

密级: _____



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: 计算机科学与技术

研究 所: 中国科学院计算技术研究所

2016 年 5 月

**Discriminant Learning 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

May, 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，针对集合中的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研究该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。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、矩阵函数等概念介绍的基础上，对矩阵流形上的优化问题进行探讨，并结合学位论文课题中的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，另一方面也为解决类似优化问题提供借鉴。2) 提出了黎曼流形上的偏最小二乘回归方法，通过借助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, PLSR）扩展到黎曼流形；并考虑到黎曼流形与欧氏空间的几何结构差异以及图像集合数据稀疏的问题，进一步设计了借助多切空间构建子流形的方法，采用逐步回归的策略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，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时空开销大等问题；并采用图嵌入（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，最后在核判别分析（Kernel Discriminant Analysis, KDA）的框架下研究了该表示下的判别学习问题，并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

Abstract

Vision functionality serves as one of the main abilities for human to percept the real world, and its importance goes without saying. The mission of CV (Computer Vision) is to endow computers with close to or even stronger ability than human to perceive the real world.

As the main input for CV tasks, images contain much more information than text, audio and so on, but 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 comes 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 a large amount of images in poor quality. So one major task in image set classification is to overcome the disadvantage of low quality and leverage the advantage of large quantity.

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

Among the categories listed above, Statistics model based methods have attracted a lot attention with its excellent performance. This thesis takes Riemannian manifold as basic tool and tries to explore statistics model based methods. The main contributions include: 1) Studied matrix function and manifold optimization theory and methods. By introducing the basic concept of manifold and matrix function, along with the real-world problems extracted from the following research topics, optimization algorithms on the manifold have been studied in this thesis (Chapter 2). On the one hand it will help readers understand and implement methods proposed in this thesis, and on the other hand it can also provide basic instructions to solve other similar problems. 2) Proposed Partial Least Square Regression methods on Riemannian manifold with sub-manifold constructed from one tangent space (usually taking 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 drawback of sparse sampling, multi-tangent space Partial Least Square Regression method has been designed. On the Symmetric Positive Definite (SPD) matrices manifold, image set classification experiment were designed to evaluate the proposed method and it is observed that the proposed method is comparable or even outperforms the state-of-the-art methods on the commonly used databases. 3) Proposed Low-Rank

PSD matrices based image set model 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 (Low-Rank symmetric Positive Semi-Definite) representations of image sets and then designed the discriminant learning methods with Kernel Discriminant Analysis framework. Experiments on the commonly used databases has shown to 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 问题的背景与意义	1
1.2 国内外研究现状	2
1.2.1 符号说明	2
1.2.2 图像集合	3
1.2.3 子空间以及流形建模的方法	4
1.2.4 仿射包建模的方法	6
1.2.5 统计建模图像集合的方法	7
1.2.6 深度学习的方法	9
1.2.7 国内外研究现状小结	11
1.3 数据介绍	11
1.4 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 对称正定矩阵流形	17
2.2 优化问题与梯度	18
2.2.1 拉格朗日对偶问题	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	23
2.3.3 矩阵函数的偏导数计算示例	24
2.4 矩阵流形上的基本优化问题	27
2.5 总结	31
第三章 黎曼流形上的偏最小二乘回归	33
3.1 偏最小二乘方法	33
3.2 黎曼流形上的投影问题	36
3.2.1 一般化的投影	36
3.2.2 对称正定矩阵流形上的均值	37
3.2.3 黎曼流形上的子流形与投影	38
3.3 黎曼流形上的偏最小二乘回归问题	38
3.3.1 黎曼流形上偏最小二乘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偏最小二乘回归	40
3.4 实验验证	46
3.4.1 原始特征构造	46
3.4.2 对称正定矩阵表示的构造	47
3.4.3 实验结果与分析	47
3.5 总结与下一步工作	48
第四章 低秩对称半正定矩阵判别学习方法	51
4.1 施蒂费尔流形和格拉斯曼流形	52
4.2 固定秩对称半正定矩阵流形	55
4.3 固定秩对称半正定矩阵流形研究概况	56
4.3.1 固定秩对称半正定矩阵表示图像集合	57
4.3.2 固定秩对称半正定矩阵流形用于图像集合分类	57
4.4 低秩对称半正定矩阵判别学习方法	58
4.4.1 融入判别信息的低秩对称半正定矩阵的构造	60
4.4.2 低秩对称半正定矩阵集合上的判别学习方法	63
4.5 实验结果与分析	64
4.6 总结与下一步工作	66

第五章 结束语	69
5.1 本文工作总结	69
5.2 反思与讨论	70
参考文献	73
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	6
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	9
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	10
图 1.10 数据库示例	12
图 3.1 欧氏空间偏最小二乘回归示意图	35
图 3.2 流形上的投影示意图	39
图 3.3 算法 7 的示意图	44
图 3.4 算法 8 的示意图	46
图 4.1 特征值分解示意图	52
图 4.2 固定秩对称半正定矩阵示意图	57
图 4.3 图嵌入框架示意图	61
图 4.4 带判别信息的低秩对称半正定矩阵模型示意图	64

表目录

表 1.1 符号说明	2
表 1.2 数据库列表.....	11
表 3.1 黎曼流形上的偏最小二乘回归算法实验结果.....	48
表 4.1 固定秩对称半正定矩阵流形中的核	58
表 4.2 对称正定矩阵流形上的距离度量.....	59
表 4.3 低秩对称半正定矩阵判别学习方法对比实验结果	65
表 4.4 Power-Euclidean 距离相关的核	66

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。

绪论作为文章的开始，将为学位论文的展开作铺垫，引领读者进入本文的研究领域——图像集合分类问题。为此，本章首先会对问题的背景和意义进行简要的说明，让读者对该问题在国内外的研究现状有个整体把握的同时，对该问题的测试数据和测试协议也有一个大致的了解，最后借助对本文的主体框架的介绍，让读者对文章的结构有一个宏观的把握。

1.1 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甚至是超过人类视觉系统对于周围环境的处理能力。图像作为计算机视觉的主要输入，为计算机理解环境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，图像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，遮挡等问题使得用单一的图像进行识别、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，多视角图像数据等都以图像集合的形式呈现出爆发式的增长。利用集合数据的优势克服以上一些问题逐渐成为了一种趋势，图像集合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的特点是：量大但质未必优（variation 大）。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，加之数据本身以集合的形式呈现的特点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来源于实际问题的，图像集合的分类也不例外，视频监控就是一个很好的例子，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，图像数据中的数据是以集合的形式存在，相较于机器学习领域的中的单点（向量）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，其数学表示也比较特殊，往往是子空间，对称正定矩阵，分布函数，流形等。而这些非线性结构表示的研究也将促进机器学习中非线性数据表示的研究。

1.2 国内外研究现状

本节将针对图像集合分类问题在国内外的研究现状进行介绍，帮助读者了解该问题的前沿动态，理解图像集合分类问题本身以及该问题的核心任务和主流的解决方案。

1.2.1 符号说明

在进入本节的主要内容之前，由于本文涉及较多数学符号；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}, \mathbf{y} ）表示向量，大写的字母（如： X, Y ）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}, \mathbf{Y} ）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 的二元组），通常为了简单起见也用 S 代表该流形（如用 \mathbb{S}_d^+ 表示对称正定矩阵流形），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
$\text{St}(n, k)$	non-compact Stiefel 流形，定义在 $n \times k$ 列满秩矩阵的集合上
$\text{St}^*(n, k)$	compact Stiefel 流形，定义在 $n \times k$ 列正交矩阵的集合上
$\text{Gr}(n, k)$	Grassmann 流形，定义在 \mathbb{R}^n 中 k 维子空间构成的集合上
$\log(\cdot)$	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

1.2.2 图像集合

图像集合，顾名思义指的就是多张图片构成的集合，其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给出了几个例子。



(a) EXAMP 01: 一段录像（图片来自 YTC[1] 数据库）



(b) EXAMP 02: 一个物体的 Multi-view（图片来自 ETH80[2] 数据库）



(c) EXAMP 03: 一个动作描述（图片来自 CMU MoBo[3] 数据库）

图 1.1 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。

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，在图像集合分类问题中 Probe 相当于测试集而 Gallery 相当于训练集），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，拥有着广泛的应用前景，在该方向上的一些主要工作有：文献 [4] 探究了静态图像到图像集合的分类问题；文献 [5] 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 [6] 借助 Affine Hull 表示图像集合，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方向；在这个问题上，根据图像集合表示方式的不同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 [7–10]；2、仿射包建模的方法 [11–14]；3、统计建模的方法 [15–21]；4、深度学习的方法 [22,23]；5、其它（稀疏编码 [24]，协同表示 [25] 等）。接下来的内容将简要对它们进行介绍。

1.2.3 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。

1.2.3.1 子空间建模的方法

工作 [7], [8] 是使用子空间建模图像集合的代表，工作 [7] 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 [8] 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 [8]）：

- 设 $\{\mathbf{x}_{ij} \in \mathbb{R}^d\}_{j=1}^{n_i}$ 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： $C_i = \frac{1}{n_i-1} \sum_{j=1}^{n_i} (\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)^T$
- 对样本协方差做特征值分解获得： $C_i = U_i \Lambda_i U_i^T$ ，指定子空间维数 $m(m < d)$ ，这里假设特征值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 $\{Y_j\}_{j=1}^n$ 的两两之间的距离；在子空间的距离度量中，主夹角是最主要的概念：

$$\cos \theta_k = \max_{\mathbf{u}_k \in \text{span}(Y_i)} \max_{\mathbf{v}_k \in \text{span}(Y_j)} \mathbf{u}_k^T \mathbf{v}_k$$

$$s.t \quad \mathbf{u}_k^T \mathbf{u}_k = 1, \mathbf{v}_k^T \mathbf{v}_k = 1 \quad (1-1)$$

$$\mathbf{u}_k^T \mathbf{u}_i = 0, \mathbf{v}_k^T \mathbf{v}_i = 0, (i = 1, 2, \dots, k-1)$$

其物理意义如图1.2所示。

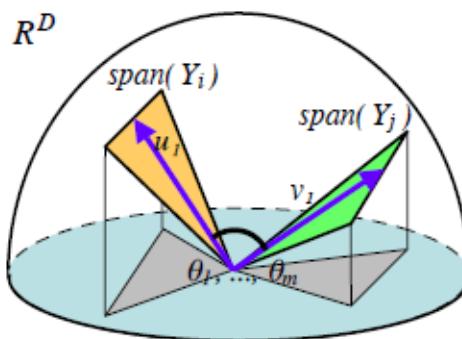


图 1.2 子空间之间的主夹角示意图（图片来自文献 [8]）

- 利用主夹角定义子空间中的距离度量：

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

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

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

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

工作 [8] 进一步在此基础上利用核判别分析 (Kernel Discriminant Analysis, KDA) 的框架进行了判别学习，在核空间中进行图像集合的分类。

1.2.3.2 流形建模的方法

流形建模的方法 [9], [10] 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，然后利用此结构定义流形与流形之间的距离进行图像集合分类，如图1.3所示。

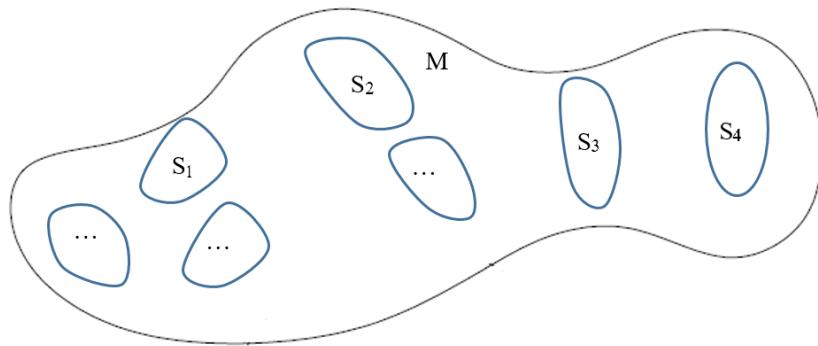


图 1.3 流形的局部线性近似示意图

其中 M 表示的是原始的流形结构，文献 [9] 根据数据构建局部线性子空间 $S_1, S_2, S_3, S_4, \dots$ 来近似表示流形 M ，然后通过点到点的距离定义点到子空间的距离再进一步定义子空间到子空间的距离，最后定义流形到流形的距离，从而进行图像集合的分类（下述定义中的 S, S_i, C_j 表示的是子空间而不是矩阵）。

- Point to point distance: $d_{ppd}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$
- Point to subspace distance: $d_{psd}(\mathbf{x}, S) = \min_{\mathbf{x}' \in S} \|\mathbf{x} - \mathbf{x}'\|$
- Subspace to subspace distance: $d_{ssd}(S_1, S_2) = \text{any valid subspace metric}$
- Point to manifold distance: $d_{pmd}(\mathbf{x}, M) = \min_{C_i \in M} d_{psd}(\mathbf{x}, C_i)$
- 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 [9] 利用上述的 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 [10] 则是在 [9] 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。

流形建模图像集合的方法是早期的流形学习的概念和图像集合问题的结合，在图像集合分类问题的研究上进行了有益的探索，也为后来的图像集合问题的研究（如 [20,21]）提供了借鉴意义。

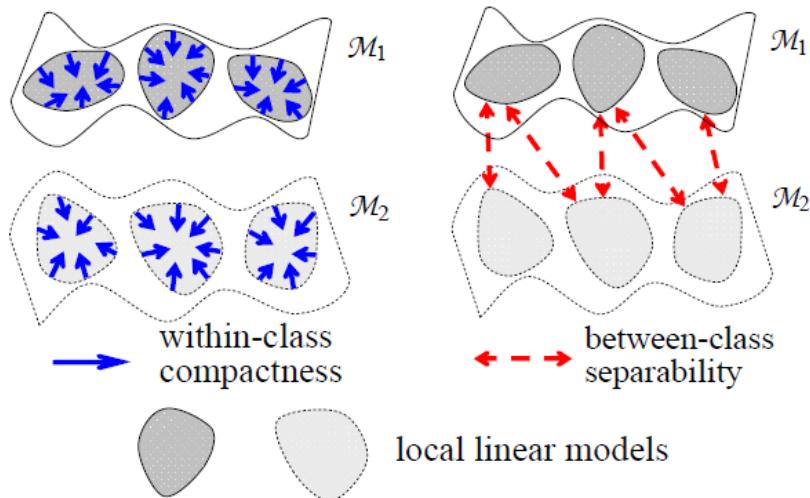


图 1.4 MDA 方法示意图 (图片来自 [10])

1.2.4 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，该方法的代表作有 [11–14]。仿射包建模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。

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

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。

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

如果直接利用公式1-4的定义进行图像集合分类的话容易出现两个仿射包相交 (也就是距离为 0) 的情况。所以仿射包建模图像集合的一大问题就是对噪声不够鲁棒，针对这个问题，文献 [11] 中提出了使用 Convex Hull 的表示方法 (如1-5所示)。

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

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, [12] 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 [13] 提出了使用 l_p 范数 (通常 $p = 2$) 来代替 l_1 范数。

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

而文章 [14] 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总结一下仿射包建模图像集合方法之间的关系。

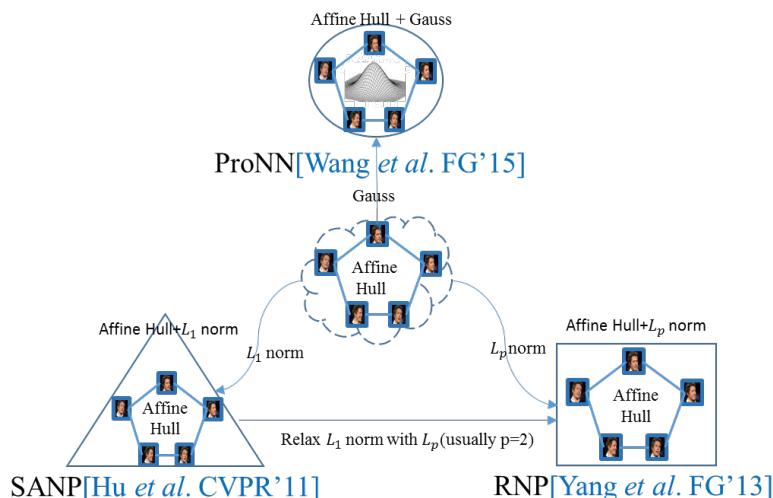


图 1.5 仿射包建模图像集合方法关系图

概括起来就是: 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。

1.2.5 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的主流方法之一, 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。

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) 一般是正定的 $C > 0$, 因此其并不构成线性空间, 图1.6给出了 2×2 对称正定矩阵在三维空间中的例子^①。

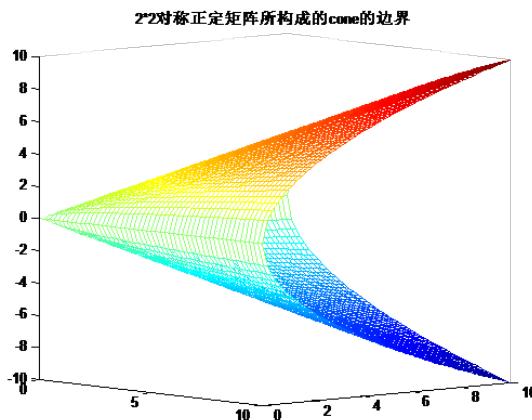


图 1.6 2×2 对称正定矩阵的外边界在 3 维空间中的结构

协方差建模图像集合的代表作 [15]: 利用核函数 $\phi = \log(\cdot)$ 将协方差矩阵流形映射到 RKHS 空间, 在新的 RKHS 空间中, 使用 KPLS[26] 回归以及 KDA[27] 对视频人脸, 多视角图像集等图像集合数据进行分类。

文献 [17] 则在对称正定矩阵集合中, 利用黎曼度量进行协方差降维及判别学习, 算法的优化空间 (投影矩阵所在的空间) 为 Grassmann 流形。

多统计模型融合的方法主要的代表作有 [18] 和 [19], 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。在具体实现过程中, 工作 [18] 首先利用核映射将不同统计模型映射到再生核希尔伯特空间 (Reproducing Kernel Hilbert Space, RKHS) 中, 然后利用局部多核度量学习将其整合到一起进行分类, 其算法流程如图1.7(a)所示。工作 [19] 首先也利用核函数将不同的统计模型映射到统一的再生核希尔伯特空间空间中, 然后在新的再生核希尔伯特空间中利用融合的特征进行分类, 其核心内容可以用图1.7描述。

为了更好的挖掘图像集合原始分布的信息, 分布函数建模图像集合的方法被提出: 文献 [20] 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, 并在核判别学习 (Kernel Discriminant Analysis, KDA)[27] 的框架下学习投影矩阵, 最后在投影空间中分类; 而文献 [21] 则使用 KDE(Kernel Density Estimation) 表示图像集合 (逼近原始分布), 并为其设计距离/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开始, 经

^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

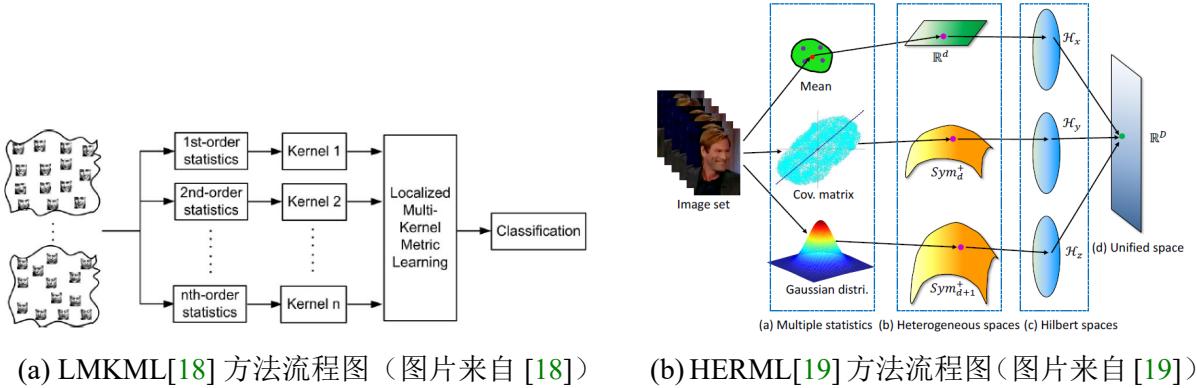


图 1.7 多统计模型建模图像集合方法

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) 由于数据表示的特殊性, 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。

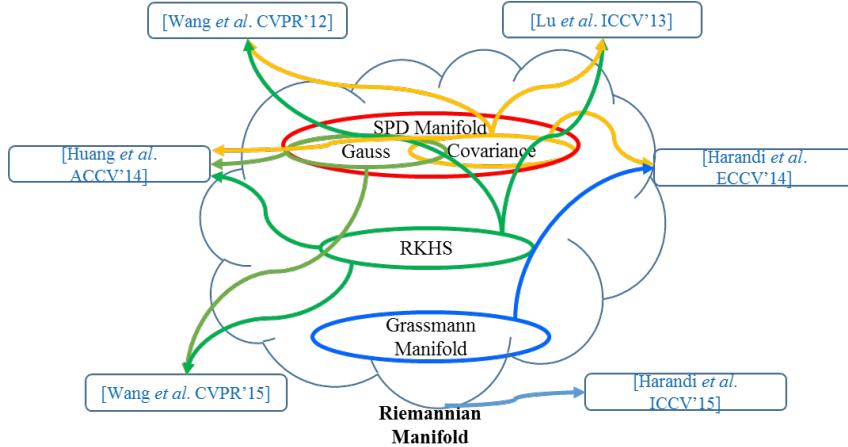


图 1.8 统计建模图像集合的方法间的关系

1.2.6 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 (Deep Learning, DL) 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 [23] 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投票进行分类。文献 [22] 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。

深度学习将计算机视觉的发展推向了一个新的高度, 也为图像集合的研究注入了新的活力, 所以这里使用一定的篇幅对其进行系统的介绍; 由于工作 [22] 是对工作 [23] 的改进, 所以这里仅以 MMDML[22] 为例对该类方法做一个介绍: 在 L+1 层的 DNN 网络

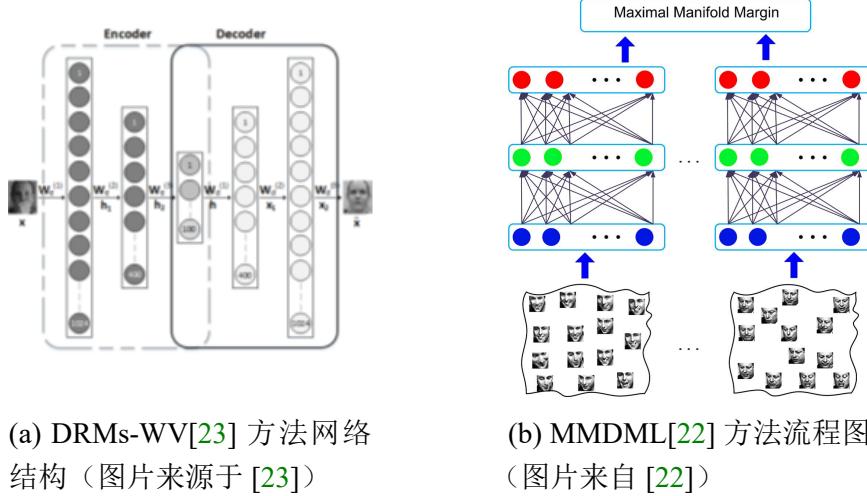


图 1.9 深度学习建模图像集合的代表性网络结构

中, 对于第 c 个图像集合的第 i 张图片其顶层输出如公式1-8所示。

$$\mathbf{h}_{ci}^L = s(W_c^L \mathbf{h}_{ci}^{L-1} + \mathbf{b}_c^L) \quad (1-8)$$

其中 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[22] 方法在网络顶层最大化不同 manifold 之间的 margin; 为此, 工作 [22] 为每类的每个样本 (如第 c 类的第 i 个样本) 定义公式1-9中的近邻关系描述。

$$\begin{cases} D_1(\mathbf{h}_{ci}^L) = \frac{1}{K_1} \sum_{p=1}^{K_1} \|\mathbf{h}_{ci}^L - \mathbf{h}_{cip}^L\|_2^2 \\ D_2(\mathbf{h}_{ci}^L) = \frac{1}{K_2} \sum_{q=1}^{K_2} \|\mathbf{h}_{ci}^L - \mathbf{h}_{cig}^L\|_2^2 \end{cases} \quad (1-9)$$

其中 \mathbf{h}_{cip}^L 表示的是当前样本的第 p 个同类近邻在网络顶层的输出, \mathbf{h}_{cig}^L 表示的是当前样本的第 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。故上式定义了第 c 类的第 i 个样本在网络顶层与 K_1 个同类近邻, K_2 个不同类近邻的关系。

MMDML 方法的目标是在网络顶层最大化不同 manifold 之间的 margin:

首先使用 $f_c = [W_c^1, W_c^2, \dots, W_c^L, \mathbf{b}_c^1, \mathbf{b}_c^2, \dots, \mathbf{b}_c^L]$ 表示网络参数, 然后定义:

$$\begin{cases} H_1 = \sum_{c=1}^C \sum_{i=1}^{N_c} g(D_1(\mathbf{h}_{ci}^L) - D_2(\mathbf{h}_{ci}^L)) \\ H_2 = \sum_{c=1}^C \sum_{l=1}^L (\|W_c^l\|_F^2 + \|\mathbf{b}_c^l\|_2^2) \end{cases} \quad (1-10)$$

其中 $g(a) = \frac{1}{\rho} \log(1 + \exp(\rho a))$ (ρ 表示锐度参数), N_c 表示第 c 个图像集合中的样本数。利用公式1-10得到 MMDML 的优化目标:

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

文献 [22] 中使用了随机次梯度下降算法训练网络参数，在最后在测试阶段，测试样本的分类结果由公式1-11获得。

$$L_q = \arg \min_c d(X_q, X_c), 1 \leq c \leq C \quad (1-11)$$

其中 $X_q = [\mathbf{x}_1^q, \mathbf{x}_2^q, \dots, \mathbf{x}_{N_q}^q]$ 表示的是一个测试图像集 X_c 表示的是一个训练图像集，而 $d(X_q, X_c)$ 的计算过程如下：1) 使用第 c 个网络将 \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 $\mathbf{h}_{ci}^L, i = 1, 2, \dots, N_c$ 的欧氏距离，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 $d(X_q, X_c)$ 。

1.2.7 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的研究可做如下描述：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，并用距离/度量进行判别学习；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中，由于数据的特殊性，往往需要对非线性的数据模型进行研究，如：子空间、统计模型等。在这些模型的研究中，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，还有其它一些方法，如：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。

1.3 数据介绍

图像集合分类问题源于实际，最终还是要回到实际中；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。图1.10给出这些数据集的一些示例图片。

表 1.2 数据库列表

数据库	描述	备注
YouTube Faces DB[28]	1595 个人，3425 段视频，低分辨高压缩率	人脸数据库
YouTube Celebrity[1]	47 个人，1910 段视频，低分辨高压缩率	人脸数据库
UIUC[29]	4 个大类，18 个子类，每类 12 张图片	材质分类数据库
ETH-80[2]	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[3]	25 个人，4 个运动（行走）类，150 段视频数据	为步态研究而搜集

YouTube Faces DB 数据库 [28] 最初是为视频人脸验证任务收集的，其收集过程中根据 LFW(Labeled Faces in the Wild[30]) 数据集的样本进行，数据库包含了 1595 个人的

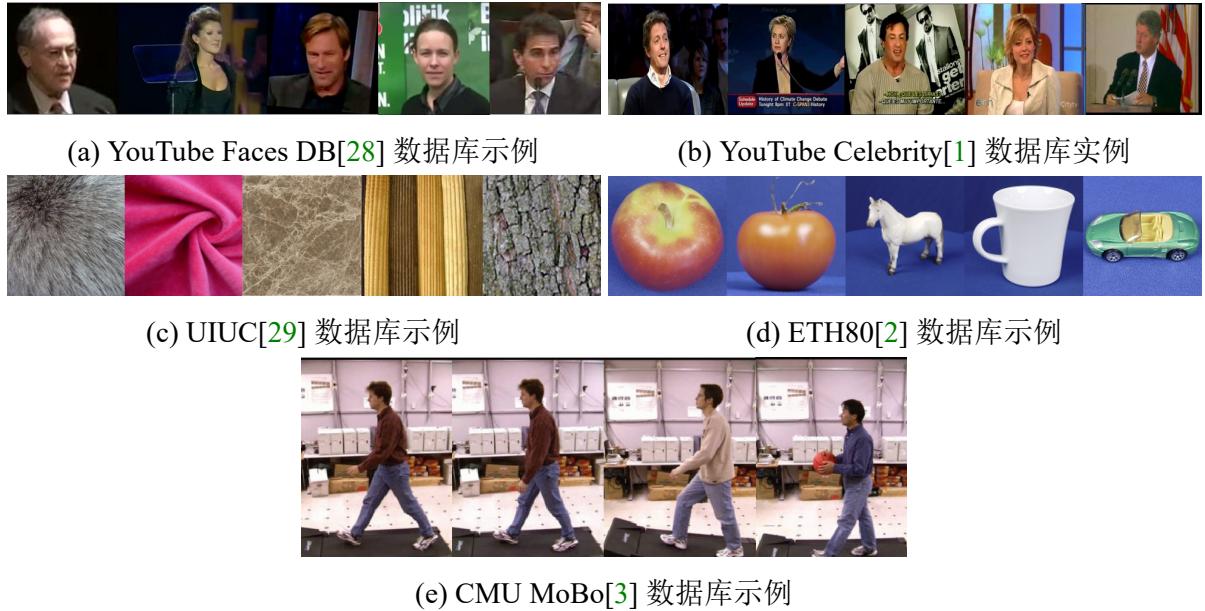


图 1.10 数据库示例

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 [1] 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，此外同样存在脸部姿态变化、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 [29] 是材质数据库 (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，但是当使用 Region Covariance[31] 为图像建模表示的时候，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 [2] 主要用于物体识别任务，数据集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)人造的类别，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 [3] 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 种行走姿态 (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[3] 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）所以这里仍然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据的划分情况如下：在 YTF(YouTube Faces DB) 上根据文献 [28] 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[1]) 上数据的划分则是参考了 [15] 以及 [16] 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 [29] 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 [15]，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。

1.4 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，这部分的内容通过对黎曼流形，矩阵函数等基本概念的介绍，对矩阵函数求导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帮助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氏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中，得到黎曼流形上偏最小二乘问题的基础版本，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方法出发提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问题后，考虑使用半正定 (symmetric Positive Semi-Definite, PSD) 矩阵表示图像集合，在对早期工作以及工作 [32] 中的不足进行分析之后，提出了使用嵌入判别信息的低秩对称半正定矩阵建模图像集合的方法，最终实验验证了该建模方法的有效性。

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，本章正是对该问题的探索。不过本章的内容也是一个一般化的问题研究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 [33] 或 [34] 中那样详细的介绍黎曼流形上的优化问题。作为基础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与优化问题，最后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。目的是一是方便读者理解作者学位论文中研究课题，目的二是为了让读者在遇到类似问题的时候能够从中提炼出思路和解决方案。

对于流形上很细节的问题读者还是到文献 [33,34] 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，将在这一节对其进行简要介绍。本节的内容主要包括两部分：第一部分是基本流形和黎曼流形的基本定义和性质的介绍；第二部分将就黎曼流形中的对称正定矩阵流形 (Symmetric Positive Define, SPD) 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, AIM 和 Log-Euclidean Metric, LEM)（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 [35] 和 [36]）。

2.1.1 黎曼流形

流形是数学上的抽象概念，它的定义则依赖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，并且将 τ 中的元素叫作开集；拓扑的研究中，点集拓扑是一个重要的内容。关于点集拓扑两个概念需要理解（系统的内容可以参看 [35]），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

^① 具有这样性质的空间也叫做第二可数的

性质则这样的的拓扑空间称为 T_2 的。Hausdorff 性质：假设 S 是拓扑空间，设 \mathbf{x} 和 \mathbf{y} 是 S 中的点，我们称 \mathbf{x} 和 \mathbf{y} 可以“由邻域分离”，如果存在 \mathbf{x} 的邻域 U 和 \mathbf{y} 的邻域 V 使得 U 和 V 是不相交的（即： $U \cap V = \emptyset$ ），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。 A_2, T_2 这两个基本的概念在本文的研究中不会涉及，此处是出于完备性的考虑将其放在这里。

定义 2.2 (r 阶连续) 若一函数是连续的，则属于 C^0 函数。若函数存在连续导函数，则属于 C^1 函数；若函数 r 阶可导，并且其 r 阶导函数连续，则属于 C^r 函数 ($r \geq 1$)。而任意光滑函数是对所有 r 都有 r 阶的连续导数，并用 C^∞ 表示这一类函数。

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 [35]。

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 到欧氏空间开集 $\varphi_\alpha(U_\alpha)$ 上的同胚^②
2. 当 $U_\alpha \cap U_\beta \neq \emptyset$ 时，若如下的转换映射：

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

属于 $C^r (r \geq 1)$ 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 $\mathcal{D} = \{(U_\alpha, \varphi_\alpha), \alpha \in \Gamma\}$ ；若 \mathcal{D} 是最大的，也就是说当坐标卡 (U, φ) 与 \mathcal{D} 中任意的 $(U_\alpha, \varphi_\alpha)$ 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 [35] 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 上任意光滑函数的全体组成的空间。设 $p \in M$ ，如果线性映射 $X_p : C^\infty(M) \rightarrow \mathbb{R}$ 满足以下条件：

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

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 $T_p M$ 。

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 [35]）

上述定义的切向量比较晦涩，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) 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 $\sigma(0) = p$, 现定义 $\sigma'(0)$ 满足:

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

可以验证 $\sigma'(0) \in T_p M$, 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 $g_p(\mathbf{x}_p, \mathbf{y}_p) = g_p(\mathbf{y}_p, \mathbf{x}_p)$

并令 g 表示映射族 $\{g_p, p \in M\}$, 则称 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:

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

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。

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

最后, 这里记录另一条对于计算距离有用的性质: 在等距同构的映射下, 新空间中测地线的长度仍然是原来空间中的测地线的长度。

2.1.2 对称正定矩阵流形

对称正定矩阵流形是本文研究的主要对象之一, 在前面的1.2.5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 (Symmetric Positive Definite, SPD) 矩阵的形式存在。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(Affine-Invariant Metric[37]) 和 LEM(Log-Euclidean metric[38])。下面首先给出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

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

当赋上适当的度量 g (如 AIM[37] 或 LEM[38]) 之后即构成黎曼流形 (\mathbb{S}_d^+, g) 。

在 PSD 矩阵流形上, AIM 度量 [37] 和 LEM 度量 [38] 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。

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

其中 H 是 X_1 处的切空间上的切向量, $D_{X_1} \log.(H)$ ^① 表示的是 $\log(X_1)$ 沿着方向 H 的方向导数, $\langle \cdot, \cdot \rangle_{X_1}$ 表示的是 X_1 的切空间上的黎曼度量, $D_{\log(X)} \exp. = (D_X \log.)^{-1}$ 是由等式 $\log \circ \exp = I$, 更多详细信息可参看文献 [38]。

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, 这里选取具有代表性的一类问题——Lagrange 对偶问题进行介绍, 接着第二部分会以梯度为主线介绍梯度在优化问题中应用, 最后简要介绍梯度下降与共轭梯度算法, 为将算法扩展到流形上做准备。

2.2.1 拉格朗日对偶问题

拉格朗日对偶问题 (Lagrange Dual Problem) 的转化是求解带约束问题的重要方法, 在带约束的优化问题中有着重要的地位, 其转化过程将在接下来的部分进行描述, 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。

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

^① 这里为了描述的方便, 使用的是原文 [38] 中的表示形式与本章后续描述的方向梯度的形式稍有区别, 它等价于本章后续的 $D_{X_1} \log(X_1)[H]$ 的形式

由 Lagrange 函数²⁻⁷定义原问题的 Lagrange 对偶函数^①如式²⁻⁸所示。

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

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 才是好的呢？对偶问题考虑：

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

所以实际上的 Lagrange 对偶问题求解的问题如式²⁻¹⁰所示。

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

最后，对偶问题与原问题的关系可由 KKT (Karush-Kuhn-Tucker，公式²⁻¹¹) 条件刻画：若对偶问题的解满足 KKT 条件，那么它也是原问题的解。

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

至此 Lagrange 对偶问题的介绍基本结束，接下来的部分主要与梯度及优化问题相关。

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确界，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）：

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

其中 $D_{\mathbf{x}}f(\mathbf{x})[\mathbf{d}]$ 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 $\frac{\partial}{\partial x_i} f(\mathbf{x})$ ，这里不再赘述；此外公式2-12另一个重要的用途是梯度检查（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到如2-13所示的优化问题中。

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

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 $\text{dist}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})$ 的时候，其最优解在 $\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ 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[38] 的距离计算框架下，问题2-13也有解析解 $\boldsymbol{\mu} = \exp(\frac{1}{n} \sum_{i=1}^n \log(x_i))$ ^①。

但是对于 $X_i \in \mathbb{S}_d^+$ 且样本数 $n > 2$ 的时候，在 Affine-Invariant Distance[37] 的距离计算框架下，问题2-13一般没有解析解，甚至于不能保证问题的唯一极小值点的存在，此时最优化的方法将发挥作用^②。关于 \mathbb{S}_d^+ 在 AID[37] 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 $\boldsymbol{\mu}$ 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：

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

关于梯度下降，首先需要注意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。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，等式右边达到极小值，且该方向是 $f(\mathbf{x}_0 + \alpha \mathbf{g})$ 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），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是为了克服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。1952 年的时候，Hestenes 和 Stiefel 为了求解线性方程组而提出该方法。后来，该方法经过修改之后被用于求解一般的无约束最优化问题，并最终成为一种有效的最优化方法。共轭梯度算法有很多种模式，其中 Fletcher-Reeves 共轭梯度法 [39]（简称 FR 法）是其中具有代表性的一种，接下来将以该算法为例进行说明。

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：

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

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 [39]：

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

关于算法的细节读者可以参看原文 [39] 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。

^① 其中 H 矩阵就是通常意义的 Hessian 矩阵

2.3 矩阵函数的导数计算

本部分的内容主要针对矩阵函数的导数计算及对称正定矩阵流形上的导数计算两个方面；矩阵函数的计算由于自变量的特殊性有其特别的地方。在“*The matrix cookbook*”[40]中包含了矩阵求导的大多数情况，但是对于一些比较特殊的情况（如 SPD 矩阵中在仿射不变距离计算框架下最小化 Fréchet Variance 的问题）可利用其问题的特点寻找更合适的解决方案；接下来的内容主要参考文献[33]和[34]。

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 的特征值分解为 $A = U\Lambda U^T$)，以及任意光滑实值函数 $f(x)$ ，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[33]中的内容，这里先从方向导数2-12开始介绍，并先给出两条计算方向导数的规律[33]：

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

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2-21中的内积定义 $\langle \cdot, \cdot \rangle$ 最常见的是矩阵的内积： $\langle A, B \rangle = \text{tr}(AB^T)$ 。

根据公式2-12的定义首先来看多项式 A^k 的方向导数（本节以后总假设 A 可对角化为 $U\Lambda U^T$ ，其中 $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_d)$ 是特征值构成的对角矩阵）：

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

利用公式2-20和公式2-22可得到：

$$\begin{aligned} D_A f(A)[H] &= \sum_{k=0}^{\infty} \alpha_k D_A A^k[H] = \sum_{k=0}^{\infty} \alpha_k U \left(\sum_{l=1}^k \Lambda^{l-1} [U^T H U] \Lambda^{k-l} \right) U^T \\ &= U \left(\sum_{k=0}^{\infty} \alpha_k \sum_{l=1}^k \Lambda^{l-1} [U^T H U] \Lambda^{k-l} \right) U^T \\ &= U D_\Lambda f(\Lambda) [U^T H U] U^T \end{aligned} \quad (2-23)$$

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

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

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

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

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

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

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 \tilde{H} : $\tilde{H} = U^{-1} H U$
- 计算矩阵 F :

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

- 计算 $D_A f(A)[H] = U(F \odot \tilde{H})U^{-1}$

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, 在本节将对该特殊情况进行讨论。

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

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 的特征值分解。 $U(\Lambda + \mu I)U^T$ 是 B 的特征分解 (为方便起见记

$D = \Lambda + \mu I \triangleq \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_d)$ ，利用公式2-23的话可得到公式2-29的形式。

$$\begin{aligned}
D_B f(B)[H] &= \sum_{k=0}^{\infty} \alpha_k D_B B^k [H] \\
&= \sum_{k=0}^{\infty} \alpha_k U \left(\sum_{l=1}^k D^{l-1} [U^T H U] D^{k-l} \right) U^T \\
&= U \left(\sum_{k=0}^{\infty} \alpha_k \sum_{l=1}^k D^{l-1} [U^T H U] D^{k-l} \right) U^T \\
&= U D_D f(D) [U^T H U] U^T
\end{aligned} \tag{2-29}$$

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

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

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：

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

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

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

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 $M_{ij} = \lim_{\mu \rightarrow 0} \sum_{k=1}^{\infty} \alpha_k \sum_{l=1}^k k \mu^{k-1} \tilde{H}_{ij} = \tilde{H}_{ij} f'(0)$ 。总结以上结果不难发现 $\lambda_i \lambda_j = 0$ 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

问题中却有着重要的意义。

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低秩对称半正定 (Low-Rank symmetric Positive Semi-Definite, Low-Rank PSD) 的 Power Euclidean 距离的交叉项的讨论 (做了一些简化, 并要求 $Z > 0$ 使得优化空间为 \mathbb{S}_d^+); 我们将关于 $f(Z)$ 的导数计算过程归纳如下:

为了方便起见, 定义 $g_1(Z) = (C_1 + Z)(C_1 + Z)^T, g_2(Z) = (C_2 + Z)(C_2 + Z)^T$ 。利用2-21中的定律, 可得到:

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

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, 故接下来仅以前一部分作为研究对象。

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

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 $B = D_Z g_1(Z)[H], g_1(Z) = U_1 \Lambda_1 U_1^T$, 于是有公式2-36的计算过程。

$$\begin{aligned} \text{Compute : } \tilde{H} &= U_1^T B U_1 \\ \text{Compute : } \tilde{F}, \text{ where } \tilde{F}_{ij} &= \begin{cases} \frac{\lambda_i^{\frac{1}{n}} - \lambda_j^{\frac{1}{n}}}{\lambda_i - \lambda_j}, & \text{if } \lambda_i \neq \lambda_j \\ \frac{1}{n} \lambda_i^{\frac{1}{n}-1}, & \lambda_i = \lambda_j \end{cases} \\ \text{Compute : } M &= \tilde{H} \odot \tilde{F} \\ \text{Compute : } D_Z (g_1(Z))^{\frac{1}{n}} [B] &= U_1 M U_1^T \end{aligned} \quad (2-36)$$

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：

$$\begin{aligned}
& \left\langle D_Z(g_1(Z))^{\frac{1}{n}} [B], (g_2(Z))^{\frac{1}{n}} \right\rangle \\
&= \left\langle U_1 M U_1^T, (g_2(Z))^{\frac{1}{n}} \right\rangle = \left\langle \tilde{H} \odot \tilde{F}, (U_1^T (g_2(Z))^{\frac{1}{n}} U_1) \right\rangle \\
&= \left\langle U_1^T B U_1, (U_1^T (g_2(Z))^{\frac{1}{n}} U_1) \odot \tilde{F} \right\rangle, \text{where } \tilde{H} = U_1^T B U_1 \\
&= \left\langle B, U_1 (U_1^T (g_2(Z))^{\frac{1}{n}} U_1) \odot \tilde{F} U_1^T \right\rangle \\
&= \left\langle (C_1^{\frac{1}{n}} + Z) H^T, U_1 (U_1^T (g_2(Z))^{\frac{1}{n}} U_1) \odot \tilde{F} U_1^T \right\rangle \\
&\quad + \left\langle H (C_1^{\frac{1}{n}} + Z^T), U_1 (U_1^T (g_2(Z))^{\frac{1}{n}} U_1) \odot \tilde{F} U_1^T \right\rangle \tag{2-37} \\
&= \text{tr} \left((C_1^{\frac{1}{n}} + Z) H^T \left(U_1 (U_1^T (g_2(Z))^{\frac{1}{n}} U_1) \odot \tilde{F} U_1^T \right)^T \right) \\
&\quad + \text{tr} \left((C_1^{\frac{1}{n}} + Z) H^T U_1 (U_1^T (g_2(Z))^{\frac{1}{n}} U_1) \odot \tilde{F} U_1^T \right) \\
&= 2 \text{tr} \left(H^T \text{symm} \left(U_1 (U_1^T (g_2(Z))^{\frac{1}{n}} U_1) \odot \tilde{F} U_1^T \right) (C_1^{\frac{1}{n}} + Z) \right) \\
&= 2 \left\langle H, \text{symm} \left(U_1 (U_1^T (g_2(Z))^{\frac{1}{n}} U_1) \odot \tilde{F} U_1^T \right) (C_1^{\frac{1}{n}} + Z) \right\rangle \\
&= 2 \left\langle H, Z \text{symm} \left(D_{g_1(Z)} (g_1(Z))^{\frac{1}{n}} [(g_2(Z))^{\frac{1}{n}}] \right) (C_1^{\frac{1}{n}} + Z) Z \right\rangle_Z
\end{aligned}$$

其中 $\text{symm}(X) = 0.5(X + X^T)$, $\langle \cdot, \cdot \rangle_Z$ 表示的是 \mathbb{S}_d^+ 在 Z 的切空间中的黎曼度量；最后综合2-34~2-37的内容得到如公式2-38所示的形式。

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

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。

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

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 $\langle H_1, H_2 \rangle_A = \langle A^{-\frac{1}{2}} H_1 A^{-\frac{1}{2}}, A^{-\frac{1}{2}} H_2 A^{-\frac{1}{2}} \rangle$, 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。

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

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

$$\begin{aligned}
 D_X f(X|A)[H] &= D_X \frac{1}{2} \langle \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}), \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) \rangle [H] \\
 &= \langle D_X \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}})[H], \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) \rangle \\
 &= \langle D_{A^{-\frac{1}{2}} X A^{-\frac{1}{2}}} \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}})[D_X(A^{-\frac{1}{2}} X A^{-\frac{1}{2}})[H]], \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) \rangle \\
 &= \langle D_{A^{-\frac{1}{2}} X A^{-\frac{1}{2}}} \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}})[A^{-\frac{1}{2}} H A^{-\frac{1}{2}}], \log(A^{-\frac{1}{2}} X A^{-\frac{1}{2}}) \rangle
 \end{aligned} \tag{2-41}$$

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：

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

利用公2-42的结果可以将公式2-41的结果进一步的写成：

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

根据公式2-43最后两行内容可得到 $\nabla_X f(X|A)$ 分别在普通欧氏空间与 X 的切空间 $T_X M^{\textcircled{1}}$ 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

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，并介绍欧氏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。

^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

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。

在欧氏空间中迭代算法的更新公式一般为 $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$ ，其物理意义相当于以 \mathbf{x}_k 为起点，沿着 \mathbf{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 \mathbf{x}_{k+1} 跑出原来的流形。一般地，流形上的 $\exp_A(H), H \in T_A M$ 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，这是欧氏空间中 $A + H$ 概念的泛化；但是流形上的 Exp 操作的一大问题是计算量较大，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 变换是流形中的切空间束 (Tangent Bundle) TM 到流形本身 M 的具有如下性质的连续映射；这里令 R_X 表示切空间 $T_X M$ 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，那么将欧氏空间中的梯度下降算法泛化到流形会十分简单。这里类似于 [33] 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 $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ ，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 $k \leftarrow 0$
 - 2: **while** not converge **do**
 - 3: 计算负梯度方向： $\mathbf{d}_k = -\nabla_{\mathbf{x}} f(\mathbf{x}_k)$
 - 4: 计算迭代步长： $\alpha_k = \arg \min_{\alpha} f(\mathbf{x}_k + \alpha \mathbf{d}_k)$
 - 5: 更新结果： $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
 - 6: 更新迭代次数： $k \leftarrow k + 1$
 - 7: **end while**
 - 8: **return** $\{\mathbf{x}_i\}_{i=0}^k, \mathbf{x}^* = \mathbf{x}_k$
-

算法1中的收敛条件不尽相同，最理想的情况是 $\|\nabla_{\mathbf{x}} f(\mathbf{x}_k)\| = 0$ ，但是一般很难达到，所以一般是要求 $\|\nabla_{\mathbf{x}} f(\mathbf{x}_k)\| < \epsilon$ ，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 变换是欧氏空间中 $X + H$ 在流形中的泛化, 利用它代替 1 中的第四和第五步可以得到流形中的梯度下降算法 2。

Algorithm 2 流形上梯度下降算法

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: $\alpha_k = \arg \min_{\alpha} f(R_{X_k}(\alpha H_k))$
 - 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** $\{x_i\}_{i=0}^k, X^* = X_k$
-

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 $f(X|A) = \frac{1}{2}\text{dist}^2(A, X) = \frac{1}{2}\text{dist}^2(X, A) = f(A|X)$ 可以得到公式 2-44 的形式。

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

利用公式 2-44 的结果, SPD 矩阵流形中 Fréchet Variance: $C(\mu) = \frac{1}{n} \sum_{i=1}^n \text{dist}^2(X_i, \mu)$ 的导数如公式 2-45 所示。

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

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：

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

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中介绍的欧氏空间中的共轭梯度算法归纳到算法3中。

Algorithm 3 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: $\alpha_k = \arg \min_{\alpha} f(\mathbf{x}_k + \alpha \mathbf{d}_k)$

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** $\{\mathbf{x}_i\}_{i=0}^k, \mathbf{x}^* = \mathbf{x}_k$

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 [34] 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3泛化到矩阵流形中需要解决的问题有两个：1) $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$ 中的加法问题，这个已经有 Retraction 变换解决；2) $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k$ 中的加法问题，这里的主要问题是流形上 $\mathbf{d}_{k+1}, \mathbf{g}_{k+1} \in T_{\mathbf{x}_{k+1}} M, \mathbf{d}_k \in T_{\mathbf{x}_k} M$ （这里流形上的元素 $\mathbf{x}_k, \mathbf{x}_{k+1}$ 使用小写是为了与算法3对应）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。

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

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

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 [33,34] 以及文章 [41] 的补充材料，这里不再展开

Algorithm 4 流形上的共轭梯度算法

Require: 目标函数 $f(X) : M \rightarrow T_X M$, 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: $\alpha_k = \arg \min_\alpha f(R_{X_k}(\alpha H_k))$
- 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** $\{X_i\}_{i=0}^k, X^* = X_k$

并且满足如下的几条性质:

- (Retraction 关联) 如果一个 Retraction (记为 R), 对任意的 $X \in M$ 满足 $T_\eta(\xi) \in T_{R_X(\eta)} M$, 则称 R 与 T 关联
- (一致性) 对任意的 $X \in M$ 满足 $T_0(\xi) = \xi, \forall \xi \in T_X M$
- (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, 这里将流形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细节可以参看 Manopt[42] 的实现。

2.5 总结

本章首先对矩阵函数以及流形等基本概念进行了介绍, 在此基础上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, 最后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。

第三章 黎曼流形上的偏最小二乘回归

在1.2节中已经提到，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该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会涉及到一种特殊的数据结构——对称正定 (Symmetric Positive Definite, SPD) 矩阵：在单统计量建模图像集合的工作 [15] 和 [17] 中均使用样本协方差 (Covariance) 建模图像集合；在多统计模型建模图像集合的工作 [18] 和 [19] 中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（根据信息几何的内容 [43]，高斯分布在适当的结构定义下构成黎曼流形，且该流形与 SPD 矩阵流形关系十分密切，详细的内容读者可以阅读文献 [43] 做进一步的了解）。分布函数建模图像集合的工作 [20] 也利用了 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进行建模。由此可见黎曼流形，特别是 SPD 矩阵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) 图像的研究（如工作 [37], [38], [44], [41] 等），它们为后续用 SPD 矩阵流形研究图像集合奠定了基础。同时工作 [44] 和 [41] 则更进一步的将欧氏空间中的两个有效数据分析方法：主成分分析 (Principle Component Analysis, PCA) 和典型相关分析 (Canonical Correlation Analysis, CCA) 扩展到了黎曼流形上；这启示我们将与 PCA 和 CCA 关系十分密切的偏最小二乘回归 (Partial Least Square Regression, PLSR) 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更稀少但是维度却很高）对其进行改进，将其适配到图像集合分类问题上。

这里将接下来的内容安排如下：首先花一些篇幅介绍一下 (Partial Least Square, PLS)，然后是黎曼流形中的投影的概念，接着是流形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偏最小二乘问题，紧接着是针对图像集合问题的黎曼流形上的偏最小二乘方法的适配和改进，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，其中还会介绍非线性迭代偏最小二乘算法 (Nonlinear Iterative Partial Least Squares algorithm, NIPALS[45])，它至今仍然是求解偏最小二乘的有力工具。此外还会对偏最小二乘方法用于分类场景做简要介绍，为后续的分类问题做准备，在本节的最后简要讨论 CCA 与 PLS 的关系。

偏最小二乘法于 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 [46–50] 都是其中具有代表性的工作。偏最

小二乘方法也被称为第二代回归方法，它是对传统的多元线性回归模型的一种扩展。它具有主成分分析，典型相关分析以及回归分析的特点，因而往往能给出更加合理的多元数据分析模型。特别地，偏最小二乘方法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偏最小二乘方法仍然可以对数据进行很好的建模，下面对偏最小二乘的数学形式进行简要的描述。

假设两个数据集合 $X \in \mathbb{R}^N$ 和 $Y \in \mathbb{R}^M$ ，并且有分别来自两个集合的 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，偏最小二乘通过公式3-1中的得分向量 (T, U) 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非线性迭代偏最小二乘算法 (Nonlinear Iterative Partial Least Squares algorithm, NIPALS[45]) 是最具代表性的算法之一，也是偏最小二乘问题求解中最常用的算法之一。该算法通过迭代寻找向量 w, c 使得： $[\text{cov}(Xw, Yc)]^2$ 最大化，即：

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

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（从上到下，从左到右）：

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

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) 当用于分类问题时偏最小二乘的标签数据集是怎样表示的。这两个问题在 [50] 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。对于第一个问题，通常的做法是在每次计算新的得分向量的时候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 $y_i \rightarrow \mathbf{p}^{y_i}$ 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，那么原本的标签向量就转换为如3-5所示的形式。

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

其中 n_1, n_2, \dots, n_C 表示的是各个类别的样本数，且有 $n = \sum_{i=1}^C n_i$ 。为了帮助理解这里给出欧式空间中的偏最小二乘回归的示意图3.1。

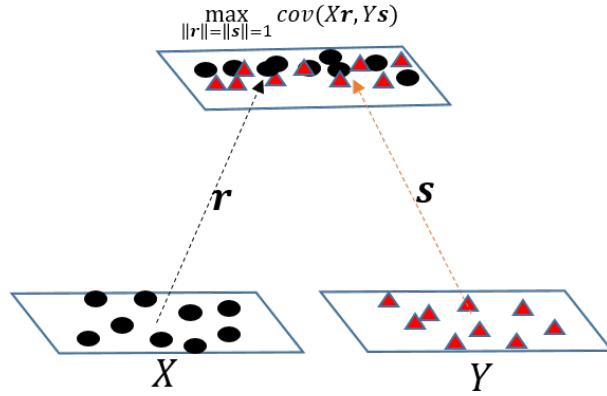


图 3.1 欧式空间偏最小二乘回归示意图

最后，这里简单介绍 PLS 与 CCA 的关系，更多关于 PLS, PCA, CCA 之间的关系可以从 [50] 中获得；CCA 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；两者名字虽然差别较大但是形式上却很相近，在一些情况下两者甚至是等价的。它们的相似性从其数学形式就可以看出来：

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

上两式的联系，从 1976 年 H. D. Vinod 关于“canonical ridge analysis”的论文 [51] 中给出的公式3-7即可看出一二。

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

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：

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

总结上述内容得到：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协方差，而 CCA 最大化的是投影后样本表示的相关系数，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 的时候两者等价；此外两者的解都与特征值问题相关，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

简单回顾欧氏空间中的 PCA, CCA 和 PLS 不难发现其中都涉及到投影的概念，它是这一类方法的核心，也是本节的主要讨论对象。这一节的内容主要分为两部分：第一部分从欧氏空间中的投影开始，逐步介绍抽象的投影的概念，然后将这个概念推广到黎曼流形；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氏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 中的投影概念是相似的，所以这里就直接以 PLS 为载体进行欧氏空间中的投影的概念的解释和一般化介绍。

3.1节给出了欧氏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，如公式3-9所示。

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

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直接应用到流形上，因为流形并没有内积的定义。为此，这里先回顾一下欧氏空间中投影的更一般的形式。

高维空间向低维空间的投影的物理意义是在低维空间中找到高维空间中的原始数据最接近的表示。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3-10所示的优化问题得到 ($d(\cdot, \cdot)$ 为距离函数)。

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

特别地，若 S_k 是一维子空间，即 S_k 是由一个向量张成的子空间 $S_1 = \{\mathbf{v} | \mathbf{v} = t\mathbf{w}, t \in \mathbb{R} \text{ and } \mathbf{w} \in \mathbb{R}^n\}$ ，则投影变换定义如3-11所示。

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

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 $S_1^x = \{\mathbf{v} | \mathbf{v} = t\mathbf{r}, t \in \mathbb{R} \text{ and } \mathbf{r} \in \mathbb{R}^n, \|\mathbf{r}\| = 1\}$, $S_1^y = \{\mathbf{v} | \mathbf{v} = u\mathbf{s}, u \in \mathbb{R} \text{ and } \mathbf{s} \in \mathbb{R}^n, \|\mathbf{s}\| = 1\}$ 使得投影后的数据表示具有最大的协方差。

3.2.2 对称正定矩阵流形上的均值

本节将就 SPD 矩阵流形为例，介绍其上的投影变换；此外总结欧氏空间中的偏最小二乘方法，不难发现，若要将偏最小二乘推广到黎曼流形需要完成四件事：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) 最大化协方差。本节的内容主要完成的是前两件事：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简要回顾 SPD 上的 Karcher mean 问题。由于在本文的第二章中，针对该问题已有详细的介绍和讨论，所以这里我们直接给出它计算的数学形式。

$$\bar{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2n} \sum_{i=1}^n \delta^2(\mathbf{x}, \mathbf{x}_i)$$

这里的 $\delta(\cdot, \cdot)$ 表示的是流形中的测地距离，在 PSD 流形中常使用也是最基本的就是仿射不便距离 (Affine Invariant Distance, AID[37])，将 AID 带入到上式的话得到：

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

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 Karcher mean 是一个矩阵，这里仍然用 μ 来表示。第二章已经介绍过，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，在第二章中我们有如下的结果：

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

将上式带入到 3-12 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_k 代表步长）。

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

3.2.3 黎曼流形上的子流形与投影

在3.2.3节介绍了黎曼流形上的均值的计算，这节将会介绍黎曼流形上高维流形向低维流形的投影的过程，此前在文献 [41,44] 中对该问题就有一些介绍，这里会对之前工作做一个总结，为黎曼流形上的偏最小二乘推广做准备。

首先，根据欧氏空间中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，它是欧氏空间中两点之间直线最短概念的推广，因此利用测地线构造子流形是自然的一种途径。一般利用样本均值（Karcher mean）出发的测地线构造这样的子流形，虽然目前还没有理论证明这样的构造是最优的，但是一部分在对称正定矩阵流形上的实验结果验证在这点可以得到不错的结果 [52]。

将从 Karcher mean 出发利用测地线构造的子空间记为 S_W ，带入到公式3-11中并使用测地距离 $\delta(\cdot, \cdot)$ 可得到公式3-16的形式。

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

更具体地，当把上述理论运用到对称正定矩阵流形的时候（使用 Affine Invariant Distance[37]），可以更具体的得到公式3-17的形式。

$$\begin{cases} S_W = \exp_\mu(\text{span}(W) \cap U) \\ \Pi_{S_W}(X) = \arg \min_{X' \in S_W} \delta^2(X', X) \\ = \arg \min_{X' = tW} \|\log_{[\exp_\mu(tW)]}(X)\|^2, t \in (-\epsilon, \epsilon) \\ t^* = \pi_{S_W}(X) = \arg \min_{t \in (-\epsilon, \epsilon)} \|\log_{[\exp_\mu(tW)]}(X)\|^2 \end{cases} \quad (3-17)$$

其中 U 是切空间中原点的一个小领域， W 是 Karcher mean 的切空间中从原点出发的切向量，类似于欧氏空间中数据中心化后的空间中的投影方向；公式3-17即包含了子流形的构造以及原始流形向子流形的投影。为了方便理解这里使用示意图3.2 帮助理解。

3.3 黎曼流形上的偏最小二乘回归问题

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做进一步的改进，使其更加适配到图像集合分类问题上。

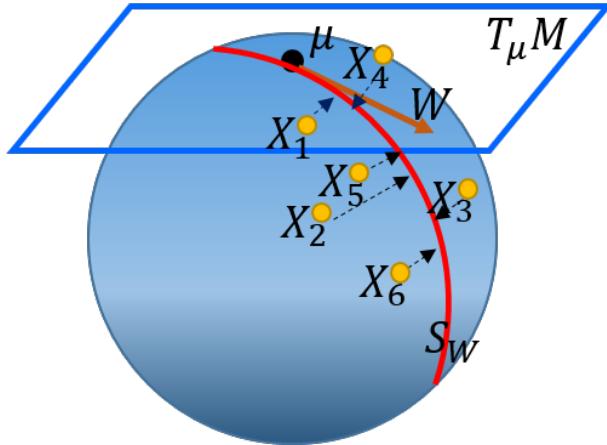


图 3.2 流形上的投影示意图

3.3.1 黎曼流形上偏最小二乘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 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：

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

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 \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：

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

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，也是定义了流形中的投影之后所能得到的最直接的形式化；但是不难发现还有如下一些问题遗留：1) 由于公式3-19没有解析解，上述问题需要大量的计算，消耗时间过长是一个问题，究其原因主要是 $t_i = \arg \min_{t \in (-\epsilon, \epsilon)} \|\log_{[\exp_{\mu_X}(tW_X)]}(X_i)\|^2; i = 1, 2, \dots, n;$ 和 $u_i = \arg \min_{u \in (-\eta, \eta)} \|\log_{[\exp_{\mu_Y}(uW_Y)]}(Y_i)\|^2; i = 1, 2, \dots, n;$ 的计算过于复杂。2) 如何像欧氏空间中的

偏最小二乘一样计算第二个（或更多个）投影方向；3) 如何将上述问题用于分类问题（或者说如何做回归问题）。在接下来的内容我们将会发现：这些问题中最主要的问题还是原问题没有解析解的问题，当通过适当近似手段将原问题简化后，上面列出的几个问题都将迎刃而解。

前面已经提到优化问题3-19的计算复杂度过高，并且该问题在图像集合分类问题上会更加严重，原因是数据的维度太高；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 [41,44] 中已有类似的介绍，主要是针对 $t_i = \arg \min_{t \in (-\epsilon, \epsilon)} \|\log_{[\exp_{\mu_x}(tW_X)]}(X_i)\|^2; i = 1, 2, \dots, n;$ 和 $u_i = \arg \min_{u \in (-\eta, \eta)} \|\log_{[\exp_{\mu_y}(uW_Y)]}(Y_i)\|^2; i = 1, 2, \dots, n;$ 的计算复杂度过高（或者说没有解析解）的问题而提出的简化方案。

以 $t_i = \arg \min_{t \in (-\epsilon, \epsilon)} \|\log_{[\exp_{\mu_x}(tW_X)]}(X_i)\|^2$ 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 $\exp_{\mu_x}(\cdot)$ 变换到 \mathbb{S}_d^+ 中，然后在 \mathbb{S}_d^+ 中度量 $\exp_{\mu_x}(tW_X), X_i$ 两者的测地距离 $\delta(\exp_{\mu_x}(tW_X), X_i)$ ；据此并参考 [44]，我们使用如下的近似方案：将在原流形中度量 $\exp_{\mu_x}(tW_X), X_i$ 两者的距离改在 μ_x 的切空间中度量两者对应切向量的距离 $\delta(\exp_{\mu_x}(tW_X), X_i) \approx \|tW_X - \log_{\mu_x}(X_i)\|_{\mu_x}$ ，于是原来的投影问题3-17就变成了：

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

其中 $\{V_i\}_{i=1}^d$ 是 μ_x 处的切空间中的标准正交基 (μ_x 的切空间是内积空间)，这点修改使得问题大大简化，稍作整理之后可以得到对称正定矩阵流形上的偏最小二乘近似算法5。

算法5中的：“在 $\{\tilde{X}_i\}_{i=1}^n$ 以及 $\{\tilde{Y}_i\}_{i=1}^n$ 之间执行 NIPALS[45] 得到 $\hat{W}_X, \hat{W}_Y, T = [\mathbf{t}_1, \dots, \mathbf{t}_n], U = [\mathbf{u}_1, \dots, \mathbf{u}_n]$ ” 使用的是欧氏空间中的标准的偏最小二乘求解算法，因为 Karcher mean 的切空间是内积空间，而通过平行移动（Parallel transport[41]）后的数据在单位阵 I 处的表示也是内积空间中的表示，所以欧氏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）和偏最小二乘回归问题的形式化，由于这只是重复前面的内容，这里就不再赘述了，最后在算法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偏最小二乘回归

这里将算法6称为基础版本的黎曼流形上的偏最小二乘回归，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 的切空间中原点的一个小邻域内，而其本质上是要求 SPD 矩阵样本的分布是比较集中；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偏最小二乘回归用到图像集合需要解决的问题；此外

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 $\mathbf{X} = \{X_i\}_{i=1}^n, \mathbf{Y} = \{Y_i\}_{i=1}^n$, 需要计算的成分的个数 k

Ensure: 在集合 \mathbf{X} 的 Karcher mean (μ_x) 处的切空间中张成子空间的 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 $\{\mathbf{t}_i\}_{i=1}^n$; 在集合 \mathbf{Y} 的 Karcher mean (μ_y) 处的切空间中张成子空间的 k 个成分 $\mathbf{W}_Y = \{W_Y^i\}_{i=1}^k$, 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: 利用群操作将样本变换到单位矩阵的切空间:

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

$$\log_{\mu_y}(Y_i) \rightarrow \mu_y^{-1/2} \log_{\mu_y}(Y_i) \mu_y^{-1/2} = \log(\mu_y^{-1/2} Y_i \mu_y^{-1/2}) \triangleq \tilde{Y}_i$$

3: 在 $\{\tilde{X}_i\}_{i=1}^n$ 以及 $\{\tilde{Y}_i\}_{i=1}^n$ 之间执行 NIPALS[45] 得到 $\hat{\mathbf{W}}_X, \hat{\mathbf{W}}_Y, T = [\mathbf{t}_1, \dots, \mathbf{t}_n], U = [\mathbf{u}_1, \dots, \mathbf{u}_n]$

4: 利用平行移动 (Parallel transport[41]) 将 $\hat{\mathbf{W}}_X$ 变换到 μ_x 的切空间得到 \mathbf{W}_X , 将 $\hat{\mathbf{W}}_Y$ 变换到 μ_y 的切空间得到 \mathbf{W}_Y

5: **return** $\mathbf{W}_X, \mathbf{W}_Y, T, U, \mu_x, \mu_y$

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 $\mathbf{X} = \{X_i\}_{i=1}^n$, 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 $\{\mathbf{t}_i\}_{i=1}^n$; 欧氏空间中标签集 Y 的投影矩阵 $\mathbf{W}_y = \{\mathbf{w}_y^i\}_{i=1}^k$ 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: 利用平行移动 (Parallel transport[41]) 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 NIPALS[45] 得到 $\hat{\mathbf{W}}_x, \mathbf{W}_y, T = [\mathbf{t}_1, \dots, \mathbf{t}_n], U = [\mathbf{u}_1, \dots, \mathbf{u}_n]$

4: 利用平行移动 (Parallel transport[41]) 将 $\hat{\mathbf{W}}_x$ 变换到 μ 的切空间得到 \mathbf{W}_x

5: **return** $\mathbf{W}_x, \mathbf{W}_y, T, U, \mu_x, \mu_y$

注意到工作 [41,44] 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, 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偏最小二乘回归方法适配到图像集合分类上, 并且我们尽量做到在几乎不增加计算复杂度的前提下对性能有较大的改进。

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 更具判别性的点构建子流形进行介绍。我们的解决方案具体如下: 首先注意到公式3-12计算 Karcher mean 的

时候是一个无监督的优化过程，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算法6中的 $\{X_i\}_{i=1}^n$ 在 μ 处的切向量表示经过平行移动（Parallel transport[41]）变换到单位阵的切空间中之后的 $\{\tilde{X}_i\}_{i=1}^n$ 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 ($T_\mu M$) 中应该具有 0 均值的特点）。

沿用之前的记号设 $\{X_i\}_{i=1}^n$ 是来自于 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，且 $n = \sum_{i=1}^C n_i$ ，样本的 label 我们用公式3-5的矩阵 Y 表示，下面是编码判别信息的形式化过程。

- 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：

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

- 定义矩阵 $P' = 1 - 2YY^T$ ，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，这里做一下平衡操作：设 $n_{\#1}=\{P'\text{ 中 }1\text{ 的个数}\}$, $n_{\#-1}=\{P'\text{ 中 }-1\text{ 的个数}\}$ 则定义平衡后的矩阵为 P ，其中：

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

- 根据工作 [44]，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，同时也有平衡量纲的作用，避免其中一方因量纲问题绝对占优。

3.3.2.2 融合判别信息的子流形构造问题的优化

公式3-23没有解析解因此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 \mathbb{S}_d^+ ，这里使用黎曼流形上的共轭梯度算法2来优化问题3-23，关于黎曼流形上的共轭梯度算法可以参看本文的第二章以及文献 [53]，使用该方法优化的时候，最主要的输入是3-23的导数。接下来将利用第二章的内容对公式3-23的导数计算作简要说明。

公式3-23主要包含两部分： $\text{tr}(FP^T)$ 和 $\frac{\lambda}{n} \sum_{i=1}^n \|\log_\mu(X_i)\|^2$ ；这里先对第一部分的导数进行计算：

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

公式3-24表明导数计算的核心是 $\frac{\partial}{\partial \mu} k_{ij}$ ，因此接下来主要是 k_{ij} 的导数计算。此前先对 k_{ij} 的形式稍加变换，方便理解后面的 k_{ij} 的导数的形式（公式3-26）：

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

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二章以及文献 [33,54] 的相关章节。

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

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 $\|\log_\mu(X_i)\|^2 = \langle \log_\mu(X_i), \log_\mu(X_i) \rangle_\mu = k_{ii}$ ，因此这部分的导数可以根据3-26直接导出来。

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

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

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

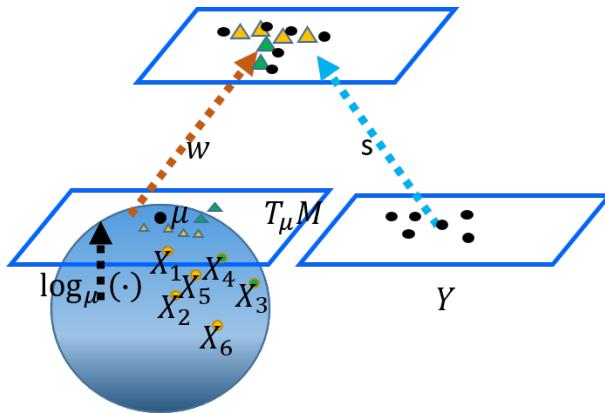


图 3.3 算法7的示意图

最后,利用第二章中的内容以及论文[33,54]中的结果将普通的梯度转换为黎曼梯度。这只需要对公式3-28做公式3-29中的变化即可:

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

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本文所提算法的最终形式,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。

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 $X = \{X_i\}_{i=1}^n$, label矩阵 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 X 在 $T_\mu M$ 中的 k 个成分 $W_X = \{W_X^i\}_{i=1}^k$,以及 $\{X_i\}_{i=1}^n$ 对应的投影 $\{\mathbf{t}_i\}_{i=1}^n$; 欧氏空间中标签集 Y 的投影向量 $W_y = \{W_y^i\}_{i=1}^k$ 及其对应的投影 $\{\mathbf{u}_i\}_{i=1}^n$

- 1: 初始化 $\mu = \mu_0$ (通常为 I)求解问题3-23获得 μ ,然后计算 $\{\tilde{X}_i = \log_\mu(X_i)\}_{i=1}^n$
- 2: 利用平行移动(Parallel transport[41])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NIPALS[45]得到 $\hat{W}_x, W_y, T = [\mathbf{t}_1, \dots, \mathbf{t}_n], U = [\mathbf{u}_1, \dots, \mathbf{u}_n]$
- 4: 利用平行移动(Parallel transport[41])将 \hat{W}_x 变换到 μ 的切空间得到 W_x

- 5: **return** W_x, W_y, T, U, μ
-

最后,为了方便理解算法7这里使用利用示意图3.3对算法的思想进行说明。

3.3.2.3 多切空间逐步回归的偏最小二乘方法

注意到算法7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近似/估计不准问题,接下来的部分将针对这个方法做进一步的改进。公式3-19中的 $t_i \in (-\epsilon, \epsilon)$ 和 $u_i \in (-\eta, \eta)$ 的条件就是要求样本数据需要足够集中才行,但是如果直接将算法7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, 而最后通过设计特定的方法整合每个切空间中的表示模型得到更具表示能力的模型帮助判别(分类)问题的研究。

上述框架的主要问题是如何选取各个切空间以及如何将它们有效的融合起来, 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7中的数据 $\mathbf{X} = \{X_i\}_{i=1}^n$ 和 label 矩阵 Y 执行算法7获得一个切空间 $T_{\mu_1}M$, 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 $\{\tilde{X}_i^n\}_{i=1}^n$ 以及 Y 利用公式3-4进行 defalte 操作: $\{\tilde{X}_i\}_{i=1}^n \xrightarrow{\text{defalte}} \{\tilde{X}_i^{res}\}_{i=1}^n, Y \xrightarrow{\text{defalte}} Y^{res}$, 然后对 $\{\tilde{X}_i^{res}\}_{i=1}^n$ 利用平行移动 (Parallel transport[41]) 从 T_1M 处将数据变换到 $T_\mu M$ 再用 $\exp_{\mu_1}(\cdot)$ 将数据 $\{\tilde{X}_i^{res}\}_{i=1}^n$ 变换到 SPD 矩阵流形得到 $\{Z_i^{res}\}_{i=1}^n$, 最后将 $\{Z_i^{res}\}_{i=1}^n, Y^{res}$ 赋值给 $\mathbf{X} = \{X_i\}_{i=1}^n$ 和 Y , 并重新初始化 μ_2 后开始第二次迭代; 反复上述过程直到获得指定个数的切空间, 算法终止。这里将上述过程用算法8描述。

Algorithm 8 对称正定矩阵流形上多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 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 \mathbf{X} 在 $T_{\mu_1}M, \dots, T_{\mu_p}M$ 中各自的 k 个成分 $\mathbf{W}_x^{(1)}, \dots, \mathbf{W}_x^{(p)}$, 以及对应的投影 T_1, \dots, T_p ; 欧氏空间中标签集 Y 逐次回归的投影矩阵 $W_y^{(1)}, \dots, W_y^{(p)}$ 及其对应的投影 U_1, \dots, U_p

- 1: 初始化 $output$ 为包含 p 个 cell 的结构: $output = cell(1, p)$
 - 2: **for** $j = 1; j \leq p; j = j + 1$ **do**
 - 3: 初始化 $\mu_j = \mu_0$ (通常为 I) 求解问题3-23获得 μ_j , 然后计算 $\{\hat{X}_i = \log_{\mu_j}(X_i)\}_{i=1}^n$
 - 4: 利用平行移动 (Parallel transport[41]) 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 NIPALS[45] 得到:

$$\hat{\mathbf{W}}_x^{(j)}, \hat{\mathbf{W}}_y^{(j)}, T_j = [\mathbf{t}_1^{(j)}, \dots, \mathbf{t}_n^{(j)}], U_j = [\mathbf{u}_1^{(j)}, \dots, \mathbf{u}_k^{(j)}]$$
 - 6: 利用平行移动 (Parallel transport[41]) 将 $\hat{\mathbf{W}}_x^{(j)}$ 变换到 μ_j 的切空间得到 $\mathbf{W}_x^{(j)}$
 - 7: 对 \mathbf{X}, Y 使用 defalte 操作: $\{\tilde{X}_i\}_{i=1}^n \xrightarrow{\text{defalte}} \{\tilde{X}_i^{res}\}_{i=1}^n, Y \xrightarrow{\text{defalte}} Y^{res}$
 - 8: 利用平行移动 (Parallel transport[41]) 从单位阵处将数据 $\{\tilde{X}_i^{res}\}_{i=1}^n$ 变换到 $T_{\mu_j}M$ 然后用 $\exp_{\mu_j}(\cdot)$ 将结果变换到 SPD 矩阵流形得到 $\{Z_i^{res}\}_{i=1}^n$
 - 9: 将 $\{Z_i^{res}\}_{i=1}^n, Y^{res}$ 赋值给 $\mathbf{X} = \{X_i\}_{i=1}^n$ 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** $output$
-

为了帮助理解，与前面类似，这里同样用一个示意图对算法8的核心进行描述说明，具体如图3.4所示。

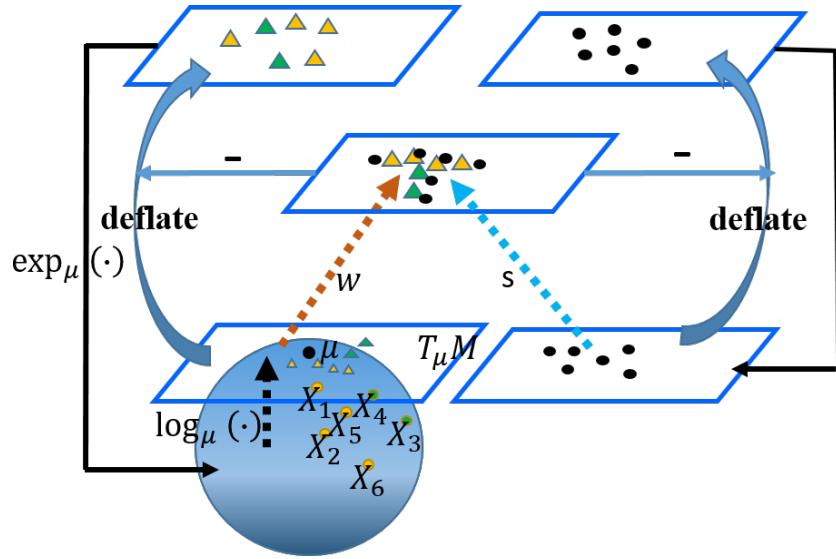


图 3.4 算法8的示意图

在训练数据上运用算法8得到训练参数后，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 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与算法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[1]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）且实验中若减小 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）；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。最后数据是否中心化对最终结果也有一定的影响，这个问题在UIUC[29]数据上做材质分类的时候尤其明显。

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。

本实验主要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[2]），材质分类（数据库UIUC[29]）以及视频人脸识别（数据集YTC[1]）；由于这些数据集已经在1.3节进行了介绍，并且这些数据集的测试协议也已经在1.3节中介绍，所以这里不再进行阐述，如果读者对数据或测试协议有什么不明的话可以到1.3一节进行查看。

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，这些底层特征的提取将用于最后的SPD矩阵（实际上也可看作是一种特征表示）的构造。首先，在

用于物体识别的 ETH80[2] 数据集上，所有的图片被预先 resize 成 20×20 的图片，然后灰度特征被直接用于物体识别任务；在材质分类任务的 UIUC[29] 数据集上，这里使用 Region Covariance[31] 表示一张图片。参考 [17]，这里的 Region Covariance 的构造中我们使用 128 维的 dense SIFT[55] 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，但与之不同的是工作 [17] 还融合了颜色特征。最后在视频人脸识别任务的数据库 YTC[1] 上，图片被 resize 到 20×20 并进行直方图均衡化操作后像素被用作底层特征。

3.4.2 对称正定矩阵表示的构造

在3.4.1节中介绍了基本的图像特征的提取和预处理方式，本节将对使用这些基本特征构造 SPD 矩阵表示做简要介绍：首先是材质分类任务的 UIUC[29] 数据集上的 SPD 表示，由于这里使用的是标准的 Region Covariance[31] 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 [31]；在物体识别的数据集 ETH80[2] 和视频人脸识别的数据集 YTC[1] 上，这里根据工作 [15] 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 [19,20] 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， \mathbf{m} 是样本均值， d 是样本的维数）：

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

最后还需要一提的是：所有的原始特征在实验中都做了 95% 的 PCA 降维预处理；当样本协方差 Σ 奇异的时候，根据 [20,56] 中的做法，一个小的正的正则项： $\delta\mathbf{I}, \delta = 10^{-3} \times \text{tr}(C)$ (\mathbf{I} 是单位阵) 被加到 Σ 上： $\Sigma + \delta\mathbf{I} \rightarrow \Sigma$ 。

3.4.3 实验结果与分析

在本实验中，我们将本文所提的方法统一用缩写 RPLS (Riemannian Partial Least Squar regression, 黎曼偏最小二乘) 表示，本小结的内容是 RPLS 方法在物体识别，材质分类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：基于偏最小二乘的协方差判别学习方法 (Covariance Discriminant Learning, CDL)[15]，黎曼稀疏编码学习的方法 (Riemannian Sparse Representation, RSR[57])，然后是发表在 2014 年欧洲计算机视觉会议 (European Conference On Computer Vision, ECCV) 上的工作：对称正定矩阵流形学习 (SPD-Manifold Learning, SPDM[17]) 在两种度量 (Stein Divergence[58] 和 Affine Invariant Metric[37]) 下的方法，使用分布函数建模集合的方法 DARG[20] 和 BeyondGauss[21] 以及 SPD 矩阵流形上的度量学习 (Metric Learning) 方法 LEML[56]。表3.1给出了这些方法在三个任务上的实验对比结果，其中所有的结果均是按照1.3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1.3节的协议下所获得的最好的结果。

表 3.1 黎曼流形上的偏最小二乘回归算法实验结果

数据集 方法	ETH80	UIUC	YTC
CDL-PLS[15]	93.25±4.72	53.89±4.06	70.28±2.13
RSR-Stein[57]	93.25±3.34	52.41±4.03	72.77±2.69
SPDML-Stein[17]	90.50±3.87	49.17±2.37	61.57±3.43
SPDML-AIM[17]	90.75±3.34	48.09±1.82	64.66±2.92
BeyondGauss[21]	84.75±6.29	N/A	71.46±2.61
DARG[20]	92.25±2.19	N/A	77.09±1.92
LEML[56]	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，RPLS 算法的下标表示的基础版本的黎曼流形上的偏最小二乘回归算法7 (RPLS_{single}) 还是多切空间逐步回归的算法8 (RPLS_{multi})。最终的实验结果验证了我们最初的猜想，多切空间偏最小二乘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结为（与 CDL[15] 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 与在单位阵切空间中的 RPLS_{single} 是等价的）；3) 最后是单切空间与多切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 对于算法的影响较大，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[21] 的方法由于是使用 KDE 来估计分布函数，而在 UIUC[29] 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)，该方法在其它数据集上的结果是在 hellinger 散度下的结果。同样由于内存问题 DARG[20] 方法在 UIUC[29] 数据集上也无法获得结果，所以也使用 N/A 代替。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 [15,37,41,44] 等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到6中，并针对图像集合分类问题与 DTI (Diffusion Tensor Images) 的不同

(主要是数据更稀疏的问题), 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的小范围内这些改进依然是适用的) 得到了多切空间偏最小二乘回归算法, 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分析部分都提到, 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 [52] 中的方法使用 Adaboost 的框架进行多个切空间中模型的组合。

第四章 低秩对称半正定矩阵判别学习方法

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模型都是与样本协方差矩阵有关，但样本协方差矩阵的计算中一个很现实的问题是样本数往往小于数据维度，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；这不管是对多视角图像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）中的视频都是很苛刻的要求。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, PSD）矩阵。在使用样本协方差矩阵建模图像集合的方法中，针对该问题的一个常用的 trick 是给样本协方差矩阵的对角线上加上一个正则项（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，这促使本章回到数据所在的原始集合——对称半正定矩阵集合中研究图像集合的建模和判别学习问题。

另一方面，当使用对称正定矩阵表示图像集合的时候，对称正定矩阵的维度往往非常高，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 [17,56] 研究了对称正定矩阵流形的降维问题；而不难发现的是当图像集合使用的对称半正定矩阵表示且要求矩阵的秩（Rank）很低的时候对称半正定矩阵表示将体现出另一个重要的性质：低秩（Low-Rank）。低秩的性质将大大的降低存储容量和计算时间，因为如果将对称半正定矩阵 C 进行 $C = WW^T, W \in \mathbb{R}^{d \times k}, \text{rank}(W) \leq k \ll d$ 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，而这正是我们想要的。

最后，回顾图像集合建模的两大分支：子空间的方法（如 [7,8] 等）和统计模型建模的方法（如 [15–20] 等），可以发现如下事实：对于样本协方差矩阵 C_i ，及其特征分解 $C_i = U_i \Lambda_i U_i^T$ ；子空间的方法只用到了 U_i 的信息，还有尺度（特征值）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，但是只有很少的文章考虑了数据中的噪声和样本稀疏带来的估计不准的问题，而因此可能导致模型不够鲁棒的问题。最后反观低秩对称半正定矩阵表示中当 $k < d$ 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：首先结合着前期关于固定秩对称半正定矩阵流形的研究，对与低秩对称半正定 (Low-Rank symmetric Positive Semi-Definite, Low-Rank PSD) 矩阵关系最密切的固定秩对称半正定 (Fixed-Rank symmetric Positive Semi-Definite, Fixed-Rank PSD) 矩阵流形进行介绍，其中为了介绍 Fixed-Rank PSD 矩阵流形还会用一些篇幅简要介绍一下 Stiefel Manifold 和 Grassmann Manifold；然后结合着工作 [32] 针对 Fixed-Rank PSD 矩阵流形建模图像集合中存在的一些问题以及 Low-Rank PSD 矩阵表示图像集合的

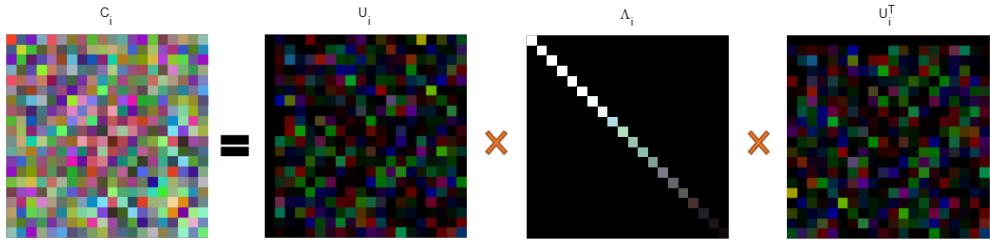


图 4.1 特征值分解示意图

优势提出了 Low-Rank PSD 矩阵建模图像集合的方法；接着是 Low-Rank PSD 矩阵图像集合表示下的 PSD 编码和判别学习方法，最后是实验验证本章所提方法和以及本章内容的总结与展望。

4.1 施蒂费尔流形和格拉斯曼流形

这部分会对对称正定矩阵流形以外的两种流形进行介绍，分别是施蒂费尔 (Stiefel) 流形和格拉斯曼 (Grassmann) 流形，这里之所以同时介绍这两种流形主要是出于以下考虑：其一是因为 Grassmann 与 Stiefel 流形的关系密切使得两者需要同时介绍，其二是为 Fixed-Rank PSD 流形的介绍做准备。本节的内容只是对 Stiefel 流形和 Grassmann 流形的简要介绍，更多关于 Grassmann 流形以及两者的关系的内容可以参看文献 [59]。以下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) Grassmann 流形是定义在 \mathbb{R}^n 中所有 k 维的线性子空间构成的集合上的拓扑结构。

$$\text{Gr}(k, n) = \{\mathbb{V} \subset \mathbb{R}^n, \mathbb{V} \text{ is a linear subspace with } \dim \mathbb{V} = k\} \quad (4-1)$$

当为其定义拓扑结构之后，即可构成流形结构，进一步的可以在其上导出黎曼度量 [59]，所以它是黎曼流形的特例。我们最终希望借助 Grassmann 流形和对称正定矩阵流形来表示固定秩对称半正定矩阵流形，为此需要了解 Grassmann 流形的数学表示，而 Grassmann 流形的表示又借助于 Stiefel 流形，所以出于数学表示的方便性的考虑，这里先对 Stiefel 流形进行介绍。

定义 4.2 (Stiefel(non-compact) 流形) non-compact Stiefel 流形^① 是定义在所有 $n \times k, (0 < k < n)$ 满秩矩阵构成的集合上的流形结构。

$$\begin{aligned} \text{St}(k, n) &= \{A \in \mathbb{R}^{n \times k}; \text{rank}(A) = k\} \\ &= \{A = (a_1, \dots, a_k) \in \mathbb{R}^{n \times k}; a_1, \dots, a_k \text{ are linearly independent}\} \end{aligned} \quad (4-2)$$

^① 以下所说的流形均指集合上定义了拓扑结构的流形，为简洁起见文中仅以集合代替，而不细说拓扑结构

定义 4.3 (Stiefel(compact) 流形) 当在 non-compact Stiefel 流形^① 的定义域中要求 A 的列向量是正交的（即： $A^T A = I_k$ ）时候，即构成了 compact Stiefel 流形定义的集合。

$$\text{St}^*(k, n) = \{A \in \mathbb{R}^{n \times k}; A^T A = I_k\} \quad (4-3)$$

这里再给出一个在数学中非常重要的概念：“广义线性群 (General Linear Group, GL)”，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 $n \times n$ 的可逆矩阵的集合称为广义线性群，并记为：

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

至此，介绍 Grassmann 流形的准备工作已基本完成，接下来将利用这些定义以及2.1.1节的内容对 Grassmann 流形的数学表示进行介绍。为此这里先来捋一捋 Grassmann 流形和 Stiefel 流形的关系：

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

$$\text{GS} : \text{St}(k, n) \rightarrow \text{St}^*(k, n) \quad (4-5)$$

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

关系 4.2 (Stiefel(non-compact) V.S Grassmann) 定义如公式4-6所示的变换 π 描述两者之间的关系。

$$\pi : \text{St}(k, n) \rightarrow \text{Gr}(k, n), A = (a_1, \dots, a_k) \rightarrow \text{span}(A) \quad (4-6)$$

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 $A \in \text{St}(k, n)$ 有

$$\pi^{-1}[\pi(A)] = \{AP; P \in \text{GL}(k)\} \quad (4-7)$$

- 映射 π 是连续 (continuous) 的开 (open) 的 (“开”的意思是说映射的像是开集的话原像也是开集)

关系 4.3 (Stiefel(compact) V.S Grassmann) 类似于关系4.2，定义 Stiefel(compact) 和 Grassmann 之间的映射 $\bar{\pi}$ 。

$$\bar{\pi} : \text{St}^*(k, n) \rightarrow \text{Gr}(k, n); A = (a_1, \dots, a_k) \rightarrow \text{span}(A), A^T A = I_k \quad (4-8)$$

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

前面给出了诸多定义和关系, 其目的主要是为了给出 Grassmann 流形的一个数学表示, 方便对其进行研究和应用。

表示 4.1 利用 non-compact Stiefel 流形将 Grassmann 流形表示为如下的商空间 (quotient space) 的形式

$$St(k, n)/GL(k, \mathbb{R}) = \{[A]|[A] \triangleq A[GL(k)]; A \in St(k, n)\} \quad (4-9)$$

表示 4.2 Grassmann 流形也可利用商空间的概念使用 compact Stiefel 流形表示:

$$\begin{aligned} St^*(k, n)/O(k) &= \{[A]|[A] \triangleq A[O(k)]; A \in St^*(k, n)\} \\ O(k) &= \{U|U \in \mathbb{R}^{k \times k}; U^T U = I_k\} \end{aligned} \quad (4-10)$$

在两种表示中, 第二种表示方法更为常用, 对该表示的研究也相对成熟一些。在这一小节的最后将介绍 Grassmann 流形上的度量表示。为此来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, 定义子空间 $V_1 = \text{span}(X_1), V_2 = \text{span}(X_2)$ 之间的主夹角为:

$$\begin{aligned} \cos \theta_i &= \max_{\mathbf{u}_i \in V_1} \max_{\mathbf{v}_i \in V_2} \mathbf{u}_i^T \mathbf{v}_i \\ s.t. \quad &\mathbf{u}_i^T \mathbf{u}_i = 1, \mathbf{v}_i^T \mathbf{v}_i = 1 \\ &\mathbf{u}_i^T \mathbf{u}_j = 0, \mathbf{v}_i^T \mathbf{v}_j = 0; j \leq i \end{aligned} \quad (4-11)$$

其中的 $\theta_i, i = 1, 2, \dots, k$ 称为主夹角。

定义 4.6 (投影变换) 在 \mathbb{R}^n 空间中, 设 S_k 是其中的一个子空间, 并且 U 是子空间中的基矩阵 ($n \times k$), 则 \mathbb{R}^n 到 S_k 中的投影算子表示如4-12所示。

$$\Pi_k : U \rightarrow UU^T; U \in St^*(k, \mathbb{R}^n) \text{ and } UU^T \in \mathbb{S}_n \quad (4-12)$$

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“主夹角”和“投影变换”两个概念可方便的介绍 Grassmann 流形上两个常用的度量的: “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 $\{\theta_i\}_{i=1}^k$, 两者之间定义投影距离度量的定义如4-13所示。

$$d(X_1, X_2) = \|\Pi_k(X_1) - \Pi_k(X_2)\|_2 = \left(\sum_{i=1}^k \sin^2(\theta_i) \right)^{\frac{1}{2}} \quad (4-13)$$

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

定义 4.8 (比奈-柯西度量) 利用主夹角的 cosine 值 $\{\cos(\theta_i)\}_{i=1}^k$ 将该度量定义如下：

$$d(X_1, X_2) = \left(1 - \prod_{i=1}^k \cos^2(\theta_i)\right)^{1/2} \quad (4-14)$$

以上是本文关于 Grassmann 流形的简要介绍，更多关于 Grassmann 流形的内容可以参看文献 [59]，里面有更加详细的介绍，但是可能会比较晦涩。

4.2 固定秩对称半正定矩阵流形

本文用 $\mathbb{S}_d^+(k)$ 表示 $d \times d$ 秩为 k 的半正定矩阵组成的集合，它也是我们前期的主要研究对象。接下来将会有对固定秩的对称半正定矩阵 (Low-Rank symmetric Positive Semi-Definite, PSD) 流形的 Geomentry 结构做简要的介绍，主要内容参考 [60]。

对于任意的元素 $A \in \mathbb{S}_d^+(k)$ 将其分解为 $A = UR^2U^T$, $U \in \text{St}^*(k, d)$, $R \in \mathbb{S}_k^+$ 则这里给出如公式4-15所示的形式。

$$A = UR^2U^T = (UR)(UR)^T \triangleq ZZ^T; Z \in \text{St}(k, d) \quad (4-15)$$

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关系：

$$\begin{aligned} R &\sim O^T RO \in \mathbb{S}_k^+ \\ U &\sim UO \in \text{St}^*(k, d) \\ ZO &= URO \sim UOO^T RO \end{aligned} \quad (4-16)$$

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

$$(U, R^2) \sim (UO, O^T R^2 O) \in \text{St}^*(k, d) \times \mathbb{S}_k^+ \quad (4-17)$$

运用上述公式的内容来表示固定秩对称半正定矩阵 A 的时候，在其切空间 $T_A \mathbb{S}_d^+(k)$ 处的无穷小变量 (Δ, D) 定义为（具体形式可参看文献 [60]）：

$$\begin{aligned} \Delta &= U_\perp B, \\ D &= RD_0R, \end{aligned} \quad (4-18)$$

其中 $U_\perp \in \text{St}^*(d-k, d)$, $U^T U_\perp = \mathbf{0}$; $B \in \mathbb{R}^{(d-k) \times d}$, 而 $D_0 \in \mathbb{S}_d$ 。有了以上的定义后，进一步参考 $\text{St}^*(k, d)$ 和 \mathbb{S}_k^+ 上的黎曼度量定义两个微小变量之间的关系4-19。

$$\begin{aligned} g_{(U, R^2)}((\Delta_1, D_1), (\Delta_2, D_2)) \\ = \text{tr}(\Delta_1 \Delta_2) + \lambda \text{tr}(R^{-1} D_1 R^{-2} D_2 R^{-1}), \lambda > 0 \end{aligned} \quad (4-19)$$

可以证明，上式定义了切空间 $T_A \mathbb{S}_d^+(k)$ 上的黎曼度量 [60]。公式4-20利用固定秩对称半正定矩阵中的黎曼度量公式4-19定义了固定秩对称半正定矩阵 A, B 之间的一条曲线。

$$\left\{ \begin{array}{l} let : A \sim (U_A, R_A^2), B \sim (U_B, R_B^2) \\ PA \text{ of } U_A, U_B : \Theta = \text{diag}(\theta_1, \theta_2, \dots, \theta_k) \\ part1 : U(t) = U_A \cos(\Theta t) + X \sin(\Theta t) \\ part2 : R^2(t) = R_A \exp(t \log(R_A^{-1} R_B^2 R_A^{-1})) R_A \\ curve : \gamma_{A \rightarrow B}(t) = U(t) R^2(t) U(t)^T \\ length : L(\gamma_{A \rightarrow B}) = \|\Theta\|_F^2 + p \|\log(R_A^{-1} R_B^2 R_A^{-1})\|_F^2 \end{array} \right. \quad (4-20)$$

公式4-20中的 PA 是 Principle Angles 的缩写； $X = (I - U_A U_A^T) U_B F$ ，其中 F 是矩阵 $\text{diag}(\sin(\theta_1), \sin(\theta_2), \dots, \sin(\theta_k))$ 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流形上的测地距离则需要找到链接 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半正定矩阵流形过复杂，不过好在文献 [60] 证明的了 $L(\gamma_{A \rightarrow B})$ 是测地距离的一个不错估计（虽然它不一定满足三角不等式），所以接下来就可以运用该“距离度量”。

$$\delta_{FRPSD}^2(A, B) = \|\Theta\|_F^2 + p \|\log(R_A^{-1} R_B^2 R_A^{-1})\|_F^2 \quad (4-21)$$

4.3 固定秩对称半正定矩阵流形研究概况

本节的主要内容是介绍固定秩对称半正定（Fixed-Rank symmetric Positive Semi-Definite, Fixed-Rank PSD）矩阵流形上的判别学习方法。关于对称半正定矩阵 (symmetric Positive Semi-Definite, PSD) 的研究主要分为如下几个方向：比较纯粹的理论研究（如工作 [60,61] 等），然后是做优化的（如半正定规划问题）研究（如工作 [62]），再者是做度量学习的工作（如 AAAI'2016 的工作 [63]），最后是做图像集合分类的工作，目前据我们所知只有发表在“Winter Conference on Applications of Computer Vision (WACV), 2016”的工作 [32]，由于该工作与我们前期的研究内容非常相近并考虑到其前瞻性，接下来将简单的介绍一下，顺带介绍一下 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，优化与度量学习的分支将会在对工作 [63] 进行介绍的时候提及，所以这里不再赘述了，接下来就先看一下工作 [32]。

工作 [32] 针对本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。文中所提出的方法十分直接，遵循了解决图像集合分类问题的两个基本步骤：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 固定秩对称半正定矩阵表示图像集合

首先，不妨假设数据包含 n 个图像集合分别属于 C 个类别，并使用标签 $y_i \in \{1, 2, \dots, C\}$, $i = 1, 2, \dots, n$ 标记 (label)。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： $S_i = \frac{1}{n_i-1} \sum_{j=1}^{n_i} (\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)^T$
- 利用公式3-30融入均值信息（此时不再添加正则项）。
- 对样本协方差矩阵做特征值分解获得： $C_i = U_i \Lambda_i U_i^T$ ，并假设分解结果按特征值由大到小排列
- 对于给定的 rank = k ，选取 U_i 的前 k 列 $Y_i = U_i(:, 1:k) \in \text{St}^*(d, k)$ 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 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 [60]

通过上述的步骤为每一个图像集合构造 Fixed-Rank PSD 矩阵的表示，为了方便理解这里使用图4.2对 Fixed-Rank PSD 进行示意，而关于特征值分解的示意则可参看图4.1。

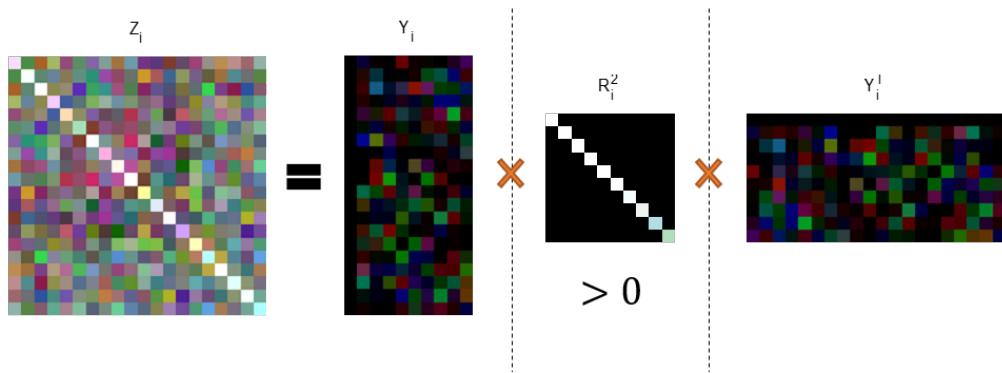


图 4.2 固定秩对称半正定矩阵示意图

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 固定秩对称半正定矩阵流形用于图像集合分类

在4.3.1一节根据 [60] 中介绍的 Fixed-Rank PSD 矩阵流形的 geometry 结构对图像集合 i 使用了 $(Y_i, R_i^2) \in Gr(n, k) \times \mathbb{S}_d^+$ 的表示，使用该表示并运用公式4-21定义的 $\delta_{FRPSD}(\cdot, \cdot)$ 即可进行图像集合的分类问题了，但是这样的分类方式过于粗糙，没有包含判别性、large margin 等一些性质，不利于模型的推广（文献 [32]），所以参考 [8,15] 等工作以及工作 [64,65] 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 $\delta_{FRPSD}(\cdot, \cdot)$ （公式4-21）这里还有一些事实需要注意（为了阐述的方便这里将公式4-19和公式4-21糅合在一起后用不同的

颜色标出前后两部分) :

$$\begin{cases} g_{(U,R^2)}((\Delta_1, D_1), (\Delta_2, D_2)) \\ \quad = \boxed{\text{tr}(\Delta_1 \Delta_2)} + \lambda \boxed{\text{tr}(R^{-1} D_1 R^{-2} D_2 R^{-1})}, \lambda > 0 \\ \delta^2(A, B) = \boxed{\|\Theta\|_F^2} + \lambda \boxed{\|\log(R_A^{-1} R_B^2 R_A^{-1})\|_F^2} \end{cases} \quad (4-22)$$

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; 对于公式4-22, 注意到其前半部分实际上是 Grassmann 流形的测地距离, 而它与4.1部分介绍的投影度量 (Projection Metric, 公式4-13) 之间仅相差一个倍数关系 [32], 所以公式4-22的前半部分可以由投影度量来代替, 对于公式4-22的后半部分, 注意到这是 SPD 矩阵流形的 AIM[37] 距离, 进一步的注意到 R_A, R_B 都是对角矩阵, 于是可以得到: $\|\log(R_A^{-1} R_B^2 R_A^{-1})\| = 2\|\log(R_A) - \log(R_B)\|$ 。综合以上两点特点 $\delta^2(A, B)$ 可以进一步的形式化为:

$$\begin{aligned} \delta_{FRPSD}^2(A, B) &= \|Y_A Y_A^T - Y_B Y_B^T\|_F^2 + \lambda \|\log(R_A) - \log(R_B)\|_F^2, \lambda > 0 \\ &= 2k - 2\|Y_A^T Y_B\|_F^2 + \lambda \|\log(R_A) - \log(R_B)\|_F^2 \end{aligned} \quad (4-23)$$

其中 Y_A, Y_B 的定义在4.3.1部分已经给出, 公式4-23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 [32], 因此容易从4-23出发构造正定的核, 表格4.1列出了文章 [32] 中使用的核。

表 4.1 固定秩对称半正定矩阵流形中的核

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	$k_R(A, B) = \exp(-\beta (\lambda \ \log(R_A) - \log(R_B)\ _F^2 - 2\ Y_A^T Y_B\ _F^2))$

最后工作 [32] 中还对比了经过核判别学习 (Kernel Discriminant Analysis, KDA) 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 [32] 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 低秩对称半正定矩阵判别学习方法

4.3.2小节结合着我们前期的一些尝试介绍了 Fixed-Rank PSD 矩阵流形用于图像集合分类的工作 [32]; 但是研究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 [60] 中对 $\delta_{FRPSD}(\cdot, \cdot)$ (公式4-21) 的形式化过程进行了详细的推导, 但是 $\delta_{FRPSD}(\cdot, \cdot)$ 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 $\delta_{FRPSD}(\cdot, \cdot)$ 仅仅借助一个平衡因子 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给出了我们的改进方案, 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 [63] 中, 为了在度量学习 (Metric Learning, ML) 的学习过程中保证马氏距离中的度量矩阵 M 是固定秩对称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二阶黎曼 Retraction 算子^① (Second Order Riemannian Retraction Operator)。文中作者构造性的使用 $Z_i = W_i W_i^T, W_i = ((C_i^{1/2} + Z) Y_i), Y_i = U_i(:, 1:k)$ 构造 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 [63] 通过要求 Z 满足 Fixed-Rank PSD 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 [63]。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, 如样本的 label 信息。

另一方面, 前面已经指出 $\delta_{FRPSD}(\cdot, \cdot)$ (公式4-21) 虽然保持了很好 geometry 相关的性质, 但是其分离的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个对称半正定矩阵之间的相似度/距离。为此, 如同文章 [60] 一样, 考虑到对称半正定矩阵与对称正定矩阵之间的关联性, 这里先来回顾一下对称正定矩阵流形中的不同距离度量, 并期望从中获得解决方案。

表 4.2 对称正定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 距离 [37]	$\ \log(X_i^{-\frac{1}{2}} X_j X_i^{-\frac{1}{2}})\ _F$	✓	✗
Log-Euclidean 距离 [38]	$\ \log(X_i) - \log(X_j)\ _F$	✓	✗
Stein 散度 [58]	$\log \det(\frac{X_1+X_2}{2}) - \frac{1}{2} \log \det(X_1 X_2)$	✗	✗
Jeffreys 散度 [66]	$\frac{1}{2} \text{tr}(X_1^{-1} X_2 + X_2^{-1} X_1) - d$	✗	✗
Cholesky 距离 [67]	$\ chol(X_1) - chol(X_2)\ _F$	✗	✓
Power-Euclidean 距离 [67]	$\frac{1}{\alpha} \ X_1^\alpha - X_2^\alpha\ _F$	✗	✓

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

表格4.2中列出的所有距离度量均可用于对称正定矩阵的之间距离的度量, 其中 Affine-Invariant 距离和 Log-Euclidean 距离分别在 [37] 和 [38] 中被提出, 且它们由各自的黎曼度量导出, 都是 \mathbb{S}_d^+ 上的测地距离, 并且 Log-Euclidean 距离以其优于 Affine-Invariant 距离的计算性质而赢得了不少青睐, 但是遗憾的是这两种距离都不能用于不满秩 (也就是半正定的) 的情况, Stein 散度 [58] 和 Jeffreys 散度 [66] 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 $\log \det(\cdot)$ 而 Jeffreys 散度需要计算 $X_i^{-1}, i = 1, 2$ 所以也不能直接用于处理半正定的输入, 最后剩下 Cholesky 距离 [67] 和 Power-Euclidean

① 关于 Retraction 算子读者可以参考本文的第二章

距离 [67] 这两种距离由于不涉及求逆或者 $\log(\cdot)$ 操作所以可以直接用于半正定矩阵，而与 $\delta_{FRPSD}(\cdot, \cdot)$ (公式4-21) 相比，其直接考虑半正定输入，没有割裂 $\text{Gr}(d, k), \mathbb{S}_k^+$ 之间的关系，应该有更好的表示能力；但需要注意到的是 Cholesky 距离和 Power-Euclidean 距离并不是 \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 距离作为半正定矩阵的距离度量。特别地，简单的验证试验发现 Power-Euclidean 距离比 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 距离进行介绍和实验。

最后还需要注意的是文章 [32] 使用 Fixed-Rank PSD 建模图像集合的方法中，Fixed-Rank 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-Euclidean 距离度量两个 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。

以上是对本文所提方法的一个概要介绍，下面将进一步细化该方法。遵循图像集合分类问题的两个主要步骤将接下来的内容大致分为：1) 如何构造一个好的 Low-Rank PSD 的表示；2) 如何利用 Power-Euclidean 距离进行判别学习以及 Low-Rank PSD 矩阵上的判别学习方法的形式化。

4.4.1 融入判别信息的低秩对称半正定矩阵的构造

在4.4节的前半部分提出了借鉴 [63] 中构造 PSD 的内容 (公式4-24)，构造图像集合的带判别性的低秩对称半正定矩阵表示：

$$Z_i = W_i W_i^T, W_i = \left((C_i^{1/2} + Z) Y_i \right), Y_i = U_i(:, 1:k), C_i = U_i \Lambda_i U_i^T, R_i^2 = \Lambda \quad (4-24)$$

在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 [32] 中 Fixed-Rank PSD 矩阵的构造方式：

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

此外，当 $Z = I - C_i^{\frac{1}{2}}$ 时：

$$Z_i = Y_i Y_i^T \quad (4-26)$$

正好是 [8] 中的投影矩阵的结果，由此可以一定程度上说明4-24构造的合理性。

说明完合理性之后，接下是如何选择公式4-24中的 Z 的问题，虽然文章[63]中给出了一种构造的方式，但是由于目的不同（文章[63]中是为了优化马氏距离中的度量矩阵，这里是为了表示图像集合）所以这里选择另一种矩阵 Z 的构造方式：借助判别学习(Discriminate Learning)的框架学习矩阵 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：

$$G_{ij} = \begin{cases} 1, & \text{if } y_i = y_j \\ -1, & \text{else} \end{cases} \quad (4-27)$$

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Graph Embedding[68]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 ：

$$G_w = \{G_{ij}^w\}_{n \times n}, \text{ where } G_{ij}^w = \begin{cases} 1, & j \text{ 属于 } i \text{ 的前 } k_w \text{ 个同类近邻} \\ 0, & \text{否者} \end{cases} \quad (4-28)$$

$$G_b = \{G_{ij}^b\}_{n \times n}, \text{ where } G_{ij}^b = \begin{cases} 1, & j \text{ 属于 } i \text{ 的前 } k_b \text{ 个不同类近邻} \\ 0, & \text{否则} \end{cases}$$

最后，利用 G_w, G_b 对公式4-27中的 G 进行重新定义：

$$G = G_w - G_b \quad (4-29)$$

其意义可用图4.3表示。此外，为了平衡正负样本对的比例，实验中还将 G 中的1, -1分

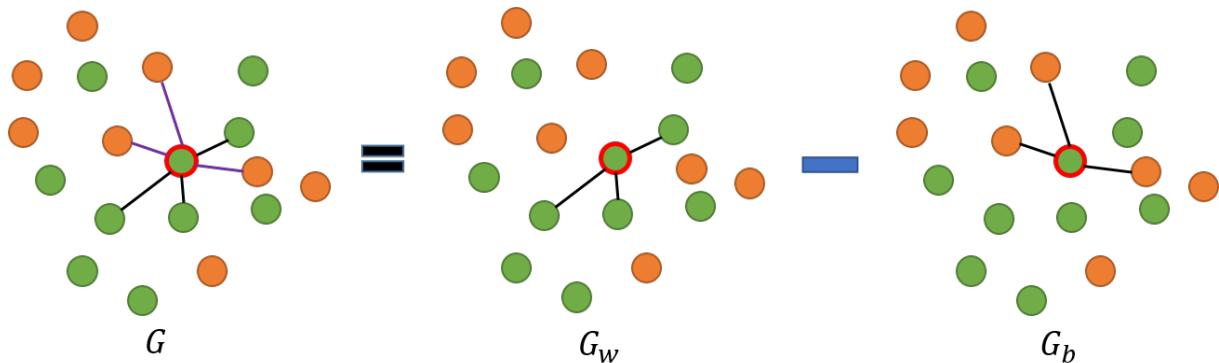


图 4.3 图嵌入框架示意图

别除以正负样本对的个数。

接下来，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4-30中的损失函数：

$$C(Z) = - \sum_{i=1}^n \sum_{j=1}^n G_{ij} \rho_{ij}^2 + \gamma \|Z\|_F^2 = -\text{tr}(GF^T) + \gamma \|Z\|_F^2, \text{ where } F_{ij} = \rho_{ij}^2 \quad (4-30)$$

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，这虽然只是一个小的 trick，但是实验中发现该 trick 却比较有效。

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，这里把第二章的两条在计算矩阵方向导数的时候的规律（rule2-21）再表述一遍。

$$\begin{cases} \text{rule 1 : } D_X(f \circ g)(X)[H] = D_{g(X)}f(g(X))[D_Xg(X)[H]] \\ \text{rule 2 : } D_X \langle f(X), g(X) \rangle [H] = \langle D_Xf(X)[H], g(X) \rangle + \langle f(X), D_Xg(X)[H] \rangle \end{cases}$$

关于矩阵函数的方向导数的具体内容，读者可以参看本文第二章的内容，此外读者也可以在 [33] 中找到关于 rule 1, rule 2 的内容。

对于最小化问题4-30，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 $\nabla_Z C(Z)$ ，为了方便起见定义 $k_{ij} = \langle Z_i, Z_j \rangle_F$ ，接下来利用公式4-31对 $C(Z)$ 计算其关于 Z 的导数。

$$\begin{aligned} C(Z) &= \text{tr}(GF^T) + \gamma \|Z\|_F^2 = \sum_{i=1}^n \sum_{j=1}^n G_{ij} F_{ij} + \gamma \|Z\|_F^2 = \sum_{i=1}^n \sum_{j=1}^n G_{ij} \frac{k_{ij}^2}{k_{ii} k_{jj}} + \gamma \|Z\|_F^2 \\ \frac{\partial}{\partial Z} C(Z) &= \sum_{i=1}^n \sum_{j=1}^n G_{ij} \left(c_1 \frac{\partial}{\partial Z} k_{ij} - c_2 \frac{\partial}{\partial Z} k_{ii} - c_3 \frac{\partial}{\partial Z} k_{jj} \right) + 2\gamma Z \quad (4-31) \\ \text{where } c_1 &= \frac{2k_{ij}k_{ii}k_{jj}}{(k_{ii}k_{jj})^2}, c_2 = \frac{k_{ij}k_{ij}k_{jj}}{(k_{ii}k_{jj})^2}, c_3 = \frac{k_{ij}k_{ij}k_{ii}}{(k_{ii}k_{jj})^2} \end{aligned}$$

公式4-31中 $\frac{\partial}{\partial Z} k_{ij}$ 的计算与第二章中第一个例子2-33的计算类似，具体过程如下：

$$\begin{aligned} D_Z k_{ij}[H] &= D_Z \langle Z_i, Z_j \rangle_F \\ &= \langle D_Z Z_i[H], Z_j \rangle_F + \langle Z_i, D_Z Z_j[H] \rangle_F \quad (4-32) \end{aligned}$$

由于 $\langle D_Z Z_i[H], Z_j \rangle_F$ 与 $\langle Z_i, D_Z Z_j[H] \rangle_F$ 的计算是类似的，所以仅以前一部分作为研究对象，

^① 注：这里不使用 $\frac{\langle Z_i^{\frac{1}{m}}, Z_j^{\frac{1}{m}} \rangle_F}{\|Z_i\|_F^{\frac{1}{m}} \|Z_j\|_F^{\frac{1}{m}}}$; $i, j = 1, 2, \dots, n$ 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

其结果可以很好的平移到另一部分。

$$\begin{aligned}
 \langle D_Z Z_i [H], Z_j \rangle_F &= \left\langle D_Z \left((C_i^{\frac{1}{2}} + Z) S_i (C_i^{\frac{1}{2}} + Z)^T \right) [H], (C_j^{\frac{1}{2}} + Z) S_j (C_j^{\frac{1}{2}} + Z)^T \right\rangle_F \\
 D_Z \left((C_i^{\frac{1}{2}} + Z) S_i (C_i^{\frac{1}{2}} + Z)^T \right) [H] &= D_Z (C_i^{\frac{1}{2}} S_i C_i^{\frac{T}{2}} + C_i^{\frac{1}{2}} S_i Z^T + Z S_i C_i^{\frac{T}{2}} + Z S_i Z^T) [H] \\
 &= C_i^{\frac{1}{2}} S_i H^T + H S_i C_i^{\frac{T}{2}} + H S_i Z^T + Z S_i H^T \\
 &= (C_i^{\frac{1}{2}} + Z) S_i H^T + H S_i (C_i^{\frac{T}{2}} + Z^T)
 \end{aligned} \tag{4-33}$$

其中 $S_i = Y_i Y_i^T, i = 1, 2, \dots, n$, 接下来利用公式4-33的结果, 得到公式4-34的结果 (其中利用了 $Z_i, S_i; i = 1, 2, \dots, n$ 是对称矩阵的性质)。

$$\left\{
 \begin{aligned}
 \langle D_Z Z_i [H], Z_j \rangle_F &= \left\langle (C_i^{\frac{1}{2}} + Z) S_i H^T + H S_i (C_i^{\frac{T}{2}} + Z^T), Z_j \right\rangle_F \\
 &= \text{tr} \left((C_i^{\frac{1}{2}} + Z) S_i H^T Z_j \right) + \text{tr} \left(H S_i (C_i^{\frac{T}{2}} + Z^T) Z_j \right) \\
 &= \text{tr} \left(H^T Z_j (C_i^{\frac{1}{2}} + Z) S_i \right) + \text{tr} \left(H S_i (C_i^{\frac{T}{2}} + Z^T) Z_j \right) \\
 &= 2 \text{tr} \left(H^T Z_j (C_i^{\frac{1}{2}} + Z) S_i \right) = 2 \left\langle H, Z_j (C_i^{\frac{1}{2}} + Z) S_i \right\rangle_F \tag{4-34} \\
 D_Z k_{ij} [H] &= 2 \left\langle H, Z_j (C_i^{\frac{1}{2}} + Z) S_i \right\rangle_F + 2 \left\langle H, Z_i (C_j^{\frac{1}{2}} + Z) S_j \right\rangle_F \\
 \frac{\partial}{\partial Z} k_{ij} &= 2 Z_j (C_i^{\frac{1}{2}} + Z) S_i + 2 Z_i (C_j^{\frac{1}{2}} + Z) S_j \\
 &= 2 (Z_j W_i Y_i^T + Z_i W_j Y_j^T)
 \end{aligned}
 \right.$$

最后结合公式4-31和4-34的结果, 即可计算出 $C(Z)$ 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。这里使用图4.4示意带判别信息的低秩对称半正定矩阵编码图像集合的模型。

4.4.2 低秩对称半正定矩阵集合上的判别学习方法

在4.4.1小节中介绍了将判别信息编码到 Low-Rank PSD 矩阵问题, 但是仅仅有内嵌入判别的编码还是不足以处理太复杂的问题, 且算法也不具有 Large Margin 等性质, 类似于[32]中的方法, 这里选择在核判别学习 (Kernel Discriminant Learning, KDA[27]) 的框架下进行判别学习。特别地, 注意到 Power-Euclidean 距离的定义可以由 $k_{ij} = \left\langle Z_i^{\frac{1}{m}}, Z_j^{\frac{1}{m}} \right\rangle_F$ 的 kernel 的形式导出, 并且容易证明 $K = \{k_{ij}\}_{n \times n}$ 的正定性。

由于 KDA[27] 的研究已经非常成熟, 且这里也不是对它的改进或相关的工作, 所以不会从头再介绍一遍, 而只会简单的回顾一下 KDA 的基本思想: 首先, KDA 的基础是线性判别分析 (Linear Discriminant Analysis, LDA), 其目标是使得内类散度小而类间散度大。KDA 则是 LDA 在再生核希尔伯特空间 (Reproducing Kernel Hilbert space, RKHS) 中的版本, 其目的与 LDA 一样。文献 [27] 中给出了十分简洁的 KDA 的形式: 设 $n_c, c = 1, 2, \dots, C$ 表示各个类别中的样本数, $\sum_{c=1}^C n_c = n$, 这里的 n 表示的是样本总

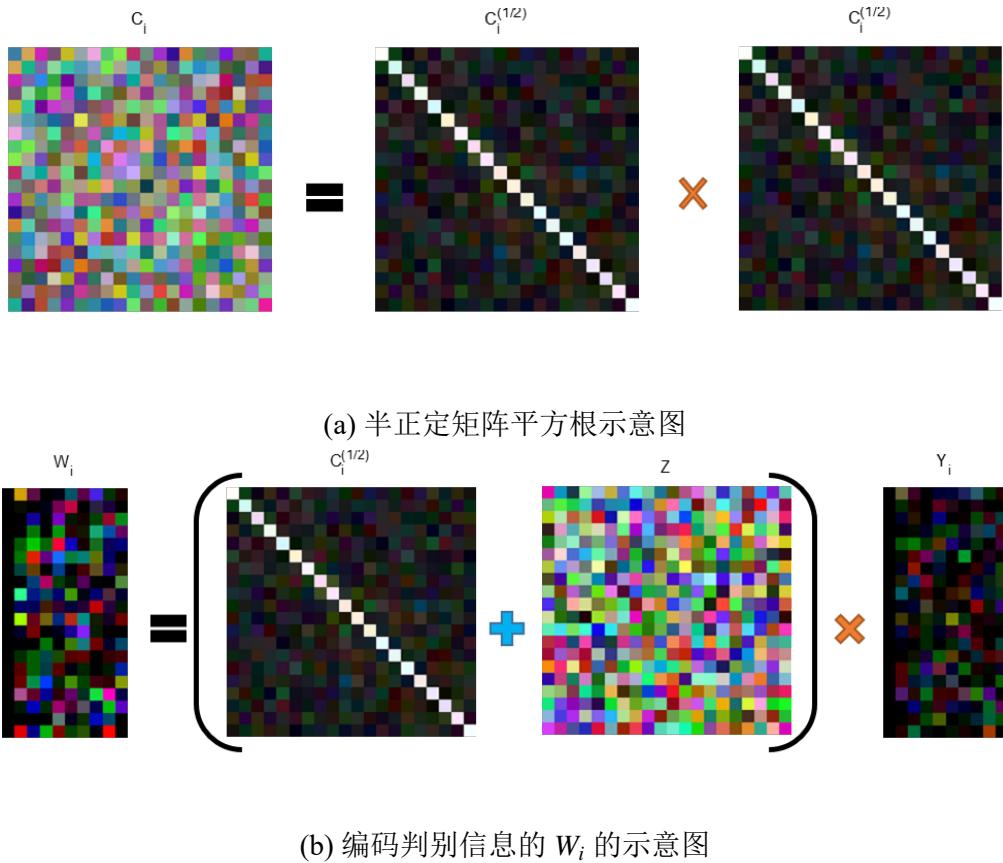


图 4.4 带判别信息的低秩对称半正定矩阵模型示意图

数, $\phi(\cdot)$ 表示的是非线性变化 (例如接下来将要使用的 $\phi(Z_i) = Z_i^{\frac{1}{n}}, n > 1$), 定义核矩阵 $K = \{k_{ij}\}_{n \times n}$, 其中 $k_{ij} = \langle \phi(Z_i), \phi(Z_j) \rangle_F$, 则 KDA 的目标形式化为公式4-35。

$$\alpha_{opt} = \arg \max \frac{\alpha^T K W K \alpha}{\alpha^T K K \alpha} \quad (4-35)$$

其中 W 的定义如公式4-36所示。

$$W = \{W_{ij}\}_{n \times n}, \text{ where } W_{ij} = \begin{cases} \frac{1}{n_c}, & \text{若 } Z_i, Z_j \text{ 同时属于第 } c \text{ 类} \\ 0, & \text{否则} \end{cases} \quad (4-36)$$

以上便是本章所提方法使用的判别学习方法——KDA[27] 的介绍, 试验中我们使用了文献 [69,70] 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[2], 材质分类数据库 UIUC[29] 以及视频人脸识别数据库 YTC[1]), 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, 这里不再赘述; 在获得原始特征之后, 利用公式3-30构造初始半正定矩阵表示 C_i (此时不需要再加正则项), 然后利用4.4.1的编码过程获得图像集合带判别性的低秩对称半正定矩阵表示。关于数据库的详细介绍以及对应的测试协议可以参看1.3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给出了我们的实验结果。

表 4.3 低秩对称半正定矩阵判别学习方法对比实验结果

数据集 方法	ETH80	UIUC	YTC
GDA[8]	92.50±3.54	53.33±2.32	66.73±3.16
CDL-PLS[15]	93.25±4.72	53.89±4.06	70.28±2.13
RSR-Stein[57]	93.25±3.34	52.41±4.03	72.77±2.69
SPDML-Stein[17]	90.50±3.87	49.17±2.37	61.57±3.43
SPDML-AIM[17]	90.75±3.34	48.09±1.82	64.66±2.92
LEML[56]	94.75±2.49	48.98±3.69	70.53±2.95
FRPSD – KDA _{Linear} [32]	94.50±3.07	52.04±3.80	70.93±3.28
FRPSD – KDA _{Polynomial} [32]	96.00±2.42	56.02±3.93	70.74±3.05
FRPSD – KDA _{Laplace} [32]	95.50±2.84	57.69±3.35	70.14±3.04
FRPSD – KDA _{RBF} [32]	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 [32] 中方法的简称 (是 Fixed-Rank PSD 和 KDA 的缩写), 其下标表示了核的类型: 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 [32] 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。还有 GDA[8] 的结果 (该结果也是自己实现并小心调参后获得的结果)。其它方法我们均从作者主页获取代码并小心调参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建模的方法 (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的主要原因有两个方面: 1) 通过与文章 [32] 中的方法 (表中的 FRPSD – KDA 一类方法) 相比, 可得出本文使用的 Power-Euclidean 距离更能刻画 PSD

矩阵的性质的结论，这与我们一开始的想法是吻合的；2) 表中的最后六行的结果与其它行的结果相比说明了 PSD 建模图像集合的有效性。最后还注意到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。

实验中还发现：公式4-30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，如 ETH[2] 和 YTC[1]（其中 YTC 属于集合个数多，但是每个集合都不是很大的类型，这在视频监控中很常见）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[29] 上我们设置 $\gamma = 0$ ），其它实验参数的设置主要包含 k_w, k_b 的设置，Power-Euclidean 距离中 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-Euclidean 距离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-Euclidean 距离相关的核

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	$k_R(A, B) = \exp\left(-\beta \ Z_i^{\frac{1}{m}} - Z_j^{\frac{T}{m}}\ _F^2\right)$

但是从实验结果中可以看出：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-Euclidean 距离有更好的促进作用），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，并且结合前期关于 Fixed-Rank PSD 流形的研究以及工作 [32] 的内容，针对使用子空间、协方差矩阵以及 Fixed-Rank PSD 矩阵建模图像集合的工作 [32] 中存在的问题，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对称半正定矩阵表示图像集合；2) 针对文献 [32] 中使用的 $\delta_{FRPSD}(\cdot, \cdot)$ （公式4-21）割裂了 Grassmann 流形和对称正定矩阵流形之间关系的问题提出了使用 Power-Euclidean 距离进行判别学习的方法；3) 在 KDA[27] 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，前面已经提到表格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使用更多 Kernel 版本的方法，可能也是一个尝试的方向。

第五章 结束语

计算视觉问题的研究经过几十年的发展，取得了巨大的成就。在计算机视觉中，集合数据的研究经历十多年时间也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这样一些模型进行研究。而本文也正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集合数据的表示和判别学习展开。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，样本稀少导致的样本协方差不满秩等问题以及子空间建模没有利用尺度信息（特征值）的问题，并结合着最新的利用固定秩对称半正定矩阵建模图像集合的方法，提出了使用低秩对称半正定矩阵建模图像集合的方法，并进行了实验验证。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 [41,44] 研究了黎曼流形中的偏最小二乘问题。该问题的研究中首先参考了 [41,44] 的工作将欧氏空间中的投影的概念泛化到了黎曼流形，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) 图像研究问题

的不同（主要是前者的数据分布更稀疏），本章提出了多切空间偏最小二乘回归的方法，在流形的多个切空间中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[29] 数据集上使用的 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角度出发进行研究，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，协方差矩阵表示维度过高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 [32] 中使用固定秩的对称半正定矩阵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距离 $\delta_{FRPSD}(\cdot, \cdot)$ （公式 4-21）割裂了 Grassmann 流形和 SPD 矩阵流形之间的关系的问题提出使用编码了判别信息的低秩对称半正定矩阵建模图像集合的方法；最后的实验中我们得到了与 state-of-the-art 可比甚至是稍好一些的结果。

本文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，成功的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读者能够引以为戒。

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。通过思考目前存在的一些不足与困扰，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存在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，从而获得了多切空间偏最小二乘回归算法。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，主要是逐步回归方案的抗过拟合能力不足导致了以上两个参数过大时出现过拟合而过小时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。工作 [52] 给出了一个很好的启示：使用 Adaboost 的框架来融合多模型，这样可以有效的控制训练和测试的误差。

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 的框架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度量与 KDA 中度量不一致也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-Euclidean 度量。

最后，注意到 Deep Learning 已经在各个领域都取得巨大的进展，而在图像集合分类问题中也有一些有益的尝试（如：[23] 和 [22]），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，这是一个很有应用前景的方向，对检索，追逃等领域有重大意义。

参考文献

- [1] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley, “Face tracking and recognition with visual constraints in real-world videos,” in *Computer Vision and Pattern Recognition*, pp. 1–8. 2008.
- [2] B. Leibe and B. Schiele, “Analyzing appearance and contour based methods for object categorization,” in *Computer Vision and Pattern Recognition*, vol. 2, pp. II–409. 2003.
- [3] R. Gross and J. Shi, “The cmu motion of body (mobo) database,” Tech. Rep., Robotics Institute, Carnegie Mellon University, 2001.
- [4] Z. Huang, R. Wang, S. Shan, and X. Chen, “Learning euclidean-to-riemannian metric for point-to-set classification,” in *Computer Vision and Pattern Recognition*, pp. 1677–1684. 2014.
- [5] Y. Li, R. Wang, Z. Huang, S. Shan, and X. Chen, “Face video retrieval with image query via hashing across euclidean space and riemannian manifold,” in *Computer Vision and Pattern Recognition*, pp. 4758–4767. 2015.
- [6] P. Zhu, L. Zhang, W. Zuo, and D. Zhang, “From point to set: Extend the learning of distance metrics,” in *International Conference on Computer Vision*, pp. 2664–2671. 2013.
- [7] O. Yamaguchi, K. Fukui, and K. Maeda, “Face recognition using temporal image sequence,” in *Automatic Face and Gesture Recognition*, pp. 318–323. 1998.
- [8] J. Hamm and D.D. Lee, “Grassmann discriminant analysis: a unifying view on subspace-based learning,” in *International Conference on Machine learning*, pp. 376–383. 2008.
- [9] R. Wang, S. Shan, X. Chen, and W. Gao, “Manifold-manifold distance with application to face recognition based on image set,” in *Computer Vision and Pattern Recognition*, pp. 2940–2947. 2008.
- [10] R. Wang and X. Chen, “Manifold discriminant analysis,” in *Computer Vision and Pattern Recognition*, pp. 429–436. 2009.
- [11] H. Cevikalp and B. Triggs, “Face recognition based on image sets,” in *Computer Vision and Pattern Recognition*, pp. 2567–2573. 2010.
- [12] Y. Hu, A.S. Mian, and R. Owens, “Sparse approximated nearest points for image set classification,” in *Computer vision and pattern recognition*, pp. 121–128. 2011.
- [13] M. Yang, P. Zhu, L. Van Gool, and L. Zhang, “Face recognition based on regularized nearest points between image sets,” in *Automatic Face and Gesture Recognition*, pp. 1–7. 2013.
- [14] W. Wang, R. Wang, S. Shan, and X. Chen, “Probabilistic nearest neighbor search for robust classification of face image sets,” in *Automatic Face and Gesture Recognition*, pp. 1–7. 2015.
- [15] R. Wang, H. Guo, L.S. Davis, and Q. Dai, “Covariance discriminative learning: A natural and efficient approach to image set classification,” in *Computer Vision and Pattern Recognition*, pp. 2496–2503. 2012.
- [16] R. Vemulapalli, J. Pillai, and R. Chellappa, “Kernel learning for extrinsic classification of manifold features,” in *Computer Vision and Pattern Recognition*, pp. 1782–1789. 2013.
- [17] M.T. Harandi, M. Salzmann, and R. Hartley, “From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices,” in *European Conference On Computer Vision*, pp. 17–32. Springer, 2014.

- [18] J. Lu, G. Wang, and P. Moulin, “Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning,” in *International Conference on Computer Vision*, pp. 329–336. 2013.
- [19] Z. Huang, R. Wang, S. Shan, and X. Chen, “Hybrid euclidean-and-riemannian metric learning for image set classification,” in *Asian Conference On Computer Vision*, pp. 562–577. Springer, 2015.
- [20] W. Wang, R. Wang, Z. Huang, S. Shan, and X. Chen, “Discriminant analysis on riemannian manifold of gaussian distributions for face recognition with image sets,” in *Computer Vision and Pattern Recognition*, pp. 2048–2057. 2015.
- [21] M. Harandi, M. Salzmann, and M. Baktashmotagh, “Beyond gauss: Image-set matching on the riemannian manifold of pdfs,” in *International Conference on Computer Vision*. 2015.
- [22] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou, “Multi-manifold deep metric learning for image set classification,” in *Conference on Computer Vision and Pattern Recognition*, pp. 1137–1145. 2015.
- [23] M. Hayat, M. Bennamoun, and S. An, “Deep reconstruction models for image set classification,” *Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 713–727, 2015.
- [24] P. Zhu, W. Zuo, L. Zhang, S.C.-K. Shiu, and D. Zhang, “Image set-based collaborative representation for face recognition,” *Information Forensics and Security*, vol. 9, no. 7, pp. 1120–1132, 2014.
- [25] Y.-C. Chen, V.M. Patel, P.J. Phillips, and R. Chellappa, “Dictionary-based face recognition from video,” in *European Conference On Computer Vision*, pp. 766–779. Springer, 2012.
- [26] R. Rosipal and L.J. Trejo, “Kernel partial least squares regression in reproducing kernel hilbert space,” *The Journal of Machine Learning Research*, vol. 2, pp. 97–123, 2002.
- [27] G. Baudat and F. Anouar, “Generalized discriminant analysis using a kernel approach,” *Neural computation*, vol. 12, no. 10, pp. 2385–2404, 2000.
- [28] L. Wolf, T. Hassner, and I. Maoz, “Face recognition in unconstrained videos with matched background similarity,” in *Computer Vision and Pattern Recognition*. 2011.
- [29] Z. Liao, J. Rock, Y. Wang, and D. Forsyth, “Non-parametric filtering for geometric detail extraction and material representation,” in *Computer Vision and Pattern Recognition*. 2013.
- [30] G.B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.
- [31] O. Tuzel, F. Porikli, and P. Meer, “Region covariance: A fast descriptor for detection and classification,” in *European Conference On Computer Vision*, pp. 589–600. Springer, 2006.
- [32] M. Faraki, M. Harandi, and F. Porikli, “Image set classification by symmetric positive semi-definite matrices,” in *Winter Conference on Applications of Computer Vision*. 2016.
- [33] N. Boumal, P.-A. Absil, et al., “Discrete curve fitting on manifolds,” in *30th Benelux Meeting on Systems and Control*. 2011.
- [34] P.A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2008.
- [35] 梅加强, 流形与几何初步, 科学出版社, 2013.
- [36] 黄智武, “黎曼度量学习及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.
- [37] X. Pennec, P. Fillard, and N. Ayache, “A riemannian framework for tensor computing,” *International Journal of Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006.

- [38] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, “Geometric means in a novel vector space structure on symmetric positive-definite matrices,” *SIAM journal on matrix analysis and applications*, vol. 29, no. 1, pp. 328–347, 2007.
- [39] R. Fletcher and C.M. Reeves, “Function minimization by conjugate gradients,” *The computer journal*, vol. 7, no. 2, pp. 149–154, 1964.
- [40] K.B. Petersen, M.S. Pedersen, et al., “The matrix cookbook,” vol. 7, pp. 15, 2008.
- [41] H.J. Kim, N. Adluru, B.B. Bendlin, S.C. Johnson, B.C. Vemuri, and V. Singh, “Canonical correlation analysis on riemannian manifolds and its applications,” in *European Conference On Computer Vision*, pp. 251–267. Springer, 2014.
- [42] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, “Manopt, a matlab toolbox for optimization on manifolds,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1455–1459, 2014.
- [43] S. Amari and H. Nagaoka, *Methods of information geometry*, vol. 191, American Mathematical Soc., 2007.
- [44] P.T. Fletcher, C. Lu, S.M. Pizer, and S. Joshi, “Principal geodesic analysis for the study of nonlinear statistics of shape,” *IEEE Transactions on Medical Imaging*, vol. 23, no. 8, pp. 995–1005, 2004.
- [45] H. Wold, “Path models with latent variables: The nipals approach,” in *International perspectives on mathematical and statistical model building*, pp. 307–357. Academic Press, 1975.
- [46] R. Rosipal and L.J. Trejo, “Kernel partial least squares regression in reproducing kernel hilbert space,” *The Journal of Machine Learning Research*, vol. 2, pp. 97–123, 2002.
- [47] M. Barker and W. Rayens, “Partial least squares for discrimination,” *Journal of chemometrics*, vol. 17, no. 3, pp. 166–173, 2003.
- [48] A. Höskuldsson, “Pls regression methods,” *Journal of chemometrics*, , no. 2, pp. 211–228, 1988.
- [49] S. Wold, M. Sjöström, and L. Eriksson, “Pls-regression: a basic tool of chemometrics,” *Chemometrics and intelligent laboratory systems*, vol. 58, no. 2, pp. 109–130, 2001.
- [50] R. Rosipal and N. Krämer, “Overview and recent advances in partial least squares,” in *Subspace, latent structure and feature selection*, pp. 34–51. Springer, 2006.
- [51] H.D. Vinod, “Canonical ridge and econometrics of joint production,” *Journal of econometrics*, vol. 4, no. 2, pp. 147–166, 1976.
- [52] O. Tuzel, F. Porikli, and P. Meer, “Pedestrian detection via classification on riemannian manifolds,” *Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1713–1727, 2008.
- [53] W.W. Hager and H. Zhang, “A survey of nonlinear conjugate gradient methods,” *Pacific Journal of Optimization*, vol. 2, no. 1, pp. 35–58, 2006.
- [54] F. Yger and M. Sugiyama, “Supervised logeuclidean metric learning for symmetric positive definite matrices,” *arXiv preprint arXiv:1502.03505*, 2015.
- [55] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [56] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen, “Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification,” in *International Conference on Machine Learning*, pp. 720–729. 2015.
- [57] M.T. Harandi, C. Sanderson, R. Hartley, and B.C. Lovell, “Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach,” in *European Conference On Computer Vision*, pp. 216–229. Springer, 2012.

- [58] S. Sra, “A new metric on the manifold of kernel matrices with application to matrix geometric means,” in *Advances in Neural Information Processing Systems*, pp. 144–152. 2012.
- [59] D. Karrasch, “An introduction to grassmann manifolds and their matrix representation,” 2014.
- [60] S. Bonnabel and R. Sepulchre, “Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank,” *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1055–1070, 2009.
- [61] G. Meyer, S. Bonnabel, and R. Sepulchre, “Regression on fixed-rank positive semidefinite matrices: a riemannian approach,” *The Journal of Machine Learning Research*, vol. 12, pp. 593–625, 2011.
- [62] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre, “Low-rank optimization on the cone of positive semidefinite matrices,” *SIAM Journal on Optimization*, vol. 20, no. 5, pp. 2327–2351, 2010.
- [63] Y. Mu, “Fixed-rank supervised metric learning on riemannian manifold,” in *AAAI Conference on Artificial Intelligence*. 2016.
- [64] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, “Kernel methods on riemannian manifolds with gaussian rbf kernels,” *Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2464–2477, 2015.
- [65] M. Harandi, R. Hartley, C. Shen, B. Lovell, and C. Sanderson, “Extrinsic methods for coding and dictionary learning on grassmann manifolds,” *International Journal of Computer Vision*, vol. 114, no. 2-3, pp. 113–136, 2015.
- [66] M. Harandi, M. Salzmann, and F. Porikli, “Bregman divergences for infinite dimensional covariance matrices,” in *Computer Vision and Pattern Recognition*, pp. 1003–1010. 2014.
- [67] I.L. Dryden, A. Koloydenko, and D. Zhou, “Non-euclidean statistics for covariance matrices, with applications to diffusion tensor imaging,” *The Annals of Applied Statistics*, pp. 1102–1123, 2009.
- [68] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, “Graph embedding and extensions: a general framework for dimensionality reduction,” *Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.
- [69] D. Cai, X. He, and J. Han, “Speed up kernel discriminant analysis,” *The International Journal on Very Large Data Bases*, vol. 20, no. 1, pp. 21–33, 2011.
- [70] D. Cai, *Spectral regression: A regression framework for efficient regularized subspace learning*, ProQuest, 2009.

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研究员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活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。

诚挚感谢山世光研究员的包容与指导；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研究员的悉心指导和帮助，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的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，让人愿意与他一起相处共事。

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，柴秀娟老师，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晓彪老师，感谢胡兰平老师，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，冯钢老师，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。感谢王骐师兄，阚美娜师姐在 Intel 的凝视矫正项目中的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区别的不同。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，我也将竭尽所能的报答你们。

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 – 2016.7，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

【攻读硕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”

