可以说两者就是等价的；他们的相似性从它们的数学形式就可以很好的看出来：

$$\begin{aligned} CCA &: \max_{\|r\|=\|s\|=1} [\text{corr}(Xr, Ys)]^2 \\ PLS &: \max_{\|r\|=\|s\|=1} [\text{cov}(Xr, Ys)]^2 \end{aligned} \quad (3-6)$$

早在 1976 年 H. D. Vinod 在他关于 canonical ridge analysis 的论文 [55] 给出的公式即可看出一二；在 H.D.Vinod 的论文中考虑如下的优化问题：

$$\max_{\|r\|=\|s\|=1} \frac{\text{cov}(Xr, Ys)}{([1 - \gamma_X] \text{var}(Xr) + \gamma_X)([1 - \gamma_Y] \text{var}(Ys) + \gamma_Y)} \quad (3-7)$$

其中 $0 \leq \gamma_X, \gamma_Y \leq 1$ ，进一步的上述问题的解对应于如下特征值问题：

$$([1 - \gamma_X] X^T X + \gamma_X I)^{-1} X^T Y ([1 - \gamma_Y] Y^T Y + \gamma_Y I)^{-1} Y^T X w = \lambda w \quad (3-8)$$

根据上述内容在这里总结一下：CCA 与 PLS 与在数学形式上很相似，其中 PLS 最大化的是投影后的向量间的协方差，而 CCA 最大化的是投影的后的相关系数，两者相差了一个尺度因子，在尺度因子（投影后的标准差）为 1 的时候两者相等；此外两者本质上都是特征值问题，且特征值问题的一般形式如式 3-8 所示。

3.2 黎曼流形上的投影问题

简单回顾欧式空间中的 PCA, CCA 和 PLS 不难发现的是其中都涉及到投影（如投影向量或投影矩阵）的概念；这也就不难理解本文在这一节介绍黎曼流形上的投影的概

念的意图。这一节的内容主要分为两部分：第一部分从欧式空间中的投影开始，逐步介绍抽象的投影的概念，这个概念很容易推广到黎曼流形空间；第二部分会具体的以 SPD 矩阵流形为例，将黎曼流形的投影的概念具体化。

3.2.1 一般化的投影

这里的投影理解为欧式空间中高维空间向低维空间的投影的一般化，此外由于 PCA, CCA 以及 PLS 中的概念是相似的所以这里就直接以 PLS 为载体进行欧式空间中的投影的概念的解释和一般化介绍。

在3.1节我们知道了欧式空间中的偏最小二乘的目标函数的形式是公式3-2，也就是最大化协方差的目标：

$$\begin{aligned} \max_{\|\mathbf{r}\|=\|\mathbf{s}\|=1} [\text{cov}(\mathbf{X}\mathbf{r}, \mathbf{Y}\mathbf{s})]^2 &= \max_{\|\mathbf{r}\|=\|\mathbf{s}\|=1} \left\{ \sum_{i=1}^n (\mathbf{x}_i^T \mathbf{r} - \bar{\mathbf{x}}^T \mathbf{r})(\mathbf{y}_i^T \mathbf{s} - \bar{\mathbf{y}}^T \mathbf{s}) \right\}^2 \\ &= \max_{\|\mathbf{r}\|=\|\mathbf{s}\|=1} \left\{ \sum_{i=1}^n [\mathbf{r}^T (\mathbf{x}_i - \bar{\mathbf{x}})][\mathbf{s}^T (\mathbf{y}_i - \bar{\mathbf{y}})] \right\}^2 \end{aligned} \quad (3-9)$$

在公式3-9中 $\mathbf{r}^T (\mathbf{x}_i - \bar{\mathbf{x}})$ 和 $\mathbf{s}^T (\mathbf{y}_i - \bar{\mathbf{y}})$, $i = 1, 2, \dots, n$ 即为高维空间向低维空间的投影变换。但是这样的概念并不能很好的泛化到流形空间（因为流形空间并没有内积的定义，但是注意到流形空间有测地距离的定义），所以这里先来回顾一下欧式空间中投影的更一般的形式（其物理意义）：

设 $\mathbf{x} \in \mathbb{R}^n$ 是 n 维空间中的向量，又有 S_k 表示 n 维空间中的一个 k 维子空间，那么 \mathbf{x} 向 S_k 中的投影可以由如下的优化问题得到（其中 $d(\cdot, \cdot)$ 表示的是距离函数）：

$$\Pi_{S_k}(\mathbf{x}) = \min_{\mathbf{x}' \in S_k} d^2(\mathbf{x}', \mathbf{x}) \quad (3-10)$$

特别地，若 S_k 是一维子空间，即 S_k 由一个向量张成的子空间 $S_1 = \{\mathbf{v} | \mathbf{v} = t\mathbf{w}, t \in \mathbb{R} \text{ and } \mathbf{w} \in \mathbb{R}^n\}$ 则有：

$$\begin{aligned} \Pi_{S_1}(\mathbf{x}) &= \min_{\mathbf{x}' \in S_1} d^2(\mathbf{x}', \mathbf{x}) \\ t^* \triangleq \pi_{S_1}(\mathbf{x}) &= \min_{t \in \mathbb{R}} d^2(t\mathbf{w}, \mathbf{x}) \end{aligned} \quad (3-11)$$

这里的 t 称为投影系数，也就是3-9中的 $\mathbf{r}^T (\mathbf{x}_i - \bar{\mathbf{x}})$ 或 $\mathbf{s}^T (\mathbf{y}_i - \bar{\mathbf{y}})$, $i = 1, 2, \dots, N$ ，因而3-9的问题在这个投影描述下变成了寻找合适的一维子空间 $S_1^x = \{\mathbf{v} | \mathbf{v} = t\mathbf{r}, t \in \mathbb{R} \text{ and } \mathbf{r} \in \mathbb{R}^n, \|\mathbf{r}\| = 1\}$, $S_1^y = \{\mathbf{v} | \mathbf{v} = u\mathbf{s}, u \in \mathbb{R} \text{ and } \mathbf{s} \in \mathbb{R}^n, \|\mathbf{s}\| = 1\}$ 使得投影后的数据具有最大的协方差。

3.2.2 SPD 矩阵流形上的均值

本节将就 SPD 矩阵流形为例，介绍其上的投影变换；此外总结欧式空间中的 PLS，不难发现，若要将 PLS 推广到黎曼流形空间需要完成四件事：1) 数据的中心化；2) 寻找合适的子流形，3) 将高维空间中的数据投影到子流形中；4) 最大的协方差；本节的

内容主要完成的是前面的两件事：数据的中心化和寻找合适的子流形。由于子流形的构造与中心化相关，所以本节先花一点篇幅介绍 SPD 上的 Karcher mean 问题：

已有相当一部分文章对黎曼流形上的均值的形式进行了研究；特别地，针对对称正定矩阵流形，文献 [35,46?] 中都对其上的均值概念在特定的度量下进行了介绍和推导，所以这里我们直接给出它计算的数学形式：

$$\bar{x} = \min_x \frac{1}{2n} \sum_{i=1}^n \delta^2(x, x_i)$$

这里的 $\delta(\cdot, \cdot)$ 表示的是流形空间中的测地距离，在 PSD 流形中常使用也是最基本的就是 AIM(Affine Invariant Metric)，将 Affine Invariant Metric 带入到上式的话可以得到：

$$\begin{aligned} \mu &= \min_X f(X) = \min_X \frac{1}{2n} \sum_{i=1}^n \delta^2(X, X_i) \\ &= \min_X \frac{1}{2n} \sum_{i=1}^n \|\log(X^{-\frac{1}{2}} X_i X^{-\frac{1}{2}})\|_F \end{aligned} \quad (3-12)$$

其中，由于使用 μ 来表示样本均值已经是约定俗成的了所以尽管这里的 Karcher mean 是一个矩阵，但是我们仍然用 μ 来表示。注意到与欧式空间不同的是，要想求出公式 3-12 的解析解几乎是不可能的，但是作为优化问题的话，优化算法仍然是可用的，因此主要的问题就是计算公式 3-12 的梯度函数，为了简化公式这里先来计算 Affine Invariant Metric 的梯度（具体的内容可参看本文第四章）：

$$\begin{aligned} \text{dist}(X, X_i) &= \|\log(X^{-\frac{1}{2}} X_i X^{-\frac{1}{2}})\|_F, i = 1, 2, \dots, n \\ \nabla_X \text{dist}(X, X_i) &= 2X^{\frac{1}{2}} \log(X^{\frac{1}{2}} X_i^{-1} X^{\frac{1}{2}}) X^{\frac{1}{2}} = -2 \log_X(X_i) \end{aligned} \quad (3-13)$$

将上式带入到 3-12 可得到：

$$\begin{aligned} \nabla f(X) &= \frac{1}{2n} \sum_{i=1}^n 2X^{\frac{1}{2}} \log(X^{\frac{1}{2}} X_i^{-1} X^{\frac{1}{2}}) X^{\frac{1}{2}} \\ &= -\frac{1}{n} \sum_{i=1}^n \log_X(X_i) \end{aligned} \quad (3-14)$$

利用 3-14 不难得到黎曼流形上的梯度下降更新公式（其中 k 是迭代次数, τ 代表步长）。

$$\mu_{k+1} = \exp_{\mu_k} \left(\frac{\tau}{n} \sum_{i=1}^n \log_{\mu_k}(X_i) \right) \quad (3-15)$$

至此基本介绍了 SPD 矩阵流形空间中的 Intrinsic Mean 的概念和形式化。

3.2.3 黎曼流形上的子流形空间投影

在前面的 3.2.3 节已经介绍了黎曼流形上的均值的计算，并且以 SPD 流形上的 AIM 度量为例给出了具体的计算公式，这节将会介绍黎曼流形上高维流形空间向低维流形空

间的投影的过程，此前在文献 [14,35] 中对该问题就有一些介绍，这里会对之前工作做一个总结，并对其进行一些修改，为黎曼流形上的 PLS 推广做准备。

首先，根据欧式空间中的高维空间向一维子空间的投影形式（公式3-11），可以看出确定一个一维的子流形空间是投影变换的首要任务，并注意到流形上的测地线是链接两点之间最短的曲线，它是欧式空间中两点之间直线最短的概念的推广，因此利用测地线构造子流形空间是自然的一种途径，而从流行上的均值（Karcher mean）出发的测地线常用于构造这样的子流形空间（目前还没有理论证明这样的构造是最优的，但是一部分在 PSD 流形上的实验结果验证了在均值这点可以得到不错的结果 [53]），于是将从 Karcher mean 出发利用测地线构造子空间记为 S_W ，带入到公式3-11中并使用测第距离可得到公式3-16的形式。

$$\pi_{S_W}(X) = \min_{X' \in S_W} \delta^2(X', X) \quad (3-16)$$

更具体地，当把上述理论运用到对称正定矩阵流形空间的时候（使用 Affine Invariant Metric），可以更具体的得到公式3-17的形式。

$$\left\{ \begin{array}{l} S_W = \exp_\mu(\text{span}(w) \cap U) \\ \Pi_{S_W}(X) = \min_{X' \in S_W} \delta^2(X', X) \\ = \min_{t \in (-\epsilon, \epsilon), X' = tw} \|\log_{[\exp_\mu(tw)]}(X)\|^2 \\ \pi_{S_W}(X) = \min_{t \in (-\epsilon, \epsilon)} \|\log_{[\exp_\mu(tw)]}(X)\|^2 = t^* \end{array} \right. \quad (3-17)$$

其中 w 就是 Karcher mean 处出发的一条切向量，类似于欧式空间中 PLS 数据中心化后的空间中的投影方向； $\log()$, $\exp(\cdot)$ 的定义参看2-4，最后的公式3-17即包含了子流形的构造以及原始空间向子流形空间的投影。

3.3 黎曼流形上的 PLS 回归问题

本节将借助前面定义的子流形和投影的概念对 SPD 矩阵流形的一般形式化进行阐述，然后在接下来的一小节中，将结合这种一般化形式针对图像集合分类问题的特点，对其进行进一步的改进，使其更加适配到图像集合分类问题上。

3.3.1 黎曼流形上 PLS 回归问题的一般形式

首先是黎曼流形上的偏最小二乘问题：假设 $\{X_i\}_{i=1}^n, \{Y_i\}_{i=1}^n \subset M$ 是来自对称正定矩阵流形空间 M 的两组样本，根据公式3-16以及公式3-9这里可以写出如下的目标函数：

$$\begin{aligned} \max_{w_x, w_y} C^2(w_x, w_y) &= \max_{w_x, w_y} \left(\sum_{i=1}^n (t_i - \bar{t})(u_i - \bar{u}) \right)^2 \\ \text{s.t. } t_i &= \min_{X(t) \in S_{w_x}, t \in (-\epsilon, \epsilon)} \delta^2(X(t), X_i); i = 1, 2, \dots, n; \\ u_i &= \min_{Y(u) \in S_{w_y}, u \in (-\eta, \eta)} \delta^2(Y(u), Y_i); i = 1, 2, \dots, n; \\ \|w_x\| &= \|w_y\| = 1; \bar{t} = \frac{1}{n} \sum_{i=1}^n t_i; \bar{u} = \frac{1}{n} \sum_{i=1}^n u_i. \end{aligned} \quad (3-18)$$

将上述的形式化具体到 SPD 矩阵流形：假设 $\{X_i\}_{i=1}^n, \{Y_i\}_{i=1}^n \subset \mathbb{S}_d^+$ 是来自对称正定矩阵流形空间 \mathbb{S}_d^+ 的两组样本，根据公式3-17以及公式3-9这里可以写出如下的目标函数：

$$\begin{aligned} \max_{W_X, W_Y} C^2(W_X, W_Y) &= \max_{W_X, W_Y} \left(\sum_{i=1}^n (t_i - \bar{t})(u_i - \bar{u}) \right)^2 \\ \text{s.t. } t_i &= \min_{t \in (-\epsilon, \epsilon)} \|\log_{[\exp_{\mu_X}(tW_X)]}(X_i)\|^2; i = 1, 2, \dots, n; \\ u_i &= \min_{u \in (-\eta, \eta)} \|\log_{[\exp_{\mu_Y}(uW_Y)]}(Y_i)\|^2; i = 1, 2, \dots, n; \\ \|W_X\| &= \|W_Y\| = 1; \bar{t} = \frac{1}{n} \sum_{i=1}^n t_i; \bar{u} = \frac{1}{n} \sum_{i=1}^n u_i. \end{aligned} \quad (3-19)$$

公式3-19描述了 SPD 矩阵流形上的偏最小二乘问题的一般形式，也是定义了流形空间中的投影之后所能得到的最直接的形式化；但是不难发现的是还有如下一些问题遗留：1) 由于公式3-19没有解析解，在工作 [35] 中已经出现了计算量大，消耗时间过长的问题，究其原因主要是 $t_i = \min_{t \in (-\epsilon, \epsilon)} \|\log_{[\exp_{\mu_X}(tW_X)]}(X_i)\|^2; i = 1, 2, \dots, n;$ 和 $u_i = \min_{u \in (-\eta, \eta)} \|\log_{[\exp_{\mu_Y}(uW_Y)]}(Y_i)\|^2; i = 1, 2, \dots, n;$ 的计算，所以计算复杂度高是上式得一个问题是，2) 如何像欧式空间中的 PLS 一般计算第二个（或更多个）投影方向（子空间）；3) 如何将上述问题用于分类问题（或者说如何做回归问题）。这些问题中最主要的问题就是原问题没有解析解（也就是问题1）的问题，下面会看到通过适当近似手段将原问题简化将使得上面列出的三个问题迎刃而解。

前面已经提到形式3-19的时间复杂度过高，并且该问题在图像集合分类问题上会更加严重（原因是数据的维度大大增加了），所以这里选择适当的近似以求在保证一定的精度的同时速度能有较大的提升。

这里的近似的方法在工作 [14,35] 中已有类似的介绍，主要的思想就是针对 $t_i = \min_{t \in (-\epsilon, \epsilon)} \|\log_{[\exp_{\mu_X}(tW_X)]}(X_i)\|^2; i = 1, 2, \dots, n;$ 和 $u_i = \min_{u \in (-\eta, \eta)} \|\log_{[\exp_{\mu_Y}(uW_Y)]}(Y_i)\|^2; i = 1, 2, \dots, n;$ 的计算复杂过高（或者说没有解析解）的问题而提出的简化方案。

以 $t_i = \min_{t \in (-\epsilon, \epsilon)} \|\log_{[\exp_{\mu_x}(tW_X)]}(X_i)\|^2$ 为例, 这里我们不难发现时间复杂度主要来自于 $\|\log_{[\exp_{\mu_x}(tW_X)]}(X_i)\|^2$ 而这一部分实际上描述了这样一个过程: 将 Karcher mean (μ_x) 的切空间中的向量 tW_X 通过 \exp_{μ_x} 变换到 \mathbb{S}_d^+ 中然后在 \mathbb{S}_d^+ 中度量两者 $(\exp_{\mu_x}(tW_X), X_i)$ 的测地距离 $\delta(\exp_{\mu_x}(tW_X), X_i)$; 据此我们使用如下的近似方案 (具体的参考 [14]), 将在原流形空间中度量 $\exp_{\mu_x}(tW_X), X_i$ 两者的距离改用在 μ_x 的切空间中度量两者之间的距离 $\text{dist}(\exp_{\mu_x}(tW_X), X_i) \approx \|tW_X - \log_{\mu_x}(X_i)\|_{\mu_x}$, 于是原来的投影问题3-17就变成了:

$$\Pi_S(X) = \exp\left(\mu_x, \sum_{i=1}^k V_i \langle V_i, \log(\mu_x, X) \rangle\right) \quad (3-20)$$

其中 $V_{i=1}^d$ 是 μ_x 处的切空间中的标准正交基 (由于 μ_x 的切空间是内积空间), 这点修改使得问题大大简化, 并使用算法5进行描述。

Algorithm 5 对称正定矩阵流形上的偏最小二乘近似算法

Require: 对称正定矩阵集合 $X = \{X_i\}_{i=1}^n, Y = \{Y_i\}_{i=1}^n$, 需要计算的成分的个数 k

Ensure: 在集合 X 的 Karcher mean (μ_x) 处的切空间处张成子空间的 k 个成分 $W_X = \{W_X^i\}_{i=1}^k$, 以及 $\{X_i\}_{i=1}^n$ 对应的投影 $\{\mathbf{t}_i\}_{i=1}^n$; 在集合 Y 的 Karcher mean (μ_y) 处的切空间处张成子空间的 k 个成分 $W_Y = \{W_Y^i\}_{i=1}^k$, 以及 $\{Y_i\}_{i=1}^n$ 对应的投影 $\{\mathbf{u}_i\}_{i=1}^n$:

- 1: 计算 μ_x, μ_y 和 $\{\hat{X}_i = \log_{\mu_x}(X_i)\}_{i=1}^n, \{\hat{Y}_i = \log_{\mu_y}(Y_i)\}_{i=1}^n$
- 2: 利用群操作 (文献 [35]) 将样本变换到单位矩阵的切空间:

$$\begin{aligned} \log_{\mu_x}(X_i) &\rightarrow \mu_x^{-1/2} \log_{\mu_x}(X_i) \mu_x^{-1/2} = \log(\mu_x^{-1/2} X_i \mu_x^{-1/2}) \triangleq \tilde{X}_i \\ \log_{\mu_y}(Y_i) &\rightarrow \mu_y^{-1/2} \log_{\mu_y}(Y_i) \mu_y^{-1/2} = \log(\mu_y^{-1/2} Y_i \mu_y^{-1/2}) \triangleq \tilde{Y}_i \end{aligned}$$

- 3: 在 $\{\tilde{X}_i\}_{i=1}^n$ 以及 $\{\tilde{Y}_i\}_{i=1}^n$ 之间执行 PLS 得到 $\hat{W}_X, \hat{W}_Y, T = [\mathbf{t}_1, \dots, \mathbf{t}_k], U = [\mathbf{u}_1, \dots, \mathbf{u}_n]$
 - 4: 利用群操作将 \hat{W}_X 变换到 μ_x 的切空间得到 W_X , 将 \hat{W}_Y 变换到 μ_y 的切空间得到 W_Y
 - 5: **return** W_X, W_Y, T, U
-

事实上, 有了算法5的描述, 从算法中的描述: “在 $\{\tilde{X}_i\}_{i=1}^n$ 以及 $\{\tilde{Y}_i\}_{i=1}^n$ 之间执行 PLS 得到 $\hat{W}_X, \hat{W}_Y, T = [\mathbf{t}_1, \dots, \mathbf{t}_k], U = [\mathbf{u}_1, \dots, \mathbf{u}_n]$ ”, 不难发现这里就是欧式空间中的标准的 PLS 问题 (因为 Karcher mean 的切空间是内积空间, 而通过群操作后的数据在单位阵 I 处的表示也是内积空间), 所以欧式空间中的方法在这里都可以直接使用, 进一步的, 通过公式3-1~3-5的内容即可得到计算多个成分 (投影子空间) 和 PLS 做回归问题的形式化, 由于这只是重复前面的内容, 这里就不再赘述了, 这里在6中给出与算法5类似的用于回归问题的算法。

3.3.2 面向图像集合分类的黎曼流形上的 PLS 回归

这里将算法6算法称为基础版本的黎曼流形上的 PLS 回归, 该算法已经可以直接运用到图像集合分类问题了, 但是仔细回顾公式3-18, 会发现在计算 t_i, u_i 的时候要求 $t_i \in (-\epsilon, \epsilon), u_i \in (-\eta, \eta)$, 也就是要求 $t_i W_X, u_i W_Y$ 在 μ_x, μ_y 的一个小邻域内, 而其本质上是

Algorithm 6 对称正定矩阵流形上的偏最小二乘回归近似算法

Require: 对称正定矩阵集合 $\mathbf{X} = \{X_i\}_{i=1}^n$, 对应的由公式3-5定义的 label 矩阵 Y , 需要计算的成分的个数 k

Ensure: 在集合的 \mathbf{X} 的 Karcher mean (μ) 处的切空间处张成子流形空间的 k 个成分

$\mathbf{W}_X = \{W_X^i\}_{i=1}^k$, 以及 $\{X_i\}_{i=1}^n$ 对应的投影 $\{\mathbf{t}_i\}_{i=1}^n$; 欧式空间中标签集 Y 的投影向量

$\mathbf{W}_y = \{w_y^i\}_{i=1}^k$ 及其对应的投影 $\{\mathbf{u}_i\}_{i=1}^n$

1: 计算 \mathbf{X} 的 Karcher mean (μ) 以及 $\{\hat{X}_i = \log_{\mu}(X_i)\}_{i=1}^n$

2: 利用群操作 (文献 [35]) 将样本移动到单位矩阵的切空间:

$$\log_{\mu}(X_i) \rightarrow \mu^{-1/2} \log_{\mu}(X_i) \mu^{-1/2} = \log(\mu^{-1/2} X_i \mu^{-1/2}) \triangleq \tilde{X}_i$$

3: 在 $\{\tilde{X}_i\}_{i=1}^n$ 以及 Y 之间执行 PLS 回归得到 $\hat{\mathbf{W}}_x, \mathbf{W}_y, T = [\mathbf{t}_1, \dots, \mathbf{t}_k], U = [\mathbf{u}_1, \dots, \mathbf{u}_n]$

4: 利用群操作将 $\hat{\mathbf{W}}_x$ 变换到 μ 的切空间得到 \mathbf{W}_x

5: **return** $\mathbf{W}_X, \mathbf{W}_y, T, U$

要求 SPD 矩阵样本的分布是比较集中的; 这也正是图像集合问题和 DTI 图像中的 Tensor 的重要区别之一 (图像集合中的样本由于维度高样本少, 所以分布往往比较稀疏), 也是将黎曼流形上的 PLS 回归用到图像集合需要解决的问题; 此外还注意到工作 [14,35] 利用 Karcher mean 的切空间构造子流形, 但是前面已经提到并没有理论证明这样的结果是最好的, 所以如果在构造子流形的过程中考虑 label 的信息话我们相信可以找到更适合构建子流形的地方。接下来的内容就是如何针对以上的问题将 PLS 方法适配到图像集合分类上, 并且将这问题的解决方案有机的结合起来, 使得在几乎不增加计算复杂度的前提下方法对分类问题的解决有较大的改进。

3.3.2.1 融合判别信息的子流形构造

本节将针对前面提到的第二个问题: 找到一个比 Karcher mean 更合适 (具有一定判别性) 的点构建子流形的我们的解决方案进行说明。具体方案如下, 首先注意到的是利用公式3-12计算 Karcher mean 的时候, 它是一个无监督的优化过程, 所以并未编码判别信息在其中; 因此这里考虑编码 label 到构造子流形的点 (相当于算法6中的 μ), 这里为方便起见仍然沿用记号 μ 。

该步骤的目标是使得在6中的 $\{X_i\}_{i=1}^n$ 在 μ 处的切向量表示 (经过群操作变换到单位阵之后) $\{\tilde{X}_i\}_{i=1}^n$ 同类相似度大, 异类相似度小 (在实际中使用的是 cosine 相似度, 也就是同类的尽量共线, 不同类的尽量正交), 并保证投影之后“方差”尽量的小 (类似于中心化, 因为直接进行中心化比较困难, 所以在优化的过程要求 $\{X_i\}_{i=1}^n$ 在 μ 处的切向量表示 (在原切空间中) 应该具有 0 均值的特点)。

沿用之前的记号 $\{X_i\}_{i=1}^n$ 是来至于 C 个类的对称正定矩阵的样本 $X_i \in \mathbb{S}_d^+$, 并假设每个类有 $n_i, i = 1, 2, \dots, C$ 个样本 $n = \sum_{i=1}^C n_i$, 样本的 label 我们用公式3-5的表示方法 (这里假设样本是按 label 排序了的), 于是我们有如下的形式化:

- 定义平方相似度矩阵 $F = [\rho_{ij}^2]_{n \times n}$, $\rho_{ij} = \frac{k_{ij}}{\sqrt{k_{ii}k_{jj}}}$, 其中 k_{ij} 由下式定义:

$$\begin{aligned} k_{ij} &= \langle \tilde{X}_i, \tilde{X}_j \rangle \\ &= \langle \log(\mu^{-1/2} X_i \mu^{-1/2}), \log(\mu^{-1/2} X_j \mu^{-1/2}) \rangle \\ &= \langle \log(\mu^{1/2} X_i^{-1} \mu^{1/2}), \log(\mu^{1/2} X_j^{-1} \mu^{1/2}) \rangle \end{aligned} \quad (3-21)$$

- 定义标签矩阵 $P' = 1 - 2YY^T$ 其中 Y 的定义参看公式3-5, 进一步的为了平衡正负样本对之间 (P' 中的 1 的个数和 -1 的个数) 的比例, 这里做一下平衡: 设 $n_0 = \{P'\}$ 中 1 的个数}, $n_1 = \{P'\}$ 中 -1 的个数} 则 定义平衡后的矩阵为 P , 其中:

$$P_{ij} = \begin{cases} \frac{P'_{ij}}{n_0}, & \text{if } P'_{ij} = 1 \\ \frac{P'_{ij}}{n_1}, & \text{if } P'_{ij} = -1 \end{cases} \quad (3-22)$$

- 根据工作 [14], 这里定义 “方差” : $\frac{1}{n} \sum_{i=1}^n \| \log_\mu(X_i) \|^2$
- 最后的目标函数为:

$$\min_{\mu} f(\mu) = \text{tr}(FP^T) + \frac{\lambda}{n} \sum_{i=1}^n \| \log_\mu(X_i) \|^2, \mu > 0 \quad (3-23)$$

其中 λ 是一个平衡因子起到平衡两者权重的作用, 同时也有统一量纲的作用, 避免其中一方因量纲问题绝对占优。

3.3.2.2 融合判别信息的切空间构造问题的优化

不难发现的是公式3-23需要通过优化来求解, 而且约束条件 $\mu > 0$ 表明了问题的解空间是 SPD 矩阵流形空间, 因此这里我们使用黎曼流形上的共轭梯度算法 (conjugate gradient methods) 来优化3-23, 该算法是欧式空间中的共轭梯度算法在黎曼流形上的泛化, 关于黎曼流形上的共轭梯度算法可以参看本文的第四章以及文献 [17], 使用该方法优化的时候, 最主要的输入就是3-23的欧式空间的导数; 这将在接下来的部分进行介绍。

在公式3-23中主要包含两部分: $\text{tr}(FP^T)$ 和 $\frac{\lambda}{n} \sum_{i=1}^n \| \log_\mu(X_i) \|^2$; 这里先对第一部分进行求导 (部分内容参考工作 [66]):

$$\begin{aligned} \text{tr}(FP^T) &= \sum_{i=1}^n \sum_{j=1}^n F_{ij} P_{ij} = \sum_{i=1}^n \sum_{j=1}^n P_{ij} \frac{k_{ij}^2}{k_{ii}k_{jj}} \\ \frac{\partial}{\partial \mu} \text{tr}(FP^T) &= \sum_{i=1}^n \sum_{j=1}^n P_{ij} \left(c_1 \frac{\partial}{\partial \mu} k_{ij} - c_2 \frac{\partial}{\partial \mu} k_{ii} - c_3 \frac{\partial}{\partial \mu} k_{jj} \right) \\ \text{where } c_1 &= \frac{2k_{ij}k_{ii}k_{jj}}{(k_{ii}k_{jj})^2}, c_2 = \frac{k_{ij}k_{ij}k_{jj}}{(k_{ii}k_{jj})^2}, c_3 = \frac{k_{ij}k_{ij}k_{ii}}{(k_{ii}k_{jj})^2} \end{aligned} \quad (3-24)$$

因此, 剩下的工作主要是 $\frac{\partial}{\partial \mu} k_{ij}$ 的导数的计算; 在此之前这里先对 k_{ij} 的形式稍加变换,

这样会比较方便理解后面的 k_{ij} 的导数的形式 (公式3-26):

$$\begin{aligned}
 & \left\langle \log_\mu(X_i), \log_\mu(X_j) \right\rangle_\mu \\
 &= \left\langle \log(\mu^{-1/2} X_i \mu^{-1/2}), \log(\mu^{-1/2} X_j \mu^{-1/2}) \right\rangle \\
 &= \left\langle \log(\mu^{1/2} X_i^{-1} \mu^{1/2}), \log(\mu^{1/2} X_j^{-1} \mu^{1/2}) \right\rangle \\
 &= \text{tr}(\log(X_i^{-1} \mu) \log(X_j^{-1} \mu)) \\
 &= \text{tr}(X_i^{-1/2} \log(X_i^{-1/2} \mu X_i^{-1/2}) X_i^{1/2} X_j^{-1/2} \log(X_j^{-1/2} \mu X_j^{-1/2}) X_j^{1/2}) \\
 &= \text{tr}(\log(X_i^{-1/2} \mu X_i^{-1/2}) X_i^{1/2} X_j^{-1/2} \log(X_j^{-1/2} \mu X_j^{-1/2}) X_j^{1/2} X_i^{-1/2}) \\
 &= \left\langle \log(X_i^{-1/2} \mu X_i^{-1/2}), X_i^{1/2} X_j^{-1/2} \log(X_j^{-1/2} \mu X_j^{-1/2}) X_j^{1/2} X_i^{-1/2} \right\rangle
 \end{aligned} \tag{3-25}$$

下面给出 k_{ij} 的导数的具体形式, 由于比较复杂, 感兴趣的读者可以参看本文第四章以及文献 [9,66] 的相关章节。

$$\begin{aligned}
 \frac{\partial}{\partial \mu} k_{ij} &= X_i^{-1/2} \left(D \log(X_i^{-1/2} \mu X_i^{-1/2}) [\text{sym}(X_i^{1/2} X_j^{-1/2} \log(X_j^{-1/2} \mu X_j^{-1/2}) X_j^{1/2} X_i^{-1/2}] \right) X_i^{-1/2} \\
 &\quad + X_j^{-1/2} \left(D \log(X_j^{-1/2} \mu X_j^{-1/2}) [\text{sym}(X_j^{1/2} X_i^{-1/2} \log(X_i^{-1/2} \mu X_i^{-1/2}) X_i^{1/2} X_j^{-1/2}] \right) X_j^{-1/2}
 \end{aligned} \tag{3-26}$$

至于第二部分的 $\frac{\lambda}{n} \sum_{i=1}^n \|\log_\mu(X_i)\|^2$ 的导数, 不难发现以下的关系 $\|\log_\mu(X_i)\|^2 = \left\langle \log_\mu(X_i), \log_\mu(X_i) \right\rangle_\mu = k_{ii}$, 因此这部分的导数可以根据3-26直接导出来。

$$\frac{\partial}{\partial \mu} \frac{\lambda}{n} \sum_{i=1}^n \|\log_\mu(X_i)\|^2 = \frac{\lambda}{n} \sum_{i=1}^n \frac{\partial}{\partial \mu} k_{ii} \tag{3-27}$$

然后整合两部分的结果可以得到公式3-23在普通欧式空间中关于 μ 的导数如公式3-28所示。

$$\nabla_\mu f(\mu) = \sum_{i=1}^n \sum_{j=1}^n P_{ij} \left(c_1 \frac{\partial}{\partial \mu} k_{ij} - c_2 \frac{\partial}{\partial \mu} k_{ii} - c_3 \frac{\partial}{\partial \mu} k_{jj} \right) + \frac{\lambda}{n} \sum_{i=1}^n \frac{\partial}{\partial \mu} k_{ii} \tag{3-28}$$

最后, 由于我们还需要将普通的梯度转换为黎曼梯度 (因为优化其实是在黎曼流形上的切空间上做的)。根据论文 [9,66] 中的介绍, 只需要对公式3-28作如下的变化即可:

$$\text{grad}_\mu f(\mu) = \mu \nabla_\mu f(\mu) \mu \tag{3-29}$$

将优化公式3-23得到的 μ 代替算法6中的 Karcher mean 并在该点做投影变化, 即得到了融入判别信息的改进方法; 虽然目前的结果不是算法的最终形式, 但是这是最终算法的基础, 在后面的章节中将会反复使用到, 所以这里将其总结在算法7中。

注意到这仍然没有解决一开始提到的另一个问题: 数据不紧凑所带来的问题, 接下来的部分将针对这个问题做进一步的改进。

Algorithm 7 对称正定矩阵流形上具有判别性的切空间偏最小二乘回归近似算法

Require: 对称正定矩阵集合 $\mathbf{X} = \{X_i\}_{i=1}^n$, label 矩阵 \mathbf{Y} , 需要计算的成分的个数 k

Ensure: 融入判别性的切空间 $T_\mu M$ 对应的 μ , 集合 \mathbf{X} 在 $T_\mu M$ 中的 k 个成分 $\mathbf{W}_X = \{W_X^i\}_{i=1}^k$, 以及 $\{X_i\}_{i=1}^n$ 对应的投影 $\{\mathbf{t}_i\}_{i=1}^n$; 欧式空间中标签集 \mathbf{Y} 的投影向量 $\mathbf{W}_y = \{\mathbf{w}_y^i\}_{i=1}^k$ 及其对应的投影 $\{\mathbf{u}_i\}_{i=1}^n$

- 1: 初始化 $\mu = \mu_0$ (通常为 I) 求解最小化问题3-23获得 μ , 然后计算 $\{\hat{X}_i = \log_\mu(X_i)\}_{i=1}^n$
- 2: 利用群操作 (文献 [35]) 将样本移动到单位矩阵的切空间:

$$\log_\mu(X_i) \rightarrow \mu^{-1/2} \log_\mu(X_i) \mu^{-1/2} = \log(\mu^{-1/2} X_i \mu^{-1/2}) \triangleq \tilde{X}_i$$

- 3: 在 $\{\tilde{X}_i\}_{i=1}^n$ 以及 \mathbf{Y} 之间执行 PLS 回归得到 $\hat{\mathbf{W}}_X, \mathbf{W}_y, T = [\mathbf{t}_1, \dots, \mathbf{t}_k], U = [\mathbf{u}_1, \dots, \mathbf{u}_n]$
- 4: 利用群操作将 $\hat{\mathbf{W}}_x$ 变换到 μ 的切空间得到 \mathbf{W}_x
- 5: **return** $\mathbf{W}_X, \mathbf{W}_y, T, U$

3.3.2.3 多切空间逐步回归的偏最小二乘方法

首先注意到的是如果直接对公式3-19中的 t_i 或 u_i 进行截断的话, 这不仅会损失精度 (因为原始数据本来分布很稀疏, 再限制 t_i 或 u_i 的取值范围无疑会使得估计更加的不准), 而且在优化的时候也会比较麻烦; 所以这里考虑寻找多个点使用多个切空间来缓解数据稀疏的问题; 在这样的框架下, 每个切空间并不需要很强的表示能力, 只需要能够反应原始数据的一部分结构就行了, 而最后的联合所有表示的模型可以融合这些子模型的特点得到更具判别性的表示。

确定了以上的方法路线后, 接下来的问题就是如何选取各个切空间以及如何将他们有效的融合起来, 这里我们使用逐步回归的方案来同时解决这两个问题: 对算法6或7中的数据 $\mathbf{X} = \{X_i\}_{i=1}^n$ 和 label 矩阵 \mathbf{Y} 执行算法7得到获得一个切空间 $T_{\mu_1} M$ 及其对应的 k 个投影方向 $\mathbf{W}_X^{(1)}, \mathbf{W}_y^{(1)}$ 和对应的 T_1, U_1 ; 然后对 $\{\tilde{X}_i\}_{i=1}^n$ 以及 \mathbf{Y} 利用公式3-4进行 defalte 操作: $\{\tilde{X}_i\}_{i=1}^n \xrightarrow{\text{defalte}} \{\tilde{X}_{res_i}\}_{i=1}^n, Y \xrightarrow{\text{defalte}} Y_{res}$, 然后对 $\{\tilde{X}_{res_i}\}_{i=1}^n$ 利用群操作从单位阵处将数据变换到 μ 再用 $\exp_{\mu_1}(\cdot)$ 将数据 $\{\tilde{X}_{res_i}\}_{i=1}^n$ 变换到 SPD 矩阵流形空间得到 $\{\tilde{Z}_{res_i}\}_{i=1}^n$, 最后将 $\{\tilde{Z}_{res_i}\}_{i=1}^n, Y_{res}$ 赋值给 $\mathbf{X} = \{X_i\}_{i=1}^n$ 和 \mathbf{Y} , 并重新初始化 μ_2 后开始第二次回归; 反复上述过程直到获得指定个数的切空间, 算法终止。将上述过程用算法8描述。

在训练数据上运用算法8的到训练参数后, 测试的部分与普通的算法7或6的测试类似, 所不同的是算法8中回归的 \mathbf{Y} 是所有切空间中结果的综合 (逐步回归相加) 得到的。

关于训练, 算法8与7和6有些许的不同: 首先是算法8的参数 k 的选择往往要比后两者的小很多, 例如在 YTC[36] 数据集上算法6和7的参数 k 默认设置为类别数减一 (46=47-1) 而且实验发现减小 k 会使得算法的性能降低, 而算法8中 k 的选择为 26, 过高或过低都会降低算法的性能 (原因可能是欠拟合或过拟合了); 算法8的另一个重要的参数的选择就是 p , 这个也会有欠拟合和过拟合的现象; 最后是在测试的时候发现, 数据是否中心化对算的最终结果也有一定的影响, 这个问题在 UIUC[39] 数据上做材料分类的时候

Algorithm 8 对称正定矩阵流形上多切空间逐步回归的偏最小二乘近似算法

Require: 对称正定矩阵集合 $X = \{X_i\}_{i=1}^n$, label 矩阵 Y , 每个切空间计算的成分的个数 k , 指定切空间的个数 p

Ensure: 融入判别性的 p 个切空间 $T_{\mu_1}M, \dots, T_{\mu_p}M$ 对应的 μ_1, \dots, μ_p , 数据 X 在 $T_{\mu_1}M, \dots, T_{\mu_p}M$ 中各自的 k 个成分 $\hat{\mathbf{W}}_x^{(1)}, \dots, \hat{\mathbf{W}}_x^{(p)}$, 以及对应的投影 T_1, \dots, T_p ; 欧式空间中标签集 Y 逐次回归的投影矩阵 $\hat{\mathbf{W}}_y^{(1)}, \dots, \hat{\mathbf{W}}_y^{(p)}$ 及其对应的投影 U_1, \dots, U_p

- 1: 初始化 $output = cell(1, p)$ 为包含 p 个 cell 的结构
- 2: **for** $j = 1; j \leq p; j = j + 1$ **do**
- 3: 初始化 $\mu_j = \mu_0$ (通常为 I) 最小化 3-23 问题获得 μ_j , 然后计算 $\{\tilde{X}_i = \log_{\mu_j}(X_i)\}_{i=1}^n$
- 4: 利用群操作 (文献 [35]) 将样本移动到单位矩阵的切空间:

$$\log_{\mu_j}(X_i) \rightarrow \mu_j^{-1/2} \log_{\mu_j}(X_i) \mu_j^{-1/2} = \log(\mu_j^{-1/2} X_i \mu_j^{-1/2}) \triangleq \tilde{X}_i$$
- 5: 在 $\{\tilde{X}_i\}_{i=1}^n$ 以及 Y 之间执行 PLS 回归得到:

$$\hat{\mathbf{W}}_x^{(j)}, \hat{\mathbf{W}}_y^{(j)}, T_j = [\mathbf{t}_1^j, \dots, \mathbf{t}_k^j], U_j = [\mathbf{u}_1^j, \dots, \mathbf{u}_k^j]$$
- 6: 利用群操作将 $\hat{\mathbf{W}}_x^{(j)}$ 变换到 μ_j 的切空间得到 $\mathbf{W}_x^{(j)}$
- 7: 对 X, Y 使用 deflate 操作 3-4: $\{\tilde{X}_i\}_{i=1}^n \xrightarrow{\text{defalte}} \{\tilde{X}_{res_i}\}_{i=1}^n, Y \xrightarrow{\text{defalte}} Y_{res}$
- 8: 利用群操作从单位阵处将数据 $\{\tilde{X}_{res_i}\}_{i=1}^n$ 变换到 μ_j 然后用 $\exp_{\mu_j}(\cdot)$ 将结果变换到 SPD 矩阵流形空间得到 $\{Z_{res_i}\}_{i=1}^n$
- 9: 将 $\{Z_{res_i}\}_{i=1}^n, Y_{res}$ 赋值给 $X = \{X_i\}_{i=1}^n$ 和 Y
- 10: 保存此次结果: $[\mu_j, \mathbf{W}_x^{(j)}, \mathbf{W}_y^{(j)}, T_j, U_j] \rightarrow output\{j\}$
- 11: **end for**
- 12: **return** $output$

尤其明显。

3.4 实验验证

前面的章节对问题的背景, 已有的方法, 存在的问题以及本文的动机和针对问题的解决的方案等做了阐述, 本节将会实验验证前面的方法并从实验结果出发分析方法的特点和存在的问题等。

本节最要会从以下几个计算机视觉的任务进行验证: 物体识别 (数据库 ETH80[37]), 材料分类 (数据库 UIUC[39]) 以及视频人脸识别 (数据集 YTC[36]); 由于这些数据集已经在 1.4 这一节进行了介绍, 并且这些数据集的测试协议也已经在这一节介绍, 所以这里就不再进行阐述, 如果读者对数据或测试协议有什么不明的话可以到 1.4 一节进行查看。

3.4.1 原始特征构造

这里将简单的介绍一下，各个任务对应的数据集上的基本特征的构造，这些特征提取属于是底层特征的提取将用于最后的 SPD 矩阵（实际上也可看做是一种特征表示）的构造。首先，在用于物体识别的 ETH80[37] 数据集上，所有的图片被预先 resize 成 20×20 的大小图片，然后灰度特征被直接用于物体识别任务；在材料识别任务的 UIUC[39] 数据集上，我们使用 Region Covariance[52] 表示一张图片，参考 [22]，这里的 Region Covariance 的构造中我们使用 128 维的 dense SIFT[40] 特征作为基本的特征，首先将图片 resize 到 400×400 然后，以 4 个像素为间隔（每个块的大小为 16×16 ，共 8 个角度，4 个 bin）划分网格，在每个网格点 128 维的 SIFT 特征被提取作为构造 Region Covariance 的基本特征（与工作 [22] 中的不同的是工作 [22] 还融合了颜色特征）；最后在视频人脸识别任务的数据库 YTC[36] 上，首先将图片 resize 到 20×20 然后直方图均衡化被用于鲁棒的特征构造。

3.4.2 SPD 矩阵表示的构造

在“原始特征构造3.4.1”这一小节介绍了基本的图像特征的提取和预处理方式，在本节中将对使用这些基本特征构造 SPD 矩阵表示做简要的介绍：首先是材料识别任务的 UIUC[39] 数据集上的 SPD 表示，由于使用的是 Region Covariance[52]，这里使用的是标准的构造方式，故不再做进一步介绍，有兴趣的读者可以参看文献 [52]；在物体识别的 ETH80[37] 和视频人脸识别的 [36] 数据集上，这里根据工作 [57] 中的内容使用 SPD 矩阵表示图像集合，但是稍微有些不同的是：根据 [29,59] 中的构造方式，均值的信息也被融入了图像集合的 SPD 特征表示当中（公式3-30中的 Σ 是样本协方差， μ 是样本均值， d 是样本的维数）：

$$C = \det(\Sigma)^{-\frac{1}{d+1}} \begin{bmatrix} \Sigma + \mu\mu^T & \mu \\ \mu^T & 1 \end{bmatrix} \quad (3-30)$$

最后还需要一提的是对于所有的原始特征实验中都做了 95% 的 PCA 降维处理；对于样本协方差矩阵 Σ 奇异的时候（往往是由于样本个数小于样本维度造成的），根据 [30,59] 中的做法，一个小小的正的正则项： $\delta I, \delta = 10^{-3} \times \text{tr}(C)$ （其中 I 是单位阵）被加到 Σ 上： $\Sigma + \delta I \rightarrow \Sigma$ 。

3.4.3 实验结果与分析

本小结的内容是 RPLS 方法在物体识别，材料识别以及视频人脸识别三个任务上的实验结果呈现，在对比方法中我们选取了具有代表性的方法：基于 PLS 的协方差判别学习方法（Covariance Discriminant Learning）CDL-PLS[57]，稀疏编码学习的方法 RSR（Riemannian Sparse Representation），然后是 ECCV (European Conference On Computer Vision) 2014 年的工作 SPDML (SPD-Manifold Learning) 的两种度量 (Stein Divergence[51])

和 Affine Invariant Metric[46]）的结果，以及使用分布建模集合的方法 BeyondGauss[20] 和 SPD 矩阵流形上的度量学习（Metric Learning）方法 LEML[30]，表3.1给出了这些方法在三个任务上的实验对比结果，所有的结果均是按照1.4节的协议获得的，对于其它文章的方法，这里从作者的主页上获得源代码并小心的调整参数后报告的是在1.4节的协议下所获得的最好的结果：

表 3.1 黎曼流形上的 PLS 回归算法实验结果

数据集 方法	ETH80	UIUC	YTC
CDL-PLS[57]	93.25±4.72	53.89±4.06	70.28±2.13
RSR-Stein[23]	93.25±3.34	52.41±4.03	72.77±2.69
SPDML-Stein[22]	90.50±3.87	49.17±2.37	61.57±3.43
SPDML-AIM[22]	90.75±3.34	48.09±1.82	64.66±2.92
BeyondGauss[20]	84.75±6.29	N/A	71.46±2.61
DARG[59]	92.25±2.19	N/A	77.09±1.92
LEML[30]	94.75±2.49	48.98±3.69	70.53±2.95
RPLS _{single}	92.75±4.32	54.72±3.61	74.48±2.79
RPLS _{multi}	95.50±2.58	56.57±3.49	77.33±2.95

其中我们将方法简称为 RPLS，R 表示的是 Riemannian 的意思。RPLS 算法的下标表示的基础版本的黎曼流形上的 PLS 算法 (RPLS_{single}) 还是多切空间逐步回归的算法 (RPLS_{multi})。最终的实验结果验证了我们最初的猜想，多切空间逐步回归算法在三个任务上都获得了 state-of-the-art 的结果；我们将取得这样的结果的原因归纳如下（与 CDL[57] 相比）：1) 首先是按照公式3-30构造的 SPD 表示中均值信息者带来了一定性能上的提升；2) 考虑切空间的选择带来了表格倒数第二行和表格第一行的变化（因为 CDL-PLS 与在单位阵的 RPLS_{single} 是等价的）；3) 最后是单切空间与多且空间的方案差别带来了表格中最后两行的变化，同时也力证多切空间逐步回归算法8的有效性。

试验中我们发现公式3-23中的第二项对算法性能有小幅的提升，试验中我们始终固定公式中 $\lambda = 0.001$ 。而在前面我们也有介绍，算法8中的参数 k, p 对于算法的影响是较大的，也是本算法需要改进的一大方向。

最后在对比的一系列的方法中，BeyondGauss[?] 的方法由于是使用 KDE 来估计分布函数，而在 UIUC[39] 这个数据集上原始的特征是 Dense Sift（样本非常多），直接导致了无法计算的问题，所以这里的结果没有汇报（N/A），其它的结果是在 hellinger 散度下的结果。表格中对比方法的结果都是从作者主页获取的代码小心调参后获得的最好的结果。

3.5 总结与下一步工作

本章从子流形与投影的概念出发，参考相关工作 [14,35,46,57] 等首先导出了黎曼流形上的偏最小二乘问题以及偏最小二乘用于回归的一般形式，然后以 SPD 矩阵流形为例将算法形式化⁶，最后针对图像集合分类问题与 DTI (Diffusion Tensor Images) 的不同（主要是数据更稀疏），提出了两点通用的改进（这里之所以说是通用的改进，是因为即便数据聚集在流形空间的小范围内这些改进依然是适用的）得到了多切空间逐步回归的偏最小二乘算法，使得算法可以适应这种数据稀疏的情况，最后的实验验证部分验证了方法的有效性。

实验分析部分以及算法8分析部分我们有提到，算法8中的参数 k, p 太大会过拟合太小又会欠拟合，分析其原因的话可能是逐步回归的方式没能有效的组合各个切空间中的信息，接下来可能参考 [53] 中的方法使用 Adaboost 的框架进行多个切空间的组合。

第四章 Low-rank PSD 矩阵判别学习方法

前面的章节已经提到统计建模图像集合的方法中有相当一部分的数据都是以对称正定 (SPD) 矩阵的形式存在的，但其中很现实的一个问题的是样本数小于数据维度，因为即使是 20×20 的小图像也有 400 维，也就是说至少需要 400 个样本才有可能使得样本协方差矩阵是非奇异的；这不管是对 multi-view 的图片集合还是对视频监控（400 帧的图像在 30FPS 的帧率下也需要 13 秒还多）都是很苛刻要求的；因此实际中获得的样本协方差矩阵实际上是对称半正定 (Positive Semi-Definite) 矩阵；在使用对称正定 (PSD) 矩阵建模图像集合的问题中，针对该问题的一个常用的 trick 是给样本协方差矩阵 Σ 的对角线上加上一个正则项（正如 3.4.2 中介绍的一样）；但是这并没从本质上解决样本协方差矩阵奇异的问题，这促使本文回到数据所在的原始空间——对称半正定矩阵空间研究图像集合的建模和判别学习问题。

注意到，虽然原始的对称半正定 (PSD) 矩阵空间是一个凸集合，是很好的一个性质；但是遗憾的是以目前了解到的情况而言，无约束的对称半正定集合上并没有很好的定义的数学结构，与之最接近的一个数学结构是 Fixed-Rank PSD 集合（公式 4-1）

$$\mathbb{S}_d^+(k) = \{A | A \in \mathbb{R}^{n \times n}, A = A^T, \langle Ax, x \rangle \geq 0, \forall x \in \mathbb{R}^n; \text{rank}(A) = k\} \quad (4-1)$$

如果在该结构上定义合适的黎曼度量的话就可以使得该集合形成一个黎曼流形结构 [8]；另一方面还注意到的是使用对称正定矩阵表示图像集合的时候，特征的维度往往非常高，如 20×20 的图像组成的集合，它的样本协方差将达到 400×400 的规模，这对存储和计算都是不小的负担，因此 [22,30] 研究了 SPD 流形的降维问题，而不难发现的是当图像集合使用的 PSD 矩阵表示且矩阵的秩 (Rank) 很低的时候 PSD 矩阵表示间体现出另一个重要的性质：Low-Rank；Low-Rank 的性质将大大的降低存储容量和计算时间，而这正是我们想要的；因为如果我们将 SPD 矩阵 S 进行 $S = WW^T$, $W \in \mathbb{R}^{d \times k}$, $\text{rank}(W) \leq k \ll d$ 分解，此时只需要存储 W 即可，这将大节省存储空间和计算量。

最后，回顾图像集合建模的两大分支：子空间的方法 [18,63] 等和统计模型建模的方法 [22,29,42,54,57,59] 可以发现如下事实：对于样本协方差矩阵 Σ ，及其特征分解 $\Sigma = U\Lambda U^T$ ；子空间的方法其实只用到了 U 的信息，还有大部分的信息没有被使用，而统计模型的方法使用了整个协方差矩阵的信息，并且只有很少的文章考虑了数据中的噪声问题，而这种包含所用信息的编码方式很可能将噪声信息也编码进了协方差的表示当中；最后反观 Low-Rank PSD 矩阵表示中当 $k < d$ 的时候，它更像是两者的中间状态，兼顾了两者优点。

接下来内容大致安排如下：首先结合着前期的一些工作将与 Low-Rank PSD 关系最

密切的 Fixed-Rank PSD 流形进行介绍，此外为了介绍 Fixed-Rank PSD 矩阵这里还会用一些篇幅介绍一下 Stiefel Manifold 和 Grassmann Manifold；然后针对 Fixed-Rank PSD 矩阵流形建模图像集合的特点结合工作 [43] 存在的一些问题提出 Low-Rank PSD 矩阵建模图像集合的问题；接着是本文提出的改进方法（也就是本章方法的介绍），紧接着是实验的验证，最后是总结与展望。

4.1 Stiefel 流形和 Grassmann 流形

这部分会对对称正定矩阵流形以外的两种流形进行介绍，它们分别是 Stiefel 流形和 Grassmann 流形，这里之所以同时介绍这两种流形结构主要是出于以下的考虑：其一，Grassmann 与 Stiefel 流形的关系十分密切使得两者需要同时介绍，其二，由于 Fixed-Rank PSD 流形的特殊性，需要借助 Grassmann 流形和 SPD 矩阵流形（在 2.1.2 节已经介绍）来研究，因此也需要先介绍 Grassmann 流形。本节的内容只是对 Stiefel 流形和 Grassmann 流形做简要的介绍，更多关于 Grassmann 流形以及两者的关系的内容可以参看 [34]

这里的内容以基本定义居多，但相较于 2.1.2 节的内容这里的相对简单一些；接下来就首先从 Grassmann 流形的定义开始

定义 4.1 (Grassmann 流形) 在 \mathbb{R}^n 中所有 k 维的线性子空间构成的集合表示为

$$\text{Gr}(k, n) = \{\mathbb{V} \subset \mathbb{R}^n, \mathbb{V} \text{ is a linear subspace with } \dim \mathbb{V} = k\} \quad (4-2)$$

当为其定义拓扑结构之后，即可构成流形空间结构，进一步的可以在其上导出黎曼度量 [34]，所以它其实是黎曼流形空间的特例。

接下来将会具体对 Grassmann 流形进行介绍，不过在具体介绍 Grassmann 流形之前，考虑到 Grassmann 流形与 Stiefel 流形之间的密切关系以及数学上表示的方便性，需要先对 Stiefel 流形进行介绍：

定义 4.2 (Stiefel(non-compact) 流形) 由所有 $n \times k, (0 < k < n)$ 的满秩矩阵构成的空间称为 Stiefel 流形空间^①

$$\begin{aligned} \text{St}(k, n) &= \{A \in \mathbb{R}^{n \times k}; \text{rank}(A) = k\} \\ &= \{A = (a_1, \dots, a_k) \in \mathbb{R}^{n \times k}; a_1, \dots, a_k \text{ are linearly independent}\} \end{aligned} \quad (4-3)$$

定义 4.3 (Stiefel(compact) 流形) 当在 non-compact Stiefel 流形的定义中要求 A 的列向量是正交的（即： $A^T A = I_k$ ）时候

$$\text{St}^*(k, n) = \{A \in \mathbb{R}^{n \times k}; A^T A = I_k\} \quad (4-4)$$

即定义了 Stiefel (compact) 流形空间^①。

^① 以下所说的流行空间均是指集合上定义了拓扑结构的流形空间，为简洁起见文中仅以集合代替，而不细说其上的拓扑结构

最后再给出一个在数学中非常重要的概念：“广义线性群（General Linear Group）”，在矩阵流形的研究中将常常见到它的身影

定义 4.4（广义线性群（General Linear Group）） 在数学中对于指定的 n 将所有 $n \times n$ 的可逆矩阵的 **空间** 称为广义线性群，并记为

$$\mathrm{GL}(n) = \{A \in \mathbb{R}^{n \times n}; \det(A) \neq 0\} \quad (4-5)$$

至此，介绍 Grassmann 流形的准备工作就算是基本完成了，接下来就是利用这些定义以及 2.1.1 节的内容对 Grassmann 流形做进一步的介绍；而在前面的章节中已经说过 Stiefel 流形和 Grassmann 流形有密切的关系，所以这里先来捋一捋这两者的关系，方便理解：

关系 4.1 (Stiefel(non-compact) VS Stiefel(compact)) 定义 $\mathrm{GS}(\cdot)$ 表示 Gram-Schmidt 正交化变换，于是

$$\mathrm{GS} : \mathrm{St}(k, n) \rightarrow \mathrm{St}^*(k, n) \quad (4-6)$$

显然 $\mathrm{GS}(\cdot)$ 变换是满射但是不是入射所以 $\mathrm{GS}(\cdot)$ 不是一一的映射

关系 4.2 (Stiefel(non-compact) VS Grassmann) 同样这里用一个变换 π 描述两者之间的关系：

$$\pi : \mathrm{St}(k, n) \rightarrow \mathrm{Gr}(k, n), A = (a_1, \dots, a_k) \rightarrow \mathrm{span}(A) \quad (4-7)$$

同样的 π 只是满射但是不是入射

关于映射 π 的还有一些性质需要了解：

- 首先已知 π 是满射，对于任意的矩阵 $A \in \mathrm{St}(k, n)$ 有

$$\pi^{-1}[\pi(A)] = \{AP; P \in \mathrm{GL}(k)\} \quad (4-8)$$

- 映射 π 是连续（continuous）的开（open）的（连续性不作过多解释，开（open）的就是说映射的像是开集的话原像也是开集）

关系 4.3 利用映射 π 以及 $\mathrm{St}(k, n)$ 即可对 Grassmann 的拓扑结构进行研究。（Stiefel(compact) VS Grassmann） 类似于定义 4.2，这里定义 Stiefel(compact) 和 Grassmann 之间的映射 $\bar{\pi}$

$$\bar{\pi} : \mathrm{St}^*(k, n) \rightarrow \mathrm{Gr}(k, n); A = (a_1, \dots, a_k) \rightarrow \mathrm{span}(A), A^T A = I_k \quad (4-9)$$

不难发现映射 $\bar{\pi}$ 与 π 之间存在如下的关系： $\pi = \bar{\pi} \circ \mathrm{GS}$

前面给出了诸多定义和关系，其目的主要是为了接下来给出 Grassmann 流形的一个数学表示，方便对其进行研究并运用到特定问题中。

表示 4.1 根据前面的介绍，这里可以将 Grassmann 流形表示为如下的商空间（quotient space）的形式

$$St(k, n)/GL(k, \mathbb{R}) = \{[A]|[A] \triangleq A[GL(k)]; A \in St(k, n)\} \quad (4-10)$$

利用上述的表示，可以定义商空间流行结构

表示 4.2 同样是利用商空间的概念：

$$\begin{aligned} St^*(k, n)/O(k) &= \{[A]|[A] \triangleq A[O(k)]; A \in St^*(k, n)\} \\ O(k) &= \{U|U \in \mathbb{R}^{k \times k}; U^T U = I_k\} \end{aligned} \quad (4-11)$$

在两种表示中，第二种表示方法更为常用，对其的研究也相对成熟一些；

在这一小节的最后我们将介绍 Grassmann 流形上的度量表示。为此先介绍两个基本概念：“主夹角”和“投影变换”

定义 4.5（主夹角）假定 $X_1, X_2 \in St^*(k, \mathbb{R}^n)$ 表示两个子空间的基矩阵，于是可以定义子空间 $V_1 = \text{span}(X_1), V_2 = \text{span}(X_2)$ 之间的主夹角为：

$$\begin{aligned} \cos \theta_i &= \max_{u_i \in V_1} \max_{v_i \in V_2} u_i^T v_i \\ \text{s.t. } u_i^T u_i &= 1, v_i^T v_i = 1 \\ u_i^T u_j &= 0, v_i^T v_j = 0; j \leq i \end{aligned} \quad (4-12)$$

其中的 $\theta_i, i = 1, 2, \dots, k$ 就称为主夹角，而 $\cos \theta_i$ 则称为典型相关系数。

定义 4.6（投影变换）将一个高维空间中的点投影到一个低维子空间的算子：

$$\Pi_k : U \rightarrow UU^T; U \in St^*(k, \mathbb{R}^n) \text{ and } UU^T \in \mathbb{S}_n \quad (4-13)$$

其中 \mathbb{S}_n 表示的是对称矩阵构成的空间。实际上，投影矩阵 UU^T 是半正定的。

利用上述两个概念（“主夹角”和“投影变换”）这里介绍在 Grassmann 流形上常用的两个度量的：“投影度量”和“比奈-柯西度量”

定义 4.7（投影度量）假定 $X_1, X_2 \in St^*(k, \mathbb{R}^n)$ 表示两个子空间的基矩阵，并且有主夹角 $\{\theta_i\}_{i=1}^k$ ，两者之间定义投影度量如下：

$$d(X_1, X_2) = \|\Pi_k(X_1) - \Pi_k(X_2)\|_2 = \left(\sum_{i=1}^k \sin^2(\theta_i) \right)^{\frac{1}{2}} \quad (4-14)$$

其中的 $\Pi_k(\cdot)$ 就是前面的投影变换，这也与“投影度量”的由来有关。

定义 4.8 (比奈-柯西度量) 前面定义的主夹角 (Principle Angle) 的 cosine 值 $\{\cos(\theta_i)\}_{i=1}^k$ 又称为典型相关系数, “比奈-柯西度量”的定义就是利用典型相关系数的乘积来定义的

$$d(X_1, X_2) = \left(1 - \prod_{i=1}^k \cos^2(\theta_i)\right)^{1/2} \quad (4-15)$$

至此, Grassmann 流形的介绍就基本结束了, 更多关于 Grassmann 流形的内容 (如其上的微分结构, 黎曼度量等) 可以参看文献 [34], 里面有更加详细的介绍, 但是可能会比较晦涩。

4.2 Fixed-Rank PSD 流形

公式 4-1 给出了 $d \times d$ 秩为 k 的半正定矩阵组成的集合, 接下来的部分参考自 [8], 将会对 Fixed-Rank PSD 流形的 Geomentry 结构做简要的介绍。

对于任意的元素 $A \in \mathbb{S}^k$ 假设其 svd 分解表示为 $A = UR^2U^T$ 则这里定义如下的问题形式:

$$A = UR^2U^T = (UR)(UR)^T \triangleq ZZ^T; F \in \text{St}(k, n) \quad (4-16)$$

其中 $\text{St}(k, n)$ 就是定义 4.2 所定义的 non-compact Stiefel 流形所在的集合; 并且这里注意到对任意的正交阵 $O \in O(k)$ 有 $(ZO)(ZO)^T = ZZ^T$ 表示的是同一个半正定矩阵因此定义如下的群变换 (Group Action):

$$\begin{aligned} R &\rightarrow O^T R O \in \mathbb{S}_k^+ \\ U &\rightarrow U O \in \text{St}^*(k, n) \\ ZO &= URO = UOO^T R O \end{aligned} \quad (4-17)$$

最后定义如下的直积的形式来重新表示对称半正定的矩阵, 其中 \sim 表示等价关系。

$$(U, R^2) \sim (UO, O^T R^2 O) \in \text{St}^*(k, \mathbb{R}^n) \times \mathbb{S}_k^+ \quad (4-18)$$

运用上述的公式 (U, R^2) 来表示固定秩的半正定矩阵 A 的时候, 在其切空间 (具体形式可参看文献 [8]) $T_A \mathbb{S}_d^+(k)$ 处的无穷小变量 (Δ, D) 可以定义为

$$\begin{aligned} \Delta &= U_\perp B, \\ D &= RD_0R, \end{aligned} \quad (4-19)$$

其中 $U_\perp \in \text{St}^*(n-k, \mathbb{R}^n)$, $U^T U_\perp = \mathbf{0}$; $B \in \mathbb{R}^{(n-k) \times n}$, 而 $D_0 \in \mathbb{S}_d$ 。有了以上的定义后, 进一步的参考 $\text{St}^*(k, \mathbb{R}^b)$ 和 \mathbb{S}_k^+ 上的黎曼 metric 可在这里定义两个微小变量之间的如下关系 4-20。

$$\begin{aligned} g_{(U, R^2)}((\Delta_1, D_1), (\Delta_2, D_2)) \\ = \text{tr}(\Delta_1 \Delta_2) + p \text{tr}(R^{-1} D_1 R^{-2} D_2 R^{-1}), p > 0 \end{aligned} \quad (4-20)$$

事实上可以证明的是上式定义了切空间 $T_A \mathbb{S}_d^+(k)$ 上的黎曼度量 [8]。有了黎曼度量之后再来看看固定秩半正定矩阵 A, B 之间的两个元素之间的一条曲线：

$$\left\{ \begin{array}{l} let : A \sim (U_A, R_A^2), B \sim (U_B, R_B^2) \\ PA \text{ of } U_A, U_B : \Theta = \text{diag}(\theta_1, \theta_2, \dots, \theta_k) \\ part1 : U(t) = U_A \cos(\Theta t) + X \sin(\Theta t) \\ part2 : R^2(t) = R_A \exp(t \log(R_A^{-1} R_B^2 R_A^{-1})) R_A \\ curve : \gamma_{A \rightarrow B}(t) = U(t) R^2(t) U(t)^T \\ length : L(\gamma_{A \rightarrow B}) = \|\Theta\|_F^2 + p \|\log(R_A^{-1} R_B^2 R_A^{-1})\|_F^2 \end{array} \right. \quad (4-21)$$

其中 PA 是 Principle Angles 的缩写； $X = (I - U_A U_A^T) U_B F$ 其中 F 是 $\text{diag}(\sin(\theta_1), \sin(\theta_2), \dots, \sin(\theta_k))$ 的逆（或伪逆）；而上式中的 $L(\gamma_{A \rightarrow B})$ 给出了曲线之间的长度，根据定义2-1如果要进一步的给出测定距离的话需要找到链接 A, B 所有曲线中长度最小的曲线 $\gamma_{A \rightarrow B}^*(t)$ ，这对于固定秩的对称正定矩阵流形太过复杂，不过好在文献 [8] 证明的了 $L(\gamma_{A \rightarrow B})$ 是测地距离的一个不错估计（虽然它不满足三角不等式），所以接下来就可以运用该“度量”（文献 [8,44] 中将其称为 polar metric）

$$\delta^2(A, B) = \|\Theta\|_F^2 + p \|\log(R_A^{-1} R_B^2 R_A^{-1})\|_F^2 \quad (4-22)$$

4.3 Fixed-Rank PSD 流形研究概况

本节的主要内容是介绍黎曼流形上的判别学习方法，主要的内容大致会分为如下的一些部分：首先是 Fixed-Rank PSD 流形用于图像集合的分类问题的研究现状和方法介绍，接着是针对目前研究现状的一些问题和个人对该问题的理解对现有方法的改进

关于 PSD 的研究主要分为以下几个方向：比较存粹的偏理论的研究（如工作 [8,44] 等），然后是做优化的（半正定规划是优化问题的一大组成部分）研究（如工作 [33]），再者是做 metric Learning 的一批研究者（如 AAAI’2016 的工作 [45]），最后是做图像集合分类的目前据我所知只有工作 [43]，由于该工作的前瞻性，所以这里会简单的介绍一下，随带介绍一下 Fixed-Rank PSD 流形做图像集合分类的这个分支，而其它的分支，如基础理论的已经在4.2部分做了介绍，优化与 metric learning 的分支将会在对工作 [45] 进行介绍的时候提及，所以这里不再赘述了，接下来就先看一下 [43] 的工作。

工作 [43] 的主要贡献是针对第二章开头部分提到的问题，首次利用 Fixed-Rank PSD 矩阵建模图像集合，用于图像集合的分类问题；文中所提出的方法也十分直接，遵循了图像集合分类问题的两条基本思路：1. 为图像集合寻找一种表示（这里选择的就是 Fixed-Rank PSD 矩阵）；2. 为这种表示寻找/推导一种度量用于描述两个集合表示的相似度/距离。

4.3.1 Fixed-Rank PSD 表示图像集合

首先，不妨假设图像集合共有 C 个类别 n 个图像集合，并使用标签 $y_i \in \{1, 2, \dots, C\}^n, i = 1, 2, \dots, n$ 标记 (label)，并且每个集合有 n_i 个样本，每个样本来自 \mathbb{R}^d 的空间，于是将由图像集合构造 Fixed-Rank PSD 矩阵表示的过程描述如下（设 rank = k ）：

- 设 $\{\mathbf{x}_{ij} \in \mathbb{R}^d\}_{j=1}^{n_i}$ 表示第 i 个图像集合
- 计算样本均值： $\bar{\mathbf{x}}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{x}_{ij}$ ，样本协方差： $C_i = \frac{1}{n_i-1} \sum_{j=1}^{n_i} (\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)^T$
- 对样本协方差做特征分解获得： $C_i = U_i \Lambda_i U_i^T$
- 对于给定的 rank = k ，选取 U_i 的前 k 列 $Y_i = U_i(:, 1:k) \in \text{Gr}(d, k)$ 以及 Λ_i 的 k 阶主子式 $R_i^2 = \Lambda_i(1:k, 1:k) \in \mathbb{S}_k^+$ 作为图像集合的 Fixed-Rank PSD 的表示 (Y_i, R_i^2) ，关于该表示具体可参看本文的4.2或参考文献 [8]

通过上述的内容也就为每一个图像集合构造了 Fixed-Rank PSD 矩阵的表示，接下来的部分将借助这种表示对图像集合的分类问题做进一步的探索。

4.3.2 Fixed-Rank PSD 流形用于图像集合分类

在4.3.1一节根据 [8] 中介绍的 Fixed-Rank PSD 矩阵流形的 geometry 结构对图像集合 i 使用了 $(Y_i, R_i^2) \in \text{Gr}(n, k) \times \mathbb{S}_d^+$ ，其实有了这个表示若运用公式4-22定义的 polar metric 即可进行图像集合的分类问题了，但是这样的分类问题过于粗糙，没有包含判别性、large margin 的一些性质也不利于模型的推广（文献 [43]），所以参考 [18,57] 等工作以及工作 [19,32] 关于黎曼流形上正定核的结论，也可以为 Fixed-Rank PSD 矩阵流形定义正定的核的形式来克服以上提到的问题。

关于 Fixed-Rank PSD 流形上（假设 rank = k ）的 polar metric4-22这里还有一些事实需要注意（为了阐述的方便这里将公式4-20和公式4-22糅合在一起后用不同的颜色标出前后两部分）：

$$\begin{cases} g_{(U, R^2)}((\Delta_1, D_1), (\Delta_2, D_2)) \\ \quad = \boxed{\text{tr}(\Delta_1 \Delta_2)} + \lambda \boxed{\text{tr}(R^{-1} D_1 R^{-2} D_2 R^{-1})}, \lambda > 0 \\ \delta^2(A, B) = \boxed{\|\Theta\|_F^2} + \lambda \boxed{\|\log(R_A^{-1} R_B^2 R_A^{-1})\|_F^2} \end{cases} \quad (4-23)$$

其中 $\Delta_i, D_i, i = 1, 2; R_A, R_B, \Theta$ 的定义请参看4.2部分的介绍；对于上式注意到公式4-23中的前半部分实际上是 Grassmann 流形的测地距离 (geodesic distance)，而它与4.1部分介绍的投影度量 (Projection Metric4.7) 之间仅相差一个倍数关系 [43]，所以公式4-23的前半部分可以由投影度量来代替，对于公式4-23的后半部分，注意到这是 SPD 矩阵流形的 AIM[46] 度量，进一步的注意到 R_A, R_B 都是对角矩阵，于是可以得到： $\|\log(R_A^{-1} R_B^2 R_A^{-1})\|_F^2 = 2\|\log(R_A) - \log(R_B)\|^2$ 。

综合以上两点特点 $\delta^2(A, B)$ 可以进一步的形式化为：

$$\begin{aligned}\delta^2(A, B) &= \|Y_A Y_A^T - Y_B Y_B^T\|_F^2 + \lambda \|\log(R_A) - \log(R_B)\|_F^2, \lambda > 0 \\ &= 2k - 2\|Y_A^T Y_B\|_F^2 + \lambda \|\log(R_A) - \log(R_B)\|_F^2\end{aligned}\quad (4-24)$$

其中 Y_A, Y_B 的定义在4.3.1部分已经给出，公式4-24中定义的距离很容易证明在 $\mathbb{S}_d^+(k)$ 上对所有的 $\lambda > 0$ 它都是负定的 [43]，因此容易从4-24出发构造正定的核，表格4.1列出了文章 [43] 中使用的核。

表 4.1 Fixed-Rank PSD 流形中的核

名称	形式化
线性核	$k_l(A, B) = \ Y_A^T Y_B\ _F^2 + \lambda \text{tr}(\log(R_A) \log(R_B))$
多项式核	$k_p(A, B) = (\beta + \ Y_A^T Y_B\ _F^2 + \lambda \text{tr}(\log(R_A) \log(R_B)))^\alpha$
拉普拉斯核	$k_L(A, B) = \exp(-\beta \sqrt{\lambda \ \log(R_A) - \log(R_B)\ _F^2 - 2\ Y_A^T Y_B\ _F^2})$
RBF 核	$k_R(A, B) = \exp(-\beta (\lambda \ \log(R_A) - \log(R_B)\ _F^2 - 2\ Y_A^T Y_B\ _F^2))$

最后文 [43] 中还对比了经过 kDA (kernel Discriminant Analysis) 与不经过 kDA 学习利用最近邻分类的结果；文章最终的实验在手势识别，视频人脸识别和动态纹理识别进行了验证，关于实验的细节以及结果可以从文章 [43] 获得，在本章的实验部分也将对这个方法作进一步的讨论。

4.4 Low-Rank PSD 流形判别学习方法

在4.3.2小结介绍了 Fixed-Rank PSD 流形用于图像集合分类的先驱性工作 [43]；在我
们前期的一些尝试中也做过类似的实验，但是实验中我们发现如下问题：1) 通过特征
分解获得 Fixed-Rank PSD 表示的方法4.3.1比较粗暴，没有考虑其它的信息（如 label）的
利用；2) 虽然工作 [8] 中对 polar metric4-22的形式化过程进行了详细的推导，但是 polar
metric4-22本身割裂了 Y_A, R_A (Grassmann 流形和 SPD 矩阵流形) 之间的联系，更切确的
说是 polar metric4-24仅仅借助加法和一个平衡因子 λ 很难完全刻画 Fixed-Rank PSD 矩
阵 $C_A = Y_A R_A^2 Y_A^T$ 的关系。

针对上述的一些问题本节将会提出我们的改进方案，最后关于方法的验证会在实验
部分给出；这里首先要介绍的部分就是如何为每个图像集合构造更具判别力的 PSD 矩
阵表示。

在 Mu Yadong 发表在的 AAAI'16 的文章 [45] 中，为了在 metric learning 的学习过
程中保证马氏距离中的度量矩阵 M 是半正定的以及为了加速算法，文章在固定秩的矩
阵流形上提出了一种新的 Second Order Riemannian Retraction Operator (因为没有合适的
翻译这里直接使用英文表示；字面翻译：二阶黎曼 Retraction 算子)，文中作者构造性的
使用 $Z_i = W_i W_i^T, W_i = ((C_i^{1/2} + Z) Y_i), Y_i = U_i(:, 1:k)$ 表示 PSD 矩阵，其中 C_i, U_i, Y_i 的定

义同4.3.1部分的定义, Z 是未知的参数, 文章 [45] 通过要求 Z 满足 Fixed-Rank 矩阵流形的切空间中的性质来获得一个好的表示 (详细内容可以参看文献 [45]); 这启示我们在 PSD 矩阵编码图像集合的时候可以借鉴这种形式, 并通过 Z 编码更多的信息 (如编码样本的 label 信息)。

另一方面, 前面已经指出 polar metric⁴⁻²²虽然有很好的性质, 但是其分离得形式使得其无法很好的刻画 $Y_i \in \text{Gr}(n, k), R_i \in \mathbb{S}_k^+$ 之间的联系, 所以需要寻找一种新的度量形式来刻画两者之间的相似度; 为此, 如同文章 [8] 一样, 为此先来回顾一下对称正定 (SPD) 矩阵流形中的不同距离度量 (因为两者有很强的关联), 期望从中获得解决方案:

表 4.2 SPD 矩阵流形上的距离度量

距离度量	形式化	是测地距离	可接受不满秩输入
Affine-Invariant 度量 [46]	$\ \log(X_i^{-\frac{1}{2}} X_j) \log(X_i^{-\frac{1}{2}})\ _F$	✓	✗
Log-Euclidean 度量 [?]	$\ \log(X_i) - \log(X_j)\ _F$	✓	✗
Stein 散度 [51]	$\log \det(\frac{X_1 + X_2}{2}) - \frac{1}{2} \log \det(X_1 X_2)$	✗	✗
Jeffreys 散度 [21]	$\frac{1}{2} \text{tr}(X_1^{-1} X_2 + X_2^{-1} X_1) - d$	✗	✗
Cholesky 距离 [13]	$\ chol(X_1) - chol(X_2)\ _F$	✗	✓
Power-Euclidean 度量 [13]	$\frac{1}{\alpha} \ X_1^\alpha - X_2^\alpha\ _F$	✗	✓

表格4.2中 $chol(\cdot)$ 表示的是 Cholesky 分解。

表格4.2中列出的所有度量均可用于对称正定矩阵的之间距离的度量, 其中 Affine-Invariant 度量和 Log-Euclidean 度量分别在 [46] 和 [?] 被提出, 且它们有各自的黎曼度量也是 \mathbb{S}_d^+ 上的测度距离, 并且 Log-Euclidean 度量以其优于 Affine-Invariant 度量的计算性质而赢得了不少青睐, 但是遗憾的是这两种度量都不能用于不满秩 (也就是半正定的) 的情况, Stein 散度 [51] 和 Jeffreys 散度 [21] 最初是为了提高计算效率而提出来的, 但是由于 Stein 散度需要计算 $\log \det(\cdot)$ Jeffreys 散度需要计算 $X_i^{-1}, i = 1, 2$ 所以也不能用于处理半正定的输入, 最后剩下 Cholesky 距离 [13] 和 Power-Euclidean 度量 [13] 这两种度量由于不涉及求逆或者 $\log(\cdot)$ 操作等, 所以可以直接用于半正定矩阵, 而与 polar metric⁴⁻²²相比, 其直接考虑半正定输入, 没有割裂 $\text{Gr}(d, k), \mathbb{S}_d^+$ 之间的关系, 应该有更好的表示能力; 不过需要注意的是 Cholesky 距离和 Power-Euclidean 度量并不是 \mathbb{S}_d^+ 上的测地距离更不是 $\mathbb{S}_d^+(k)$ 上的测地距离, 因此不能完全刻画 $\mathbb{S}_d^+(k)$ 的 geometry 的结构, 不过本文相信它们的联合的形式的优点能够弥补其不是测地距离的不足 (最后的实验验证了我们的观点)。接下来这里选择 Cholesky 距离和 Power-Euclidean 度量作为半正定矩阵的度量。特别地, 简单的验证试验发现 Power-Euclidean 度量比 Cholesky 距离能够更好的刻画数据本身的性质, 所以本章接下来的部分以 Power-Euclidean 度量进行介绍和实验。

最后还需要注意的是文章 [43] 使用 Fixed-Rank PSD 建模图像集合的方法中, Fixed-

Rank 的性质的约束主要是为了能够在 PSD 上定义流形的 geometry 结构，并利用该 geometry 结构进行判别学习；但是在使用 Power Metric 度量两个 PSD 矩阵之间的关系的时候 Fixed-Rank 的性质却不是那么必要，而此时 Low-Rank 成为更本质的一个要求，因此接下来的内容中我们使用更一般的 Low-Rank PSD 矩阵建模图像集合。

以上针对目前使用 PSD 矩阵建模图像集合的，下面是该方向的进一步细化，大致分为如下的一些内容，首先是如何利用 Power-Euclidean 度量构造一个好的 Low-Rank PSD 的表示，然后是如何利用 Power-Euclidean 度量进行判别学习以及 Low-Rank PSD 矩阵上的判别学习方法的形式化。

4.4.1 融入判别信息的 Low-Rank PSD 矩阵的构造

在4.4节的前半部分提出了借鉴 [45] 中构造 PSD 的内容（公式4-25）：

$$Z_i = W_i W_i^T, W_i = \left((C_i^{1/2} + Z) Y_i \right), Y_i = U_i(:, 1:k), C_i = U_i \Lambda_i U_i^T, R_i^2 = \Lambda \quad (4-25)$$

这里先简单说明一下这种构造方式的合理性：在 $Z_i = \left((C_i^{1/2} + Z) Y_i \right) \left((C_i^{1/2} + Z) Y_i \right)^T$ 中当 $Z = \mathbf{0}$ 时，则 Z_i 等价于文章 [43] 中 Fixed-Rank PSD 矩阵的构造方式：

$$\begin{aligned} Z_i &= \left((C_i^{1/2} + Z) Y_i \right) \left((C_i^{1/2} + Z) Y_i \right)^T \\ &= \left((U_i R_i U_i^T) U_i(:, 1:k) \right) \left((U_i R_i U_i^T) U_i(:, 1:k) \right)^T \\ &= (U_i R_i(:, 1:k)) (U_i R_i(:, 1:k))^T \\ &= (U_i(:, 1:k) \Lambda (1:k, 1:k) U_i(:, 1:k))^T \end{aligned} \quad (4-26)$$

此外，当 $Z = I - C_i^{\frac{1}{2}}$ 时：

$$Z_i = Y_i Y_i^T \quad (4-27)$$

正好是 [18] 中的投影矩阵的结果，由此可以一定程度上说明4-25构造的合理性。

在说明完合理性之后，接下就是如何选择公式4-25中的 Z 的问题，虽然文章 [45] 中给出了一种构造的方式，但是由于目的不同（文章 [45] 中是为了优化马氏距离中的度量矩阵，这里是表示图像集合）所以这里选择另一种矩阵 Z 的构造方式：借助 discriminate learning 学习 Z 将判别信息编码到图像集合的 PSD 表示中。

要把判别信息融入到 Low-Rank PSD 矩阵表示的编码中，一个直接有效的方法就是要求同类的样本更相似而不同类的样本则尽量不相似。为此，利用样本标签 $y_i, i = 1, 2, \dots, n$ 定义两两样本之间的关系（同类或不同类）表示矩阵 $G \in \{-1, 1\}^{n \times n}$ ，其中 G_{ij} 的定义如下：

$$G_{ij} = \begin{cases} 1, & \text{if } y_i = y_j \\ -1, & \text{else} \end{cases} \quad (4-28)$$

其中 $y_i, i = 1, 2, \dots, n$ 表示的是样本的标签。经过简单的变换并利用公式3-5的表示，可以得到 $G = 2YY^T - 1$ 。但是注意到，在实际的判别学习的方法研究中，要求所有的同类样本对的相似度尽量大而不同类样本对的相似度尽量小是不太现实的，而且这样也会增加算法的计算量，所以这里进一步的考虑在在 Graph Embedding[64] 的框架下构造正负样本对：首先定义两个参数 k_w, k_b ，其意义类似于 kNN 分类器中的参数 k ， k_w 描述的是当前样本与同类样本的近邻关系， k_b 描述的是当前样本与不同类的样本的近邻关系，利用 k_w, k_b 定义 G_w, G_b ：

$$G_w = \{G_{ij}^w\}_{n \times n}, \text{ where } G_{ij}^w = \begin{cases} 1, & \text{if } j \text{ is one of } i's \text{ first } k_w \text{ neighbours with same label} \\ 0, & \text{otherwise.} \end{cases}$$

$$G_b = \{G_{ij}^b\}_{n \times n}, \text{ where } G_{ij}^b = \begin{cases} 1, & \text{if } j \text{ is one of } i's \text{ first } k_b \text{ neighbours with different label} \\ 0, & \text{otherwise.} \end{cases}$$
(4-29)

最后，利用 G_w, G_b 对公式4-28中的 G 进行重新定义：

$$G = G_w - G_b \quad (4-30)$$

此外，为了平衡正负样本对之间的比例，实验中还将 G 中的 $1, -1$ 分别除以正负样本对的个数。

接下来，我们使用 $\rho_{ij} = \frac{\langle Z_i, Z_j \rangle_F}{\|Z_i\|_F \|Z_j\|_F}; i, j = 1, 2, \dots, n$ 来度量两个表示之间的相似度^①；利用 ρ_{ij} 和 G_{ij} 的定义，给出公式4-31中的损失函数：

$$C(Z) = - \sum_{i=1}^n \sum_{j=1}^n G_{ij} \rho_{ij}^2 = \text{tr}(GF^T) + \gamma \|Z\|_F^2, \text{ where } F_{ij} = \rho_{ij}^2 \quad (4-31)$$

其中函数 $C(Z)$ 的最后一项是正则项，其目的是为了防止过拟合，这只是一个小小的 trick，实验中我们发现了该 trick 的虽然简单但是却有效。

我们的目标是使最终编码的 $Z_i, i = 1, 2, \dots, n$ 让 $C(Z)$ 最小；这是一个矩阵函数的优化问题，为此需要计算目标函数的导数。为了叙述的方便，公式4-32引入两条在计算矩阵方向导数的时候的规律 (rule)

$$\begin{cases} \text{rule 1 : } D(f \circ g)(X)[H] = Df(g(X))(g(X))[Dg(X)[H]] \\ \text{rule 2 : } D\langle f(X), g(X) \rangle(X)[H] = \langle Df(X)[H], g(X) \rangle + \langle f(X), Dg(X)[H] \rangle \end{cases} \quad (4-32)$$

公式的4-32中的 $Df(\cdot)(Z)[H]$ 表示的是矩阵函数 $f(\cdot)$ 关于变量 Z 在 H 方向上的方向导数，关于矩阵函数的方向导数的内容，读者可以参看文献 [9] 中的内容，此外关于 rule 1, rule 2

^① 注：这里不使用 $\frac{\langle Z_i^{\frac{1}{m}}, Z_j^{\frac{1}{m}} \rangle}{\|Z_i\|_F^{\frac{1}{m}} \|Z_j\|_F^{\frac{1}{m}}}; i, j = 1, 2, \dots, n$ 的原因主要是这会大大增加计算复杂度而且当 $n > 1$ 时 $x^{\frac{1}{n}}$ 的导数未定义，所以这里退而求其次

的内容读者也可以在 [9] 中找到。

对于最小化问题4-31，这里使用共轭梯度算法进行求解，为此需要预先计算 $C(Z)$ 的梯度 $\nabla_Z C(Z)$ ，其中主要的就是计算 $\nabla_Z \rho_{ij}^2$ ，为了方便起见定义 $k_{ij} = \langle Z_i, Z_j \rangle_F$ ，接下来利用公式4-33对 $C(Z)$ 计算其关于 Z 的导数

$$\begin{aligned} C(Z) &= \text{tr}(GF^T) = \sum_{i=1}^n \sum_{j=1}^n G_{ij} F_{ij} = \sum_{i=1}^n \sum_{j=1}^n G_{ij} \frac{k_{ij}^2}{k_{ii} k_{jj}} + \gamma \|Z\|_F^2 \\ \frac{\partial}{\partial Z} C(Z) &= \sum_{i=1}^n \sum_{j=1}^n G_{ij} \left(c_1 \frac{\partial}{\partial Z} k_{ij} - c_2 \frac{\partial}{\partial Z} k_{ii} - c_3 \frac{\partial}{\partial Z} k_{jj} \right) + 2Z \quad (4-33) \\ \text{where } c_1 &= \frac{2k_{ij}k_{ii}k_{jj}}{(k_{ii}k_{jj})^2}, c_2 = \frac{k_{ij}k_{ij}k_{jj}}{(k_{ii}k_{jj})^2}, c_3 = \frac{k_{ij}k_{ij}k_{ii}}{(k_{ii}k_{jj})^2} \end{aligned}$$

为了计算4-33的结果，主要需要计算的是 $\frac{\partial}{\partial Z} k_{ij}$ ；利用论文 [9] 5.3 节相关的内容推导 $\frac{\partial}{\partial Z} k_{ij}$ 的相关形式如下（这里选择从方向导数出发）：

$$\begin{aligned} Dk_{ij}(Z)[H] &= D \langle Z_i, Z_j \rangle_F \\ &= \langle DZ_i(Z)[H], Z_j \rangle_F + \langle Z_i, DZ_j(Z)[H] \rangle_F \end{aligned} \quad (4-34)$$

由于 $\langle DZ_i(Z)[H], Z_j \rangle_F$ 与 $\langle Z_i, DZ_j(Z)[H] \rangle_F$ 的计算是类似的，所以仅以其中的一部分作为研究对象，其结果可以很好的平移到另一部分：

$$\begin{aligned} \langle DZ_i(Z)[H], Z_j \rangle_F &= \left\langle D \left((C_i^{\frac{1}{2}} + Z) S_i (C_i^{\frac{1}{2}} + Z)^T \right) (Z)[H], (C_j^{\frac{1}{2}} + Z) S_j (C_j^{\frac{1}{2}} + Z)^T \right\rangle_F \\ D \left((C_i^{\frac{1}{2}} + Z) S_i (C_i^{\frac{1}{2}} + Z)^T \right) (Z)[H] &= D(C_i^{\frac{1}{2}} S_i C_i^{\frac{T}{2}} + C_i^{\frac{1}{2}} S_i Z^T + Z S_i C_i^{\frac{T}{2}} + Z S_i Z^T)(Z)[H] \\ &= C_i^{\frac{1}{2}} S_i H^T + H S_i C_i^{\frac{T}{2}} + H S_i Z^T + Z S_i H^T \\ &= (C_i^{\frac{1}{2}} + Z) S_i H^T + H S_i (C_i^{\frac{T}{2}} + Z^T) \end{aligned} \quad (4-35)$$

其中 $S_i = Y_i Y_i^T, i = 1, 2, \dots, n$ ，接下来利用公式4-35的结果，得到公式4-36的结果（其中利用了 $Z_i, S_i; i = 1, 2, \dots, n$ 是对称矩阵的结果）。

$$\left\{ \begin{aligned} \langle DZ_i(Z)[H], Z_j \rangle_F &= \left\langle (C_i^{\frac{1}{2}} + Z) S_i H^T + H S_i (C_i^{\frac{T}{2}} + Z^T), Z_j \right\rangle_F \\ &= \text{tr} \left((C_i^{\frac{1}{2}} + Z) S_i H^T Z_j \right) + \text{tr} \left(H S_i (C_i^{\frac{T}{2}} + Z^T) Z_j \right) \\ &= \text{tr} \left(H^T Z_j (C_i^{\frac{1}{2}} + Z) S_i \right) + \text{tr} \left(H S_i (C_i^{\frac{T}{2}} + Z^T) Z_j \right) \\ &= 2 \text{tr} \left(H^T Z_j (C_i^{\frac{1}{2}} + Z) S_i \right) = 2 \left\langle H, Z_j (C_i^{\frac{1}{2}} + Z) S_i \right\rangle_F \quad (4-36) \\ Dk_{ij}(Z)[H] &= 2 \left\langle H, Z_j (C_i^{\frac{1}{2}} + Z) S_i \right\rangle_F + 2 \left\langle H, Z_i (C_j^{\frac{1}{2}} + Z) S_j \right\rangle_F \\ \frac{\partial}{\partial Z} k_{ij} &= 2 Z_j (C_i^{\frac{1}{2}} + Z) S_i + 2 Z_i (C_j^{\frac{1}{2}} + Z) S_j \\ &= 2 (Z_j W_i Y_i^T + Z_i W_j Y_j^T) \end{aligned} \right.$$

最后结合公式4-33和4-36的结果，即可方便的计算出 $C(Z)$ 的梯度，将其作为共轭梯度算法的输入，最小化 $C(Z)$ 获得 Z^* 即可用于对图像集合的 PSD 编码的形式。

4.4.2 Low-Rank PSD 矩阵集合上的判别学习方法

在4.4.1小节中介绍的如何用 Fixed-Rank PSD 矩阵编码图像集合的问题，但是仅仅有了带判别的编码还是不足以处理 Max margin 等问题，类似于 [43] 中的方法，这里选择在 KDA[7] 的框架下进行判别学习，特别地注意到 Power Metric 的定义可以由 $k_{ij} = \left\langle Z_i^{\frac{1}{m}}, Z_j^{\frac{1}{m}} \right\rangle_F$ 的 kernel 的形式导出，并且容易证明 $K = \{k_{ij}\}_{n \times n}$ 的正定性，所以 K 是正定核（关于正定性的证明，这里就不再赘述了）。

由于 KDA[7] 的研究已经非常的成熟，而且这里也不是对 KDA 的改进或者相关的工作，所以这里不会从头再把 KDA 的框架再介绍一遍，而只会简单的回顾一下 KDA 的基本思想。

首先，LDA(Linear Discriminant Analysis) 的目标是使得内类散度小而类间散度大，而 KDA 则是 LDA 在再生核希尔伯特空间 (RKHS: reproducing kernel Hilbert space) 中的版本，其目的也是一样的。文献 [7] 中给出了十分简洁的 Kernel Discriminant Analysis 的形式：设 $n_c, c = 1, 2, \dots, C$ 表示各个类别中的样本数，其中 C 表示的是类别数， $\sum_{c=1}^C n_c = n$ ，这里的 n 表示的是样本总数， $\phi(\cdot)$ 表示的是非线性变化（例如接下来将要使用的 $\phi(Z_i) = Z_i^{\frac{1}{n}}, n > 1$ ），利用内积的形式定义核 $K = \{k_{ij}\}_{n \times n}, k_{ij} = \left\langle \phi(Z_i), \phi(Z_j) \right\rangle_F$ ，则 KDA 的目标形式化为公式4-37：

$$\alpha_{opt} = \arg \max \frac{\alpha^T K W K \alpha}{\alpha^T K K \alpha} \quad (4-37)$$

其中 W 的定义如公式4-38所示：

$$W = \{W_{ij}\}_{n \times n}, \text{ where } W_{ij} = \begin{cases} \frac{1}{n_c}, & \text{if } Z_i, Z_j \text{ are both in the } c\text{-th class} \\ 0, & \text{otherwise.} \end{cases} \quad (4-38)$$

本节简单的介绍了 KDA 的框架，试验中我们使用了文献 [? ?] 提供的代码。

4.5 实验结果与分析

本章所做的问题是使用 Low-Rank PSD 矩阵表示集合数据，并在该表示下进行集合数据的分类问题，任务与第二章的任务类似，都是集合数据的分类问题所以这里使用了相同的数据进行实验（物体识别数据库 ETH[?]，材料识别数据库 UIUC[39] 以及视频人脸识别数据库 [36]），这几个数据库上的实验任务已经在3.4部分做了相对细致的介绍，同时数据库的规模（包含了小数据库 ETH，中等规模数据库 UIUC，较大规模的数据库 YTC）也具有一定的代表性；各个数据集合上的基本特征的提取与3.4.1节介绍的方式相同，这里再赘述；在获得原始特征之后3-30构造初始分半正定矩阵表示（公式4-25中的

C_i), 后面的编码过程在4.4.1节中描述。最后关于数据库的详细介绍以及对应的测试协议可以参看1.4部分的内容, 而关于原始图像特征的提取部分的内容则可以参看3.4.1节的内容; 最后表4.3给出了实验结果。

表 4.3 Fixed Rank PSD 流形判别学习算法实验结果

数据集 方法	ETH80	UIUC	YTC
GDA[18]	92.50±3.54	53.33±2.32	66.73±3.16
CDL-PLS[57]	93.25±4.72	53.89±4.06	70.28±2.13
RSR-Stein[23]	93.25±3.34	52.41±4.03	72.77±2.69
SPDML-Stein[22]	90.50±3.87	49.17±2.37	61.57±3.43
SPDML-AIM[22]	90.75±3.34	48.09±1.82	64.66±2.92
LEML[30]	94.75±2.49	48.98±3.69	70.53±2.95
FRPSD – KDA _{Linear} [43]	94.50±3.07	52.04±3.80	70.93±3.28
FRPSD – KDA _{Polynomial} [43]	96.00±2.42	56.02±3.93	70.74±3.05
FRPSD – KDA _{Laplace} [43]	95.50±2.84	57.69±3.35	70.14±3.04
FRPSD – KDA _{RBF} [43]	95.50±3.50	57.78±4.10	70.96±3.05
LRPSD – KDA	93.25±4.72	59.17±3.48	74.37±2.96
LRPSD – KDA _{discrim}	94.50±2.48	59.81±3.58	74.73±2.91

其中, FRPSD – KDA 是文章 [43] 中方法的简称 (是 Fixed-Rank PSD 和 KDA 的缩写), 其下边表示了核的类型: 线性核 (*linear*)、多项式核 (*polynomial*)、Laplace 核 (*Laplace*) 以及 RBF 核 (*RBF*), LRPDS – KDA 则表示本章所提的方法的简称 (是 Low-Rank PSD 和 KDA 的缩写), 其中下标 *discrim* 表示的方法是否使用4.4.1部分的编码学习方式构造图像集合的 Low-Rank PSD 矩阵表示。

表格4.3中选取的方法与3.4.3部分选择方法类似, 包含了目前取得 state-of-the-art 的一些主要方法, 此外还包含了与我们最相关的工作 [43] 中的方法的结果 (由于作者没有公布源代码, 所以我们小心的实现并进行细致的参数调节之后汇报我们所获得的最好的结果) 以及 GDA[18] 的结果 (该方法也是自己实现)。所有对比的方法都是小心调参之后汇报的最好的结果。

从表格4.3我们的可以得到的信息是: 本章所提的 LRPDS – KDA 方法在三个数据集上获得了与 state-of-the-art 可比甚至是更好的性能, 相较于子空间和协方差统计量 (表格4.3的前两行), LRPDS – KDA 方法在三个任务上都有相对明显的提升。我们认为获得这种提升主要有两个方面: 1) 通过与文章 [43] 中的方法 (表中的 FRPSD – KDA_{Linear}一类方法) 相比, 可得出本文使用的 Power metric 更能刻画 PSD 矩阵的性质的结论, 这与我们一开始的想法是吻合的; 2) 表中的最后两行相比可看出融入判别信息带来提升, 虽然幅度不大 (这与本文选择的相对简单的融合方式有关) 但是可以看出方向的正确性, 这里还有上升的空间。

实验中我们还发现：公式4-31中的正则项 $\gamma \|Z\|_F^2$ 对结果的影响与数据集相关，对于每个集合中数据较少或噪声较大的数据集（ETH 和 UIUC）该项的设置很重要，但是对于每个集合中样本较多的数据集该项则可忽略掉（如：在 UIUC 上我们设置 $\gamma = 0$ ），其它实验参数的设置主要包含4-29够造中 k_w, k_b 的设置，ower metric 中 n （或 $\alpha = \frac{1}{n}$ ）的设置，以及 Low-Rank 约束的上界 k 的设置。由于都是整数设置方法比较常规这里就不再一一赘述。

最后，需要注意的是利用 power metric 还可以定义其它的核的形式，表4.4中给出了 kernel 的形式：

表 4.4 Power Metric 相关的核

名称	形式化
线性核	$k_l(A, B) = \text{tr}\left(Z_i^{\frac{1}{m}} Z_j^{\frac{T}{m}}\right)$
多项式核	$k_p(A, B) = \left(\beta + \text{tr}\left(Z_i^{\frac{1}{m}} Z_j^{\frac{T}{m}}\right)\right)^{\alpha}$
拉普拉斯核	$k_L(A, B) = \exp\left(-\beta \sqrt{\ Z_i^{\frac{1}{m}} Z_j^{\frac{T}{m}}\ _F^2}\right)$
RBF 核	$k_R(A, B) = \exp\left(-\beta \ Z_i^{\frac{1}{m}} Z_j^{\frac{T}{m}}\ _F^2\right)$

但是从实验结果中我们不难得到的看出的是在线性核下我们已经得到了与 state-of-the-art 可比甚至是更好一些的结果，此外 Kernel 的方法的引入虽然会对最终的结果有所提高（部分测试试验中发现 Laplace Kernel 对于 Power Metric 有更好的促进作用），但是由于引入了更多的参数需要调节，所以使得算法的实际运用价值打了折扣；因此这里仅把 kernel 的方法作为一个未来深入方向，而不在这里做深入讨论。

4.6 总结与下一步工作

在本章中我们针对对集合数据的建模问题以及集合数据特征本身存在的一些问题，使用 PSD 矩阵建模图像集合的问题，并且集合前期关于 Fixed-Rank PSD 流形的研究以及工作 [43] 的内容，对存在的问题以及 PSD 矩阵建模图像集合的问题，提出了 Low-Rank PSD 矩阵的判别学习方法，主要的内容可以归纳为如下几点：1) 使用带有判别性的低秩的 PSD 的矩阵表示图像集合；2) 针对文献 [43] 中的使用的 polar metric 割裂了 U, R 之间关系的问题提出了使用 power metric 进行判别学习的方法；3) 在 KDA[7] 的框架下进行判别学习获得与 state-of-the-art 可比甚至是更好的结果。

最后，前面以及提到表格4.3中的结果显示判别学习与非判别学习的结果提升不是特别的明显，究其原因的话可能是 Graph Embedding 的框架与最后的 KDA 的框架没有很好的适配的原因，这里的问题值得深入研究，此外就是前面提到的跟多的 Kernel 版本的方法，可能也是一个可以尝试的方向。

第五章 结束语

计算视觉问题的研究经过几十年的研究，取得了巨大的成就。在计算机视觉中，集合数据的研究虽然只有十多年但是已然成为视觉任务中的一个热点，其中集合数据主要用图像集合这样一个概念来描述，它有可能是视频、物体的多视角图片、主题相册等。本文的内容主要是针对这样一种集合数据的建模和对应模型下的判别学习方法。

经过 10 多年的发展，根据图像集合的表示方式的不同，图像集合分类问题相关方法逐渐形成了以下的一些类别：1) 流形和子空间的方法，2) 仿射包相关的方法，3) 统计建模的方法，4) 深度学习的方法，5) 字典学习/稀疏编码的方法等。其中统计建模的方法以其强大的信息编码能力以及简洁的模型表示逐渐发展成为集合数据研究的主要方法之一，同时也由于统计模型的特殊表现形式而需要引入如黎曼流形这样的数学工具对该这样一些模型进行研究。而本文也就是在这样的背景下所进行的集合数据建模以及非线性数据结构的判别学习方法的研究。

5.1 本文工作总结

本文的工作主要围绕的是集合数据的表示和判别学习展开的。首先，作为基础本文在第二章中探讨了矩阵函数的相关问题，并结合学位论文课题中提炼出的相关实例对矩阵流形优化做进行介绍；然后针对使用对称正定矩阵建模图像集合的方法（从最初的协方差矩阵建模图像集合，到后来的高斯模型表示再到最近的 GMM 模型建模都可以用对称正定矩阵表示）中缺少偏最小二乘的回归这样一个强有力的数据分析工具的问题，本文在第三章中提出了黎曼流形上的多切空间逐步回归的偏最小二乘回归方法，并把它用于集合数据的分类问题中。最后，针对二阶统计量表示数据维度过高，存在不满秩和样本稀少带来的估计不准等问题以及子空间建模没有利用尺度信息（特征值）的问题。接下来依次对前几章的内容进行总结说明。

第二章中我们围绕矩阵函数和流形优化问题进行介绍。在对矩阵函数，流形等基本概念介绍的基础上，针对矩阵流形上的优化问题进行讨论与探究，并结合着从研究生学位论文课题中提炼出的相关实例对矩阵流形优化进行介绍，一方面最终希望帮助读者理解并复现本文提出的方法和结论，另一方面也为解决类似流形优化问题提供借鉴。

第三章在统计模型建模集合数据的大背景下，以黎曼流形为研究工具结合已有工作 [14,35] 研究了黎曼流形空间中的偏最小二乘问题。该问题的研究中首先参考了 [14,35] 的工作将欧式空间中的投影的概念泛化到了黎曼流形空间，并借此定义了黎曼流形上的偏最小二乘的基本版本。后注意到图像集合问题与 DTI(Diffusion Tensor Image) 研究问题的不同（主要是前者的数据分布更分散也更稀疏）本章在基本版本的基础上提出了偏

最小二乘逐步回归的方案，在流形的多个切空间进行偏最小二乘问题的学习，并利用逐步回归的思想将学习的结果整合起来；最后以非奇异协方差矩阵即对称正定矩阵黎曼流形为实例，在图像集合问题上实验证明了该方法的有效性，此外文章提出的逐步回归的方案是一个通用的方案，该方案对于其它类型的 SPD 矩阵流形的表示（如在 UIUC[39] 上使用的 Region Covariance 的表示）也是可用的。

第四章的内容是从图像集合的表示的问题的角度出发进行研究的，考虑到使用协方差建模图像集合的时候协方差表示不满秩以及特征表示维度过高 $d \times d$ 等问题以及子空间建模没有利用尺度信息（特征值）的问题。提出了用低秩半正定矩阵建模图像集合的方法，并针对 [43] 中使用固定秩的对称半正定矩阵（Fixed-Rank PSD 矩阵）流形建模图像集合中固定秩对称半正定矩阵获得方式简单以及 polar metric⁴⁻²²割裂了 Grassmann Manifold 和 SPD 矩阵流形之间的关系的问题提出编码了判别信息的低秩半正定矩阵建模图像集合的方法，并实验证了该问题，得到了与目前的 state-of-the-art 可比甚至是稍好一些的结果。

本文的中的内容由图像集合问题衍生出来，更偏向于基础理论。探索了集合数据的建模表示以及非线性结构表示下的判别学习的问题；在此过程中温习原有知识的同时对新的知识也有了更加深入的理解，尤其是矩阵函数相关的问题以及流形上的优化问题，从中也获益匪浅。另一方面，这些探索工作也是实验室原有研究方向的延展，期间的尝试有成功也有失败，成功地方希望能够为后来的读者起到参考的意义，而失败的地方也希望能读者能够引以为戒。以上是本文的工作总结，接下来会再用一小部分的空间对本文中的工作进行反思与未来可能方向的讨论。

5.2 反思与讨论

本节是对本文中介绍的工作的反思和讨论，主要与第三和第四章的内容相关；思考目前存在的一些不足与困扰；并期望能对读者有所帮助，以下是具体内容。

在第三章中为了克服基础版本的黎曼流形上的偏最小二乘回归的问题提出了使用逐步回归的方法在多个切空间中进行逐步回归学习的方法，这虽然带来了性能上的提升，但是方法中的切空间的个数以及切空间中投影方向的数目的选择是两个与算法性能直接相关的参数，究其原因主要是逐步回归的方案的抗过拟合的能力不足导致以上两个参数过大则出现过拟合而过小又会出现欠拟合的问题。因此后续需要考虑更好的多切空间模型融合（整合）的方法，在以往的工作中 [53] 给出了一个很好的启示：使用 Adaboost 的框架来融合多模型，这样就可以有效的控制训练和测试的误差了。

在第四章中我们已经指出，虽然提出了使用带判别信息的低秩半正定矩阵来编码 label 信息，但是编码的框架使用的是相对比较直接的 Graph Embedding 的形式来编码以及编码过程中由于函数 $x^{\frac{1}{n}}, n > 1$ 在 0 点的导数未定义问题使得编码过程中的 metric 与 KDA 中 metric 不一致也难免会对最后的结果造成一定的损失。这部分要求针对 PSD 矩

阵寻找更合适的度量或者散度（如 Bregman Divergence）来代替现在的 Power Metric。

最后，注意到 Deep Learning 已经在各个领域都取得不小的进展，而在图像集合分类问题中也有一些有益的尝试（如：[24] 和 [41]），这些都是初步的尝试，并没有深入的针对图像集合问题进行研究，所以这一部份对接下来的图像集合问题的研究应该是意义重大的。另一个重要的方向是图像集合与静态图像的匹配/分类的问题，这是一个很有前景的运用方向，在检索，追逃等领域的意义会更重大。

参考文献

- [1] 黄智武. 黎曼度量学习极其在视频人脸识别中的应用研究. 博士学位论文, 北京: 中国科学院研究生院, 2015.
- [2] 梅加强. 流形与几何初步. 科学出版社, 2013.
- [3] P. A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.
- [4] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2007.
- [5] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM journal on matrix analysis and applications*, 29(1):328–347, 2007.
- [6] Matthew Barker and William Rayens. Partial least squares for discrimination. *Journal of chemometrics*, 17(3):166–173, 2003.
- [7] Gaston Baudat and Fatiha Anouar. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12(10):2385–2404, 2000.
- [8] Silvere Bonnabel and Rodolphe Sepulchre. Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1055–1070, 2009.
- [9] Nicolas Boumal, Pierre-Antoine Absil, et al. Discrete curve fitting on manifolds. In *30th Benelux Meeting on Systems and Control*, 2011.
- [10] Nicolas Boumal, Bamdev Mishra, P-A Absil, and Rodolphe Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014.
- [11] Hakan Cevikalp and Bill Triggs. Face recognition based on image sets. In *Computer Vision and Pattern Recognition*, pages 2567–2573. IEEE, 2010.
- [12] Yi-Chen Chen, Vishal M Patel, P Jonathon Phillips, and Rama Chellappa. Dictionary-based face recognition from video. In *European Conference On Computer Vision*, pages 766–779. Springer, 2012.
- [13] Ian L Dryden, Alexey Koloydenko, and Diwei Zhou. Non-euclidean statistics for covariance matrices, with applications to diffusion tensor imaging. *The Annals of Applied Statistics*, pages 1102–1123, 2009.
- [14] P Thomas Fletcher, Conglin Lu, Stephen M Pizer, and Sarang Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, 23(8):995–1005, 2004.
- [15] Reeves Fletcher and Colin M Reeves. Function minimization by conjugate gradients. *The computer journal*, 7(2):149–154, 1964.
- [16] Ralph Gross and Shi Jianbo. The cmu motion of body (mobo) database. Technical report, 2001.
- [17] William W. Hager and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization*, 2(1):35–58, 2006.

- [18] Jihun Hamm and Daniel D Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th international conference on Machine learning*, pages 376–383. ACM, 2008.
- [19] Mehrtash Harandi, Richard Hartley, Chunhua Shen, Brian Lovell, and Conrad Sanderson. Extrinsic methods for coding and dictionary learning on grassmann manifolds. *International Journal of Computer Vision*, 114(2-3):113–136, 2015.
- [20] Mehrtash Harandi, Mathieu Salzmann, and Mahsa Baktashmotagh. Beyond gauss: Image-set matching on the riemannian manifold of pdfs. *International Conference on Computer Vision*, 2015.
- [21] Mehrtash Harandi, Mathieu Salzmann, and Fatih Porikli. Bregman divergences for infinite dimensional covariance matrices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1003–1010, 2014.
- [22] Mehrtash T Harandi, Mathieu Salzmann, and Richard Hartley. From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices. In *European Conference On Computer Vision*, pages 17–32. Springer, 2014.
- [23] Mehrtash T Harandi, Conrad Sanderson, Richard Hartley, and Brian C Lovell. Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. In *Computer Vision–ECCV 2012*, pages 216–229. Springer, 2012.
- [24] Munawar Hayat, Mohammed Bennamoun, and Senjian An. Deep reconstruction models for image set classification. *Pattern Analysis and Machine Intelligence*, 37(4):713–727, 2015.
- [25] Agnar Höskuldsson. Pls regression methods. *Journal of chemometrics*, (2):211–228, 1988.
- [26] Yiqun Hu, Ajmal S Mian, and Robyn Owens. Sparse approximated nearest points for image set classification. In *Computer vision and pattern recognition*, pages 121–128. IEEE, 2011.
- [27] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.
- [28] Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Learning euclidean-to-riemannian metric for point-to-set classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1677–1684, 2014.
- [29] Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Hybrid euclidean-and-riemannian metric learning for image set classification. In *Asian Conference On Computer Vision*, pages 562–577. Springer, 2015.
- [30] Zhiwu Huang, Ruiping Wang, Shiguang Shan, Xianqiu Li, and Xilin Chen. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 720–729, 2015.
- [31] H.Wold. Path models with latent variables: The nipals approach. In *International perspectives on mathematical and statistical model building*, pages 307–357. Academic Press, 1975.
- [32] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. Kernel methods on riemannian manifolds with gaussian rbf kernels. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(12):2464–2477, 2015.
- [33] Michel Journée, Francis Bach, P-A Absil, and Rodolphe Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

- [34] Daniel Karrasch. An introduction to grassmann manifolds and their matrix representation. 2014.
- [35] Hyunwoo J Kim, Nagesh Adluru, Barbara B Bendlin, Sterling C Johnson, Baba C Vemuri, and Vikas Singh. Canonical correlation analysis on riemannian manifolds and its applications. In *European Conference On Computer Vision*, pages 251–267. Springer, 2014.
- [36] Minyoung Kim, Sanjiv Kumar, Vladimir Pavlovic, and Henry Rowley. Face tracking and recognition with visual constraints in real-world videos. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- [37] Bastian Leibe and Bernt Schiele. Analyzing appearance and contour based methods for object categorization. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–409. IEEE, 2003.
- [38] Yan Li, Ruiping Wang, Zhiwu Huang, Shiguang Shan, and Xilin Chen. Face video retrieval with image query via hashing across euclidean space and riemannian manifold. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4758–4767, 2015.
- [39] Zicheng Liao, Jason Rock, Yang Wang, and David Forsyth. Non-parametric filtering for geometric detail extraction and material representation. In *CVPR*, 2013.
- [40] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [41] Jiwen Lu, Gang Wang, Weihong Deng, Pierre Moulin, and Jie Zhou. Multi-manifold deep metric learning for image set classification. In *Conference on Computer Vision and Pattern Recognition*, pages 1137–1145, 2015.
- [42] Jiwen Lu, Gang Wang, and Pierre Moulin. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *International Conference on Computer Vision*, pages 329–336. IEEE, 2013.
- [43] M. Harandi M. Faraki and F. Porikli. Image set classification by symmetric positive semi-definite matrices. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.
- [44] Gilles Meyer, Silvère Bonnabel, and Rodolphe Sepulchre. Regression on fixed-rank positive semidefinite matrices: a riemannian approach. *The Journal of Machine Learning Research*, 12:593–625, 2011.
- [45] Yadong Mu. Fixed-rank supervised metric learning on riemannian manifold. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016.
- [46] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.
- [47] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008.
- [48] Roman Rosipal and Nicole Krämer. Overview and recent advances in partial least squares. In *Subspace, latent structure and feature selection*, pages 34–51. Springer, 2006.
- [49] Roman Rosipal and Leonard J Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *The Journal of Machine Learning Research*, 2:97–123, 2002.
- [50] Roman Rosipal and Leonard J Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *The Journal of Machine Learning Research*, 2:97–123, 2002.
- [51] Suvrit Sra. A new metric on the manifold of kernel matrices with application to matrix geometric means. In *Advances in Neural Information Processing Systems*, pages 144–152, 2012.

- [52] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *European Conference On Computer Vision*, pages 589–600. Springer, 2006.
- [53] Oncel Tuzel, Fatih Porikli, and Peter Meer. Pedestrian detection via classification on riemannian manifolds. *Pattern Analysis and Machine Intelligence*, 30(10):1713–1727, 2008.
- [54] Raviteja Vemulapalli, Jaishanker Pillai, and Rama Chellappa. Kernel learning for extrinsic classification of manifold features. In *Computer Vision and Pattern Recognition*, pages 1782–1789, 2013.
- [55] Hrishikesh D Vinod. Canonical ridge and econometrics of joint production. *Journal of econometrics*, 4(2):147–166, 1976.
- [56] Ruiping Wang and Xilin Chen. Manifold discriminant analysis. In *Computer Vision and Pattern Recognition*, pages 429–436. IEEE, 2009.
- [57] Ruiping Wang, Huimin Guo, Larry S Davis, and Qionghai Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Computer Vision and Pattern Recognition*, pages 2496–2503. IEEE, 2012.
- [58] Ruiping Wang, Shiguang Shan, Xilin Chen, and Wen Gao. Manifold-manifold distance with application to face recognition based on image set. In *Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [59] Wen Wang, Ruiping Wang, Zhiwu Huang, Shiguang Shan, and Xilin Chen. Discriminant analysis on riemannian manifold of gaussian distributions for face recognition with image sets. In *Computer Vision and Pattern Recognition*, pages 2048–2057, 2015.
- [60] Wen Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Probabilistic nearest neighbor search for robust classification of face image sets. In *Automatic Face and Gesture Recognition*, pages 1–7. IEEE, 2015.
- [61] Svante Wold, Michael Sjöström, and Lennart Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, 58(2):109–130, 2001.
- [62] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *in Proc. IEEE Conference. Computer Vision and Pattern Recognition*, 2011.
- [63] Osamu Yamaguchi, Kazuhiro Fukui, and Ken-ichi Maeda. Face recognition using temporal image sequence. In *Automatic Face and Gesture Recognition*, pages 318–323. IEEE, 1998.
- [64] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):40–51, 2007.
- [65] Meng Yang, Pengfei Zhu, Luc Van Gool, and Lei Zhang. Face recognition based on regularized nearest points between image sets. In *Automatic Face and Gesture Recognition*, pages 1–7. IEEE, 2013.
- [66] Florian Yger and Masashi Sugiyama. Supervised logeuclidean metric learning for symmetric positive definite matrices. *arXiv preprint arXiv:1502.03505*, 2015.
- [67] Pengfei Zhu, Lei Zhang, Wangmeng Zuo, and David Zhang. From point to set: Extend the learning of distance metrics. In *International Conference on Computer Vision*, pages 2664–2671. IEEE, 2013.
- [68] Pengfei Zhu, Wangmeng Zuo, Lei Zhang, Simon Chi-Keung Shiu, and Dejing Zhang. Image set-based collaborative representation for face recognition. *Information Forensics and Security*, 9(7):1120–1132, 2014.

致 谢

三年的研究生生涯现在走到了尾声，想想当初研究生录用时的喜悦好像又是不久之前的事；研究生三年的时间里获益良多，值此论文付梓之际，希望向所有帮助，支持过我的老师，同学，朋友以及家人表示由衷的感谢。

衷心感谢导师陈熙霖教授将我带入计算所的大门，并为我们的研究与工作提供了高标准的环境。他的指导为我们指出了前进的方向；同时陈老师既是良师也是益友，不仅给予我们学习和研究上的指导，在日常生活中也给予了我们极大的帮助。是陈老师将我带入了计算机视觉的领域，并在这里接触到世界上计算机视觉前沿的研究与工作，开拓了自己的眼界，也让自己的数学背景得以发挥作用；此外，陈老师对科研的热情以及对生多的态度也在潜移默化中改变着自己，他的言传身教将使我终生受益。

诚挚感谢山世光教授的包容与指导；在人脸组的时间，山老师的言传身教给每一位人脸组的同学以极大的鼓舞，山老师的问题往往能一语中的，让人在交谈中豁然开朗；同时山老师对于计算机视觉这个领域的理解和见地也指导着我们的研究与工作，帮助我们拨去心中的疑惑；山老师对别人的包容与理解也给了我们极大的宽慰和鼓舞。山老师以其自身的博学多识，丰富的阅历以及对问题的独到的见解和眼光吸引了一大批优秀的人才；这些优良的品质也是我们学习的榜样和楷模。

由衷的感谢王瑞平副教授的悉心指导帮助，不管是在生活还是在学习研究上，王老师都给予了我极大的帮助与指导；正是在王老师的指导下我进入本文的主要研究课题，在与王老师的讨论中他对计算机视觉的热情，对于研究的严谨态度以及对于问题的独到的见解都深深的影响着我，让我快速定位问题解决问题的同时也能从问题中获得启示帮助其它研究的推进；同时，王老师对于大方向的把握，长远的目光以及坚定的信念在折服我们同时为我们的研究工作指明了方向为我们坚定了前进的信念。在生活中王老师亦师亦友，竭尽所能地帮助学生，鼓励学生并且不失幽默风趣，给人一种平易近人的感觉，所以与王老师的相处十分愉快。在科研上，王老师的科研热情和态度，严谨的行事风格以及对于问题的独到见解等都是我们学习的榜样，生活上，王老师以其独特的个人魅力吸引着身边的人，让人愿意与他一起共事。

还有很多需要感谢的老师。感谢黄庆明老师，常虹老师，蒋树强老师，苗军老师，蔡秀娟老师，韩琥老师，卿来云老师的教导与解惑，他们的宝贵意见我将终身受用；他们的丰硕的科研成果也让我钦佩万分并给我的研究工作的开展作了重要的启示。感谢实验室办公室的王小彪老师，感谢胡兰平，蔡光辉老师，正是他们的辛勤工作为实验室提供舒适的工作环境，为我们解决了后顾之忧。感谢研究生部周世佳老师，李丹老师，宋守礼老师，张平老师，冯刚老师，李琳老师的默默付出，为我的入学，开题，中期，答

辩，就业提供了极大的帮助。

此外还有很多师兄师姐需要感谢，感谢李岩师兄在我刚到实验室的时候帮助迷茫的我排忧解难，他的悉心指导与帮助我度过迷茫的时期。感谢黄智武师兄在研究工作中的指导和帮助，在他的指导和帮助下我得以相对快速的进入研究工作中，并帮助我回到研究的正轨上来。感谢王琪师兄，阚美娜师姐在 Intel 的凝视矫正项目中的理解悉心指导和建议以及在平时生活与工作中的帮助，让我在工作与研究中找到平衡并从中学习了做事的方法明白了做研究与做项目的区别。感谢李绍欣师兄，刘昕师兄，王雯师姐，尹芳师姐，刘梦怡师姐，王汉杰师兄，张杰师兄，梁孔明师兄，林宇舜师兄，方正鹏师兄，刘文献师兄，谢广志师兄在我遇到问题时无私的提供帮助。

感谢与我同届的刘昊淼，姜华杰，李振林，李健超，吕雄，邬书哲，邓雪松，叶明全，尹肖贻，张川，许震，杨世杰，王智一，付晓慧几位同学，与他们一起度过了百味的研究生的时光，同他们的交流让我获益匪浅。也要感谢实验室的师弟师妹卢宇衡，乔师师，吴望龙，徐梓宁，何建锋，张梦茹，王芳给实验室注入活力带来了欢乐，也让我反思自身。

感谢 UCASTHESIS 的作者朝鲁的无私分享，UCASTHESIS 的存在让我的论文写作轻松自在了许多；感谢 Intel 的刘伟，汤振宇，孙忆晨，郭林楠在我参与 Intel 凝视矫正项目期间的支持与帮助，从这个项目中学到了很多。

最后，感谢我的家人与朋友是你们在我背后默默的支持着我；虽然求学期间我们聚少离多，但是这并不影响我们之间的关系，正是你们的支持与关怀才让我走到现在，再多的话语也无法表达对你们的感激之情，感谢你们为我所做的一切，我也将竭尽所能回应你们。

作者简介

姓名：李显求 性别：男 出生日期：1990.12.27 籍贯：贵州省兴义市

2013.9 – 现在，中国科学院计算技术所，计算机应用技术专业，硕士

2009.9 – 2013.7，华中科技大学（武汉），统计学专业，本科

【攻读硕士学位期间发表的论文】

- [1] Zhiwu Huang, Ruiping Wang, Shiguang Shan, **Xianqiu Li**, and Xilin Chen. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pages 720–729, 2015.

【攻读硕士学位期间参加的项目】

- [1] Intel 的凝视矫正项目，2014 年 9 月至 2015 年 7 月

【攻读博士学位期间的获奖情况】

- [1] 理光 Theta 相机高校创新挑战赛“优秀奖”

