魯東大學

本 科 毕 业 设 计

基于流形学习子空间的人脸识别算法

姓　　名	姜希成
学　　院	信息与电气工程学院
专　　业	计算机科学与技术
年　　级	2015
学　　号	20152203031
指导教师	王庆军

2019 年 5 月 4 日

独 创 声 明

本人郑重声明：所呈交的毕业设计，是本人在指导老师的指导下，独立进行研究工作所取得的成果，成果不存在知识产权争议。尽我所知，除文中已经注明引用的内容外，本设计不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体均已在文中以明确方式标明。

此声明的法律后果由本人承担。

作者签名：_____

年　　月　　日

毕业设计使用授权声明

本人完全了解鲁东大学关于收集、保存、使用毕业设计的规定。

本人愿意按照学校要求提交设计的印刷本和电子版，同意学校保存设计的印刷本和电子版，或采用影印、数字化或其它复制手段保存设计；同意学校在不以营利为目的的前提下，建立目录检索与阅览服务系统，公布设计的部分或全部内容，允许他人依法合理使用。

（保密论文在解密后遵守此规定）

论文作者：_____

年　　月　　日

毕业设计开题报告

姓 名	姜希成	学 院	信息与电气工程学院	年级	2015	学号	20152203031
题 目	基于流形学习子空间的人脸识别算法						
课题来源	教师推荐	课题类别	应用实践研究				

选题意义（包括科学意义和应用前景，研究概况，水平和发展趋势，列出主要参考文献目录）：

近年来，人脸识别市场得到了突飞猛进的发展，早在 2017 年，其市场规模就已经超过 40.5 亿美元，预计 2022 年会超过 77.6 亿美元，因此人脸识别技术的市场前期非常好，这预示着人脸识别的爆点已经到来，而且经过长时间的积淀，人脸识别技术已经十分成熟，而流形学习是进行人脸识别的常用方法，流形学习自上世纪 60 年代首次被提出到现在已经发展了近 60 年，相继出现许多经典算法和经典理论，所以非常具有研究价值，本文将以人脸识别为应用背景，对几种流性学习方法进行研究。

主要参考文献：

[1] 王庆军，张汝波，刘冠群. 一种应用于人脸识别的核正交等度规映射算法[J]. 光电子激光，2010(11): 1702-1705.

[2] 左加阔. 基于流形学习算法的新生儿疼痛表情识别[D]. 南京邮电大学: 信号与信息处理，2011.

[3] 陶晓燕，姬红兵，景志宏. 一种用于人脸识别的正交邻域保护嵌入算法[N]. 西安电子科技大学学报（自然科学版），2008(6).

[4] 陈江峰. 线性图嵌入算法及其应用[D]. 北京交通大学: 信号与信息处理，2012.

研究主要内容和预期结果（说明具体研究内容和拟解决的关键问题，预期结果和形式，如在理论上解决哪些问题及其价值，或应用的可能性及效果）：

主要研究内容：本文将主要介绍邻域保护嵌入算法（NPE）和等距映射算法（IsoProjection），并在其基础上进一步优化这两种算法，即改进为正交邻域保护嵌入算法（ONPE）和正交等距映射算法（OIsoProjection）。

预期结果： 在 NPE 算法和 IsoProjection 算法的基础上，通过正交化的方法改进出 ONPE 算法和 OIsoProjection 算法，进一步提升这两种算法的识别率，并在 ORL 人脸库和 Yale 人脸库上验证这两种改进算法的有效性。

拟采取的研究方法和技术路线（包括理论分析、计算，实验方法和步骤及其可行性论证，可能遇到的问题和解决方法，以及研究的进度与计划）：

研究方法：实验法、经验总结法、文献研究法、功能分析法

技术路线： 通过查阅整理相关学习资料学习流形学习的相关算法如 MDS 算法，ISOMAP 算法，LLE 算法，NPE 算法和 IsoProjection 算法等，并使用 MATLAB 软件运行这些算法并分析算法原理，有了一定的基础后再在 NPE 算法和 IsoProjection 算法的基础上引入正交化的功能，从理论上分析了该正交化算法的可行性，并在 MATLAB 上实现 NPE 算法和 IsoProjection 算法，最后在 ORL 人脸库上验证这两种算法的有效性。

研究的进度与计划：2019 年 3 月 1 日，选题。

2019 年 3 月 1 日---2019 年 3 月 6 日，查阅并整理相关资料。

2019 年 3 月 6 日---2019 年 3 月 10 日，完成开题报告。

2019 年 3 月 10 日---2019 年 4 月 20 日，完成程序创建与前期功能实现。

2019 年 4 月 20 日---2019 年 5 月 1 日，完善程序功能，完成论文初稿。

指导教师意见（对论文选题的意义、应用性、可行性、进度与计划等内容进行评价，填写审核结果：同意开题、修改后再开题、不同意开题）：

针对 NPE 和 OIsoProjection 的优缺点，在原算法的基础上拟加入正交化的思想，显然会使得人脸识别率明显提高。本设计的进度与计划安排合理，研究内容具有一定的研究价值和现实指导意义，且方案切实可行。同意开题。

签名：

年　　月　　日

学院毕业设计领导小组意见：

同 意 开 题

（签章）

年　　月　　日

毕业设计结题报告

姓 名	姜希成	学 院	信息与电气工程学院	年级	2015	学号	20152203031
题 目	基于流形学习子空间的人脸识别算法						
课题来源	教师推荐	课题类别	应用实践研究				

本课题完成情况介绍（包括研究过程、实验过程、结果分析、存在的问题及应用情况等。）

研究过程：查阅大量流形学习和人脸识别的相关文献，并学习流形学习相关算法的实现原理如 MDS 算法、ISOMAP 算法、NPE 算法和 IsoProjection 算法，在查阅许多正交化改进算法方法有关的文献后自己动手改进出正交算法，即 ONPE 算法和 OIsoProjection 算法。

实验过程：在 ORL 人脸库和 Yale 人脸库上运行 ONPE 算法和 OIsoProjection 算法，并对比之前学过的其他算法，分析各个算法之间的特点。

结果分析： ONPE 算法和 OIsoProjection 算法确实可以提高原算法的识别率。

存在的问题：（1）改进的算法识别率没有预期的高，还需进一步改进算法。（2）还有许多算法的原理没有搞明白，需要进一步学习。（3）在算法改进过程中遇到许多问题还未解决，因为时间有限，虽然最终程序能跑出来，但是还是不够完美，还有改进的可能。

指导教师评语：

该生顺利完成毕业设计，在 NPE 和 IsoProjection 算法的基础上，通过在人脸库上的实验，证明了算法的有效性，达到了预期效果，并且在理论上分析了正交化算法的可行性，如期完成了设计内容及计划。实践证明，该设计具有可行性，且应用性强，算法具有有效性，设计思想新颖，具有创造性，且应用前景广泛。同意结题，参加毕业答辩。

签名：

年 月 日

学院毕业论文（设计）领导小组意见：

同 意 结 题

（公章）

年 月 日

毕业设计成绩评定表

学院（公章）：信息与电气工程学院　　学号：20152203031

<table>
<tr><td>姓　名</td><td>姜希成</td><td>总成绩</td><td></td></tr>
<tr><td colspan="2">题　目</td><td colspan="2">基于流形学习子空间的人脸识别算法</td></tr>
<tr><td rowspan="2">评阅人评语</td><td colspan="3"></td></tr>
<tr><td colspan="3">签名：　　年　月　日</td></tr>
<tr><td rowspan="2">答辩小组评语</td><td colspan="3"></td></tr>
<tr><td colspan="3">答辩成绩：　　组长签名：　　年　月　日</td></tr>
</table>

注：总成绩=答辩成绩（100%）。总成绩由百分制转换为五级制，填入本表相应位置。

目 录

基于流形学习子空间的人脸识别算法

姜希成

（鲁东大学信息与电气工程学院）

摘 要： 近年来，人脸识别技术得到突飞猛进的发展，与此同时，流形学习方法也取得巨大进展，所以本文以人脸识别为应用背景，对几种常见的流形学习算法进行研究。主要研究了等距映射算法（IsoProjection）、邻域保护嵌入算法（NPE）以及正交的等距映射算法（OIsoProjection）和正交的邻域保护嵌入算法（ONPE），设计实现了 NPE 算法和 IsoProjection 算法。在本文最后将对这些算法进行对比，研究每种算法的优缺点，并在 ORL 人脸库上进行试验，验证这些算法的有效性。

关键字： 人脸识别；流形学习；数据降维；正交化。

Face Recognition Algorithm based on Manifold Learning Subspace

Jiang XiCheng

(School of Information and Electrical Engineering, Ludong University)

Abstract: In recent years, face recognition technology has been developed by leaps and bounds, at the same time, the manifold learning method has made great progress, so this paper takes face recognition as the application background to study several common manifold learning algorithms. This paper mainly studies the isometric mapping Algorithm (IsoProjection), the Neighborhood Protection embedding Algorithm (NPE), the orthogonal isometric mapping algorithm (OIsoProjection) and the orthogonal Neighborhood Protection embedding algorithm (ONPE), The NPE algorithm and IsoProjection algorithm are designed and implemented. At the end of this paper, we will compare these algorithms, study the advantages and disadvantages of each algorithm, and experiment on the ORL face library to verify the effectiveness of these algorithms.

Key words: Face recognition; Manifold Learning; Data reduction dimension; Orthogonal .

1 引言

人脸识别技术和扩展已应用于生活的各个方面，人脸识别技术具有广泛的应用前景，可以用于考勤、门禁、关口同行、社区安防、民航、保险、军事安全、银行金融系统、追击嫌疑犯和反恐等等。近年来，人脸识别技术及相关算法的飞速发展也为该技术提供了强有力的支持[10]。

人脸识别技术在生物特征识别技术中占据非常重要的位置，本文将介绍如何使用流形学习子空间方法来实现人脸识别技术。

1.1 人脸识别技术简介

党的十九大报告提出要推进互联网、大数据、人工智能和实体经济的深度融合，人脸识别技术作为人工智能与实体经济相结合的重要技术支撑，近几年来得到飞速发展。

1.1.1 人脸识别技术的发展情况

人脸识别在生物特征识别技术中占据非常重要的位置，人脸识别技术不仅具有非侵犯性，而且符合人们自身识别习惯，是一种非常“人性化”的技术。人脸识别技术从上世纪 60 年代开始，到现在已经发展了将近 60 年，在这 60 年中人脸识别技术得到突飞猛进的发展，相继出现了许多经典算法、经典思想和经典的人脸数据库。

当前，人脸识别技术的最高识别率已经超过了 99.5%，然而，在同等条件下的最高人眼识别率仅为 97.52%。所以，人脸识别的准确率已经做到了比肉眼更精准，我相信在未来人脸识别技术一定能我们创造更加便利的生活条件。

1.1.2 人脸识别技术的优势

人脸识别是生物识别的一种，生物识别因为具有易检测、唯一性和终身不变

的特点，所以十分适合互联网时代用户对安全的需求，如今生物识别技术的识别速度更快，准确率更高，因此也更具有研究的价值。

目前主流的生物识别技术有人脸识别、指纹识别、虹膜识别、语音识别、静脉识别等。相对于其他的生物识别技术，人脸识别技术的成本更低、稳定性更好、准确率更高，因此也更具有发展优势，所以在多数应用场景中都会首选人脸识别技术。

1.1.3 人脸识别技术的市场前景

市场前景是判断一项技术是否有研究价值的重要指标。人脸识别市场前景非常好，发展势头也非常迅速，2017 年全球人脸识别的市场规模超过 40.5 亿美元，预计 2022 年达到 77.6 亿美元，复合年增长率高达 13.9%。

根据前瞻产业研究院的报告，目前，人脸识别技术主要应用在考勤和门禁等领域，其市场份额比率约占 42%；安防是人脸识别技术最早的应用领域之一，它的市场份额比率约占 30%；作为未来人脸识别的重要应用领域之一，金融领域，其市场份额比率约占 20%。从市场需求的产品结构来看，嵌入式设备约占人脸识别市场的 53%，剩下的是软件开发包(SDK)支持的联机应用。

同时大规模普及的软硬件基础条件已具备，产品系列达 20 多种类型，金融、安防、互联网等主要下游领域需求强劲，这一切都标志着人脸识别技术的爆点已经到来。

1.1.4 人脸识别系统的研究内容

图 1 显示了人脸识别的一般步骤：[2]

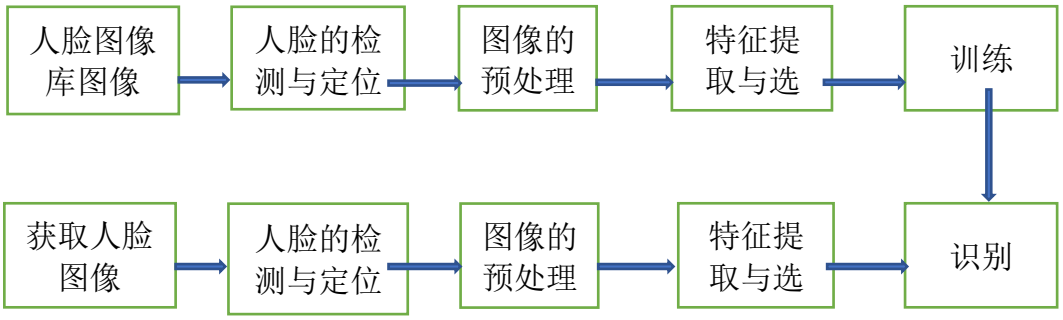

图 1 人脸识别流程图

可见人脸识别系统一般包括人脸检测与定位、图像预处理、人脸图像的特征提取与选择等。横线上方的是人脸识别系统的训练部分，横线下方的人脸识别系统的识别部分。

人脸识别系统在进行特征提取之前需要进行人脸检测与定位和图像预处理，这些操作主要包括几何归一化和灰度归一化。几何归一化是根据人脸检测定位的结果将图像中的人脸变换到同一位置和大小，灰度归一化是将图像进行光照补偿等处理来克服图像中的光照变化对人脸识别的影响。人脸识别过程是将待识别图像的特征和库中的特征进行比对匹配来确定图像中人脸的身份。

1.2 流形学习

流形学习的主要思想是将高维的数据映射到低维，使该低维的数据能够反映原高维数据的某些本质结构特征。流形学习的前提是有一种假设，即某些高维数据，实际是一种低维的流形结构嵌入在高维空间中。流形学习的目的是将其映射回低维空间中，揭示其本质。[14][34]

1.2.1 流形学习的研究背景

随着互联网时代的发展，在各个研究领域，每时每刻都在快速生成大量的数据，而在这些数据背后的规律却难以发现，人们虽然获取了海量的信息却发现自己正处于“数据丰富，知识匮乏”的尴尬境地，因此如何从海量信息中提取自己所需的知识是当今各个领域共同面临的巨大挑战。[22]

在许多实际应用中，尤其是在人脸识别中，往往要对成千上万张图片进行处理，而每张图片又有极高的维度，这种高维的特质往往隐藏了数据间关系的本质，对于传统的数据分析方法往往会造成“维数灾难”，这种情况就需要通过降维来把高维空间的数据间的关系映射到低纬度空间，这样就可以更加方便快速的处理数据。因此降维就成了这一任务吸引了许多科研人员的注意，也成为了如人脸识别、机器学习和数据挖掘等领域的热门研究问题。而流形学习就是解决这类问题的方案之一。

自从流形学习方法被提出到现在，研究它的工作就一直在进行着，特别是近年来随着数据挖掘和人脸识别等技术的高速发展，“维度灾难”的问题就成了相关研究领域的重大障碍。本文将利用流行学习方法解决这类问题。[13][20]

1.2.2 流形学习的发展情况

1984 年斯坦福大学统计系的 Hastie 在一份技术报告中首次提出主曲线和主曲面的概念。“流形学习”的概念在 1995 年由 Bregler 和 Omohundro 首次提出，主要用于语音识别和图像插值中。2000 年 Seung 和 Lee 在《科学》杂志上发表《认知的流行模式》一文，提出了视觉感知的流行结构假说。就在同年同期的《科学》上还刊登了另外两篇著名的文章，它们提出了两个经典算法 LLE 和 Isomap。随着后期学者不断地深入研究，又有许多经典算法相继被提出，比如 Belkin 和 Niyogi 提出拉普拉斯特征映射算法（LE），使高维空间相近的点映射到低微空间时也相近；Donoho 和 Grimes 提出海森特征映射（HLLE），是对 LLE 算法的扩展；He 和 Hiyogi 提出局部保持投影算法（LPP），是对 LE 算法的线性扩展；Zhang 和 Zha 提出局部切空间校准算法（LTSA），基于“局部拟合，全局整合”的思想；Lin 和 Zha 提出黎曼流行（RML），利用局部黎曼正交坐标系将高维空间的数据映射到低微本质空间中去。这些经典算法被提出后，又有许多学者为了弥补这些算法的缺陷而相继提出了很多经典的改进算法。

此外，很多学者发现不同的流形学习算法之间存在着一定的联系，又提出了一些框架将多种流形学习算法纳入其中，比如经典的核主成分分析框架（KPCA）将 MDS、LLE、LE、Isomap 和谱聚类进行统一，GA 框架在将 LE、LLE、LPP 和 Isomap 等流形学习方法纳入其中的同时又将 PCA 和 LDA 等传统线性降维方法统一进去。[9]

本文将介绍 LGE（线性图嵌入）框架，该框架由浙江大学计算机科学学院的蔡登教授等人开发，为基于图的子空间学习提供一般框架，该框架将 LPP，NPE，IsoProjection，LSDA，MMP 等流形学习算法进行统一，本文将在 2.5 节中详细介绍 LGE 算法的推导过程和算法实现流程。[19]

1.2.3 流形的定义

关于豪斯多夫（Hausdorff）空间，直观的理解：如果某空间中任两点可用开集合将彼此“豪斯多夫”开来，该空间就是“豪斯多夫”的。

拓扑流形：设 M 是豪斯多夫（Hausdorff）空间，若对 M 中的任意一点x ∈ M，都存在 x 的一个邻域 U 与R^d中的一个开集同胚，则称 M 是一个 d 维的拓扑流形，简称 d 维流形。[4]

根据上述定义可以看出，欧氏空间R^d本身就是一个 d 维流形。除了拓扑流形，学者们还进一步定义了微分流形、光滑流形、等距流形和黎曼流形等概念。[11][12][15][17]

1.2.4 流形学习的定义

给定高维数据集X = $\{x_1, x_2, \ldots, x_N\} \subset R^D$，并假设 X 中的样本是由低维空间中的数据集y = $\{y_1, y_2, \ldots, y_N\} \subset R^d$(d << D)通过某个未知的线性变换 g 所生成，即：

$$x_i = g(y_i) + \varepsilon_i$$

其中，ε_i表示噪声，g: $R^d \to R^D$是C^∞的嵌入映射。那么流形学习的目的就是通过观测数据集 X：[5]

（1） 获取低维表示y = $\{y_1, y_2, \ldots, y_N\} \subset R^d$。

（2） 构造高维空间到低维空间的非线性降维映射$g^{-1}: R^D \to R^d$。

1.3 小结

本文主要用到 NPE 算法和 IsoProjection 算法并详细介绍算法实现原理，本文会在后面介绍正交的 NPE 和 IsoProjection 算法，并对比这些算法的优缺点。

2 流形学习相关算法

NPE 和 IsoProjection 算法的实现需要用到 LGE 算法，IsoProjection 算法因为涉及最短路径选择所以用到迪杰斯特拉算法，而 IsoProjection 算法又是基于

ISOMAP 算法的改进，ISOMAP 算法又涉及到 MDS 算法。通过 LGE 算法构造 OLGE 算法要用到规范正交基。

2.1 规范正交基

定义：设V $\subset R^n$是一个向量空间。

（1）若$\alpha_1, \alpha_2, \dots, \alpha_r$是向量空间 V 的一个基，且是一个两两正交的向量组，那么就称$\alpha_1, \alpha_2, \dots, \alpha_r$是向量空间 V 的一个正交基。

（2）若$e_1, e_2, \dots, e_r$是向量空间 V 的一个基，$e_1, e_2, \dots, e_r$两两正交，且都是单位向量，则称$e_1, e_2, \dots, e_r$是向量空间 V 的一个规范正交基（或标准正交基）。

求法：设$\alpha_1, \dots, \alpha_r$是向量空间 V 的一个基，要求 V 的一个规范正交基，也就是要找一组两两正交的单位向量$e_1, \dots, e_r$，使$e_1, \dots, e_r$与$\alpha_1, \dots, \alpha_r$等价。这一过程称为把基$\alpha_1, \dots, \alpha_r$规范正交化。

可以先进行施密特正交化在进行单位化。本节接下来将详细介绍施密特正交化和单位化的计算方法。

2.1.1 施密特(Schmidt)正交化

令：

$\beta_1 = \alpha_1$；

$\beta_2 = \alpha_2 - \frac{[\beta_1, \alpha_2]}{[\beta_1, \beta_1]}\beta_1$；

……

$\beta_r = \alpha_r - \frac{[\beta_1, \alpha_r]}{[\beta_1, \beta_1]}\beta_1 - \frac{[\beta_2, \alpha_r]}{[\beta_2, \beta_2]}\beta_2 - \dots - \frac{[\beta_{r-1}, \alpha_r]}{[\beta_{r-1}, \beta_{r-1}]}\beta_{r-1}$，

则易验证$\beta_1, \dots, \beta_r$两两正交，且$\beta_1, \dots, \beta_r$与$\alpha_1, \dots, \alpha_r$等价。且满足：对任何 k$(1 \le \mathrm{k} \le \mathrm{r})$，向量组$\beta_1, \dots, \beta_k$与$\alpha_1, \dots, \alpha_k$等价。[24]

2.1.2 单位化

令：

$$e_1 = \frac{\beta_1}{\|\beta_1\|}，\ e_2 = \frac{\beta_2}{\|\beta_2\|}，\ \dots，\ e_r = \frac{\beta_r}{\|\beta_r\|}$$

则$e_1, e_2, \dots, e_r$是 V 的一个规范正交基。

施密特正交化过程中可将R^n中的任一线性无关的向量组$\alpha_1, \dots, \alpha_r$化为与之等价的正交向量组$\beta_1, \dots, \beta_r$；在经过单位化，得到与$\alpha_1, \dots, \alpha_r$等价的规范正交向量组$e_1, e_2, \dots, e_r$。

2.2 迪杰斯特拉(Dijkstra)算法

迪杰斯特拉(Dijkstra)提出了一个按路径长度递增的次序产生最短路径的算法。该算法利用广度优先搜索的思想，从起始点向外逐渐搜索，直到找到最终点为止。

2.2.1 迪杰斯特拉(Dijkstra)算法流程

1. 准备工作。指定一个起点 s，并且引入两个集合 S 和 U：S 集合记录已算出最短路径的顶点和相应最短路径的长度；U 集合记录还未求出最短路径的顶点和该顶点到 S 的距离，若不相邻则为∞。
2. S 中只有 s，U 中是其他顶点。
3. 将 U 中的最短顶点移除并加入 S 中。
4. 更新 U。
5. 重复 3、4 步，直到遍历完所有顶点。[27]

2.3 多维缩放(MDS)算法

MDS 与 PCA 一样，是一种有效的降维方式，其可获得样本间相似性的空间表达。MDS 的原理可以简述为，利用样本的成对相似性，构建一个低维空间，使每对样本在高维空间的距离与在构建的低维空间中的样本相似性尽可能保持一致。

2.3.1 MDS 算法推导

MDS 算法的核心思想是：降维前后，各自样本间的距离是不变的。由此可得到如下关系：假设 m 个样本在原始空间的距离矩阵为$\text{dist} = (dist_{ij})_{m\times m}$，则

原空间中的两样本$\mathrm{x}_i, \mathrm{x}_j$之间的距离$\mathrm{dist}(\mathrm{i}, \mathrm{j})$等于降维后这两样本$\mathrm{z}_i, \mathrm{z}_j$之间的距离$\|\mathrm{z}_i - \mathrm{z}_j\|$，即$\mathrm{dist}_{ij} = \|\mathrm{z}_i - \mathrm{z}_j\|$，B 为降维后样本的内积矩阵：

$$\mathrm{B} = Z^T Z \in R^{m \times m},\ b_{ij} = z_i^T z_j \tag{1}$$

Z 就是样本集在降维后空间的坐标矩阵。

所以，根据降维前后各样本间欧式距离保持不变，得：

$$\mathrm{dist}_{ij}^2 = \|\mathrm{z}_i\|^2 + \left\|\mathrm{z}_j\right\|^2 - 2z_i^T z_j = b_{ii} + b_{jj} - 2b_{ij} \tag{2}$$

假设降维后的样本集 Z 被中心化，即$\sum_{i=1}^{m} \vec{z_i} = \vec{0}$，则矩阵 B 的每行之和均为零，每列之和均为零，即：

$$\sum_{i=1}^{m} b_{i,j} = 0,\ j = 1,2,\dots,m$$

$$\sum_{j=1}^{m} b_{i,j} = 0,\ i = 1,2,\dots,m$$

为表示方便将 dist 简写为 d，则有：

$$\sum_{i=1}^{m} d_{ij}^2 = \sum_{i=1}^{m} b_{i,i} + N b_{j,j} = tr(B) + N b_{j,j}$$

$$\sum_{j=1}^{m} d_{ij}^2 = \sum_{j=1}^{m} b_{j,j} + N b_{i,i} = tr(B) + N b_{i,i}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}^2 = \sum_{i=1}^{m} (tr(B) + N b_{i,i}) = 2N tr(B)$$

其中 tr 表示矩阵的迹。

令：

$$d_{i,\cdot}^2 = \frac{1}{N}\sum_{j=1}^{m} d_{ij}^2 = \frac{tr(B)}{N} + b_{i,i} \tag{3}$$

$$d_{j,\cdot}^2 = \frac{1}{N}\sum_{i=1}^{m} d_{ij}^2 = \frac{tr(B)}{N} + b_{j,j} \tag{4}$$

$$d_{\cdot,\cdot}^2 = \frac{1}{N^2}\sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}^2 = 2\frac{tr(B)}{N} \tag{5}$$

代入(2)得：

$$b_{ij} = -\frac{1}{2}\left(dist_{ij}^2 - dist_{i\cdot}^2 - dist_{\cdot j}^2 + dist_{\cdot\cdot}^2\right) \tag{6}$$

现在的问题是，已知 B 如何求 Z。

对 B 做特征值分解：

$$\mathrm{B} = \mathrm{V}\mathrm{A}\mathrm{V}^T \tag{7}$$

这里 V 是特征向量矩阵、A 是由特征值构成的对角阵。此时，把 A 中的特征值排序后，把其中每个非 0 特征值拿出来构成对角矩阵A_*，其对应的特征向量V_*也需按特征值的大小改变排列顺序，组成新的特征向量矩阵。最终，通过以下公式完成降维操作（降到 d 维）：

$$\mathrm{Z} = A_*^{1/2} V_*^T \in R^{d\times m} \tag{8}$$

另外，在现实应用中为了有效降维，往往只需要降维后两样本间的距离应尽可能和原空间中两样本的距离相近就好了，不需要强行一致，因此上面特征值构成的对角阵和特征向量矩阵有了一些变化。

变化为：本来特征值取的是所有非 0 的特征值排序。现在变成排序好后，从大到小取特征值获得特征值对角矩阵$\widetilde{\mathrm{A}}$与其对应的特征向量矩阵$\tilde{V}$。比如要降维到 n 维，就从大到小取 n 个特征值。则通过以下公式完成降维操作（降到 n 维）：[43]

$$\mathrm{Z} = \tilde{A}^{1/2}\tilde{V}^T \in R^{n\times m} \tag{9}$$

2.3.2 MDS 算法流程

输入：dist $\in R^{m\times m}$，低维空间数 d。

过程：

根据式(3)，式(4)和式(5)计算$d_{i,\cdot}^2$，$d_{j,\cdot}^2$和$d_{\cdot,\cdot}^2$。

根据式(6)计算矩阵 B。

根据式(7)对矩阵 B 进行特征值分解。

取 n 个最大的特征值构成对角矩阵$\widetilde{\mathrm{A}}$，并获得特征向量矩阵$\tilde{V}$。

根据式(9)完成降维，获得低维空间中的矩阵 Z。

输出：低维空间中的矩阵 Z。

2.4 等度量映射(ISOMAP)算法

ISOMAP 算法可以理解为是对 MDS 算法的改造，也就是将原始空间中的距离计算从欧氏空间中的欧氏距离转换为流形上的测地距离。

由于流形结构是未知的，所以需要构造近邻连接图来计算最短路径问题。

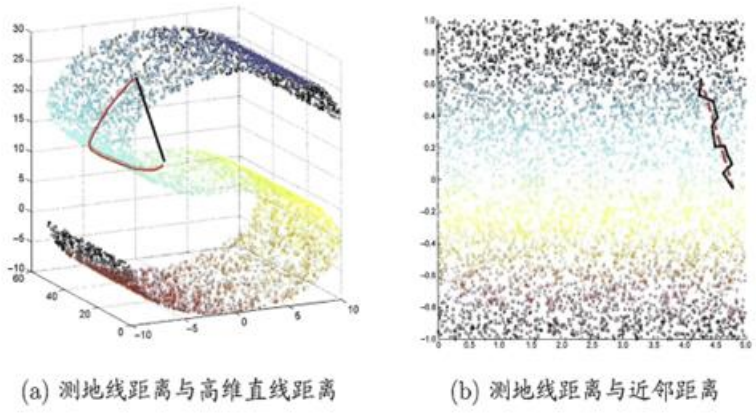

图 2 ISOMAP 原理

从图 2 中可知，低维嵌入流形上的测地线距离（红色）不能用高维空间的直线距离计算，但能用近邻距离来近似。[18]

2.4.1 ISOMAP 算法步骤

ISOMAP 算法分为三步：

1. 构造近邻连接图。利用流形与欧氏空间在局部上同胚的性质，使用欧氏距离找出每个点在低维流形上的近邻点，建立近邻连接图。
2. 计算测地距离。计算近邻连接图上任意两点的最短路径。
3. 数据嵌入。使用 MDS 算法。[21]

2.4.2 ISOMAP 算法流程

输入：样本集$\mathrm{D}=\{\mathrm{x}_1,\mathrm{x}_2,\ldots,\mathrm{x}_N\}$；近邻参数 k；低维空间维数 n。

过程：

对每个样本点x_i，计算它的 k 近邻；同时x_i与它的 k 近邻的距离设置为欧氏距离，与其他点的距离设置为无穷大。

通过最短路径算法计算任意两个样本点之间的距离，获得距离矩阵$D \in R^{N\times N}$。

通过 MDS 算法，获得样本集在低维空间中的矩阵。

输出：样本集 D 在低维空间的投影$Z = \{z_1, z_2, \ldots, z_n\}$。

2.4.3 ISOMAP 算法的特点

ISOMAP 算法的优点是使用了测地距离来计算样本间的距离，这样可以更好地反映流形结构，可以减少数据的流失，使高维数据可以更全面的映射到低维空间。

ISOMAP 算法还有一个很大的缺点，对于新的样本，难以将其映射到低维空间。理论上虽然可以将新样本添加到样本集中，然后重新调用 ISOMAP 算法，但是这种方法的计算量太大了。解决方法是：训练一个回归学习器来对新样本的低维空间进行预测。再者 ISOMAP 算法本身的拓扑不稳定性会导致图的错误链接，其次 ISOMAP 算法针对的是凸形流形，对于其他流形 ISOMAP 算法将无法处理。ISOMAP 算法对流形中的“孔洞”也不能很好地处理。[42]

2.5 线性图嵌入算法(LGE)

为基于图的子空间学习提供一般框架。该算法将由 NPE 和 IsoProjection 调用。

2.5.1 LGE 算法推导

数据集$X = \{w_1, \ldots, w_n\}, w_i \in R^m$，n 表示样本数量，m 表示样本维度。LGE 算法采用无向有权图$G(V, W)$描述数据集的流形结构，顶点集$V = \{v_1, \ldots, v_n\}$对应数据w_i，$W = \{w_{ij}\}_{n\times n}$，$w_{ij}$表示$v_i, v_j$边的权重，W 矩阵是对称的。LGE 算法在保持图的邻接关系的前提下，寻找 X 的低维表示。令$y = \{y_1, \ldots, y_n\}^T$表示

从图到实线的映射，则 LGE 的目标函数如下：

$$\min \sum_{i,j} (y_i - y_j)^2 w_{ij} \tag{10}$$

变换后可得：

$$\frac{1}{2}\sum_{i,j} (y_i - y_j)^2 w_{ij} = y^T L y,\ L = D - W \tag{11}$$

L 是图的拉普拉斯矩阵，D 是对角矩阵，D_{ii}对应 W 矩阵第i列（行）所有元素之和，即$D_{ii} = \sum_j w_{ji}$。

令$y^T D y = 1$（消除嵌入时的量化影响），则：

$$\text{argmin}_{y^T D y=1} y^T L y = argmin \frac{y^T L y}{y^T D y} \tag{12}$$

若图到实线的映射为线性，则$\text{y} = X^T a$，则：

$$\text{argmin}_{y^T D y=1} y^T L y = argmin \frac{a^T X L X^T a}{a^T X D X^T a} \tag{13}$$

则最优向量a对应于以下最小特征值所对应的特征向量。

$$\text{XL}X^T a = \lambda X D X^T a \tag{14}$$

矩阵$\text{XD}X^T$通常是奇异的，所以 LGE 算法要先把 X 集投影到 PCA 空间，使$\text{XD}X^T$变成可逆矩阵，然后求解特征值问题。经过 PCA 处理后上式可通过奇异值分解进行求解：

$$(X D X^T)^{-1}(X L X^T)\text{w} = \lambda \text{w} \tag{15}$$

可解出W_{LGE}，设W_{PCA}为 PCA 空间特征向量，则 LGE 算法的解为：

$$\text{W} = \text{W}_{PCA} W_{LGE} \tag{16}$$

$(X D X^T)^{-1}(X L X^T)$矩阵是非对称的，所以 LGE 算法的解往往是非正交的。本节会在接下来介绍正交的 LGE 算法(OLGE)。

许多常用的线性子空间算法如 LPP 等，都可以通过定义其权重矩阵 W 将该算法统一于 LGE 框架下，通过

$$\text{XL}X^T a = \lambda X D X^T a \tag{17}$$

进行求解。[25][26][41]

2.6 局部线性嵌入算法（LLE）

LLE 是流形学习方面经典的局部非线性方法，它试图保留邻域内样本之间的线性关系。它有参数少、计算快、易求全局最优并在图像分类、图像识别、谱重建、数据可视化等方面都有着广泛的应用。

2.6.1 LLE 算法原理

矩阵$\mathrm{X}=[x_1,x_2,\ldots,x_N]$是数据矩阵，包括所有的训练样本 $x_{i\,i=1}^{N}\in R^m$。维度m通常是很大的，（稀疏）线性降维的目标是变换数据从原来的高维空间到低维空间。

$$\mathrm{y}=A^Tx\in R^d \tag{18}$$

$d\ll \mathrm{m}$，$x\in R^m$，$\mathrm{A}=(a_1,a_2,\ldots,a_d)$并且$a_i(i=1,\ldots,d)$是一个 m 维列向量。

若相邻样本处于非线性流形的局部线性片上，则每个样本x_i可由其 k 个最近邻居的加权线性组合近似表示。最小化如下成本函数：

$$\varepsilon(W)=\sum_i\left\|x_i-\sum_{j\in N_k(x_i)}W_{ij}x_j\right\|^2 \quad s.t.\sum_{j\in N_k(x_i)}W_{ij}=1 \tag{19}$$

$N_k(x_i)$ 是x_i的 k 个最近邻居索引集，W_{ij}是最佳局部最小二乘重建系数。得到 W 后，可以通过最小化一下函数来得到最终嵌入坐标。

$$\varepsilon(\mathrm{Y})=\sum_i\left\|y_i-\sum_j W_{ij}y_j\right\|^2=tr(Y(I-W)^T(I-W)Y) \quad s.t.YY^T=I \tag{20}$$

$Y=[y_1,y_2,\ldots,y_d]$ 。对应于特征函数的较小特征值的特征向量是数据集的最终嵌入。

$$(I-W)^T(I-W)y=\lambda y \tag{21}$$

其中y是对应于特征值λ的特征向量。

2.6.3 LLE 算法流程

输入：数据集 X，邻居个数 k，降至的维度 d。

第一步：计算对应距离和寻找邻居。

先计算向量x_i到x_j的距离，得到距离矩阵 distance，然后每行排序，取前 k 个，对应原下标就是近邻点的编号。

第二步：解决重建权重问题。

对于任意 i，若 j 不属于$N_k(x_i)$，则$W_{ij}=0$，故 W 只需存 K 行 N 列。

当且仅当每一项求和项极小时，$\varepsilon(W)$极小。因此需要计算每一项的极小值，也就是极小化$\left\|x_i-\sum_{j\in1,\dots,k}W_{ij}x_j\right\|^2\ s.t.\sum_{j\in1,\dots,k}W_{ij}=1$。

第三步：从成本矩阵的特征嵌入计算$\mathrm{M}=(\mathrm{I}-\mathrm{W})^T(\mathrm{I}-\mathrm{W})$。

第四步：嵌入计算。

输出：降维后的矩阵 Y。

2.7 等距映射算法（IsoProjection）

IsoProjection 是 ISOMAP 的线性近似。[3]

2.7.1 IsoProjection 算法推导

设$\mathrm{X}=[x_1,\dots,x_N]$为输入数据集，其中$x_i\in R^n$，寻找投影矩阵 W 使得$y_i=W^Tx_i$，其中$y_i\in R^d, d<<n$。Y 就是降维后的数据集。

定义一个距离矩阵 D，D_{ij}表示x_i到x_j之间的测地距离，定义矩阵 S 令$\mathrm{S}_{ij}=D_{ij}^2$，令$\mathrm{H}=\mathrm{I}-\frac{1}{N}EE^T$，I 为单位矩阵，E 为元素全为 1 的列向量，则可得内积矩阵：$\mathrm{P}=-\frac{1}{2}HSH$。则 IsoProjection 的目标函数如下：

$$\mathrm{w}=\mathrm{argmax}_w\frac{w^TXPX^Tw}{w^TXX^Tw}\ s.t.\ w^TXX^Tw=1 \tag{22}$$

使用拉格朗日算法将上式的解向量问题转化为求解下式的特征值和特征向

量的问题。

$$\mathrm{XP}X^T w = \lambda X X^T w \tag{23}$$

所以要求XX^T是非奇异矩阵，因此 IsoProjection 的最佳投影向量就是上式最大特征值所对应的特征向量。

IsoProjection 算法求出来的投影向量往往是非正交的，这种非正交的性质会在从高维空间向低维空间的投影过程中扰乱数据的流形结构，而且会使算法本身对降维后的子空间的维数十分敏感，很难估计样本的内蕴维数。[1]

2.7.2 IsoProjection 算法流程

IsoProjection 算法分五步执行：

1. 先使用 PCA 算法对样本进行降维。因为$\mathrm{X}X^T$是奇异矩阵，所以要进行去奇异处理。使用 PCA 算法的主要目的是将$\mathrm{X}X^T$变为非奇异矩阵，然后在低维空间使用 IsoProjection 算法。
2. 使用 K-近邻方法构造近邻图 G。
3. 使用迪杰斯特拉算法计算最短路径矩阵 D。
4. 计算最佳投影矩阵 W。
5. 特征提取。令$\mathrm{W} = W_{PCA}W_{IsoP}$，则$\mathrm{x} \to \mathrm{y} = V^T x$。

2.8 领域保护嵌入算法（NPE）

2.8.1 NPE 算法推导

令 NPE 算法的权重矩阵为 S，w 为一投影向量，使得满足$y_i = w^T x_i$。则 NPE 算法的目标函数如下：

$$w_{opt} = argmin_w \sum_i \left\| y_i - \sum_j S_{ij} y_j \right\|^2 \quad s.t. w^T X X^T w = I \tag{24}$$

对该目标函数做出如下推导：

$$w_{opt} = argmin_w \sum_i \left\| y_i - \sum_j S_{ij} y_j \right\|^2 = min \| w^T X(I-S) \|^2$$

$$= min\ tr(w^T X(I-S)^T (I-S) X^T w)$$

$$= min\ tr(w^T X M X^T w) \quad (25)$$

上式中，令$M = (I-S)^T(I-S)$。则简化后的目标函数如下：

$$w_{opt} = argmin_w w^T X M X^T w \quad (26)$$

则 NPE 算法的最佳投影向量就是以下特征方程的最小特征值所对应的特征向量。[29][31]

$$\mathrm{XM}X^T w = \lambda X X^T w \quad (27)$$

2.8.2 NPE 算法流程

NPE 算法的实现分为三步：

1. 构造近邻图。使用 K-近邻法寻找与数据点欧氏距离最近的K个近邻点。
2. 确定权值。用近邻对各个数据点进行重构。
3. 计算特征映射。在低维空间中，对各个进行数据点重构，即在保持重构权值不变的情况下，使重构误差最小，计算出降维矩阵。[28][30][39]

3 正交化算法

3.1 OIsoProjection

OIsoProjection 算法是在 IsoProjection 算法的基础上，得到一组正交基向量，因此该算法在保留 IsoProjection 算法线性的特点的同时又能够保持高维数据的流形结构。

OIsoProjection 算法希望通过一组正交基改造目标函数，正交基的求法也相类似，都是通过拉格朗日乘子法引入正交基约束条件来推导出一个特征方程，然后求出该特征方程的最大特征值对应的特征向量以求出正交基，再利用该正交基构造特征方程进行特征提取。

3.1.1 OIsoProjection 算法推导

首先要计算出一组正交基$w_1, w_2, \ldots, w_d$，则目标函数如下：

$$\begin{cases} w_1 = \underset{w}{\operatorname{argmax}} \dfrac{w^T XPX^T w}{w^T XX^T w}, s.t.\ w_1^T XX^T w_1 = 1 \\ w_d = \underset{w}{\operatorname{argmax}} \dfrac{w^T XPX^T w}{w^T XX^T w}, s.t.\ w_d^T XX^T w_d = 1 \\ w_d^T w_i = 0 (i = 0,1, \ldots, d-1, d \geq 2) \end{cases} \tag{28}$$

则向量w_1可以通过求解$\mathrm{XP}X^T w = \lambda XX^T w$特征方程的最大特征值所对应的特征向量获得。

若已有 d-1 个正交基向量$w_1, w_2, \ldots, w_{d-1}(d \geq 2)$。则可以利用如下正交约束条件获得$w_d$：$w_d^T w_1 = w_d^T w_2 = \ldots = w_d^T w_{d-1} = 0$。

利用拉格朗日乘子法得到如下方程：

$$\mathrm{L}(w_d) = w_d^T XPX^T w_d - \lambda (w_d^T XX^T w_d - 1) - \sum_{i=1}^{d-1} \beta_i w_d^T w_i \tag{29}$$

令：$\frac{\partial \mathrm{L}(w_d)}{\partial w_d} = 0$，则：

$$2\mathrm{XP}X^T w_d - 2\lambda XX^T w_d - \sum_{i=1}^{d-1} \beta_i w_i = 0 \tag{30}$$

方程两边同时左乘w_d^T：

$$2w_d^T \mathrm{XP}X^T w_d - 2\lambda w_d^T XX^T w_d = 0 \tag{31}$$

得：

$$\lambda = \frac{w_d^T \mathrm{XP}X^T w_d}{w_d^T XX^T w_d} \tag{32}$$

将(30)式方程两边同时左乘$\mathrm{w}_1^T (XX^T)^{-1}, \mathrm{w}_2^T (XX^T)^{-1}, \ldots, \mathrm{w}_{d-1}^T (XX^T)^{-1}$，得到 d-1 个式子：

$$2w_1^T (XX^T)^{-1} \mathrm{XP}X^T w_d - \sum_{i=1}^{d-1} \beta_i w_1^T (XX^T)^{-1} w_i = 0$$

$$2w_2^T (XX^T)^{-1} \mathrm{XP}X^T w_d - \sum_{i=1}^{d-1} \beta_i w_2^T (XX^T)^{-1} w_i = 0$$

……

$$2w_{d-1}^{T}(XX^{T})^{-1}\mathrm{XP}X^{T}w_{d}-\sum_{i=1}^{d-1}\beta_{i}w_{d-1}^{T}(XX^{T})^{-1}w_{i}=0$$

为方便求解做出如下定义：

$\beta^{(d-1)}=[\beta_1,\beta_2,\ldots,\beta_{d-1}]^T$；

$W^{d-1}=[w_1,w_2,\ldots,w_{d-1}]^T$；

$S_{ij}^{(d-1)}=w_i^T(XX^T)^{-1}w_j$；

$S^{(d-1)}=[S_{ij}^{(d-1)}]=[W^{d-1}]^T(XX^T)^{-1}W^{d-1}$；

所以就可以用矩阵形式表示这$\mathrm{d}-1$个方程了：

$$2\left[W^{(d-1)}\right]^{T}(XX^{T})^{-1}\mathrm{XP}X^{T}w_{d}-S^{(d-1)}\beta^{(d-1)}=0 \tag{33}$$

得：

$$\beta^{(d-1)}=2\left[S^{(d-1)}\right]^{-1}\left[W^{(d-1)}\right]^{T}(XX^{T})^{-1}\mathrm{XP}X^{T}w_{d} \tag{34}$$

将(30)两边同乘$(\mathrm{XX}^T)^{-1}$，得：

$$2(\mathrm{XX}^{T})^{-1}\mathrm{XP}X^{T}w_{d}-2\lambda w_{d}-\sum_{i=1}^{d-1}\beta_{i}(\mathrm{XX}^{T})^{-1}w_{i}=0 \tag{35}$$

矩阵表示：

$$2(\mathrm{XX}^{T})^{-1}\mathrm{XP}X^{T}w_{d}-2\lambda w_{d}-(\mathrm{XX}^{T})^{-1}W^{(d-1)}\beta^{(d-1)}=0 \tag{36}$$

由(34)和(36)得：

$$\left\{I-(\mathrm{XX}^{T})^{-1}W^{(d-1)}\left[S^{(d-1)}\right]^{-1}\left[W^{(d-1)}\right]^{T}\right\}(\mathrm{XX}^{T})^{-1}\mathrm{XP}X^{T}w_{d}=\lambda w_{d} \tag{37}$$

得：

$$U^{(d)}=\left\{I-(XX^{T})^{-1}W^{(d-1)}\left[S^{(d-1)}\right]^{-1}\left[W^{(d-1)}\right]^{T}\right\}(XX^{T})^{-1}XPX^{T} \tag{38}$$

得特征方程：

$$U^{(d)}w_{d}=\lambda w_{d} \tag{39}$$

可通过求解上式最大特征值对应的特征向量得到正交基w_k。

所以，OIsoProjection 算法的正交投影向量就是特征方程$U^{(d)}w_d=\lambda w_d$的最大特征值所对应的特征向量。[7][8][23][32]

3.1.2 OIsoProjection 算法流程

OIsoProjection 算法的前三步与 IsoProjection 算法的步骤相同，此处不再赘述。直接从第四步开始介绍：

4．计算正交投影向量。

通过 5.2.1 中的算法计算出一组正交基$\{w_1, w_2, \ldots, w_d\}$，具体流程如下：

(1) 通过求解XP$X^T w = \lambda X X^T w$特征方程的最大特征值所对应的特征向量获得计算w_1。

(2) 通过求解$U^{(k)} w_k = \lambda w_k$特征方程的最大特征值所对应的特征向量获得$w_k$，$\mathrm{k} = 2,3,\ldots,\mathrm{d}$。

5. 特征提取。

令通过 OIsoProjection 算法得到的最佳投影矩阵为 d 维的W_{OIsoP}，通过 PCA 算法得到的投影矩阵为W_{PCA}，则令$\mathrm{V} = W_{PCA} W_{OIsoP}$，则通过线性嵌入：

$$\mathrm{x} \to \mathrm{y} = V^T x \tag{40}$$

可以得到对任意 x 的 d 维特征。

3.2 ONPE

ONPE 算法在 NPE 算法的基础上添加了正交迭代处理。改进的思路与 OIsoProjection 算法类似，算法的推导过程也与 OIsoProjection 算法类似。

3.2.1 ONPE 算法推导

定义邻域保护函数为：

$$\mathrm{f(a)} = \frac{a^T X M X^T a}{a^T X X^T a} \ , a \in R^n \tag{41}$$

最小化该函数将得到 NPE 算法，而 ONPE 算法既要找到一组正交向量又要满足上式最小，因此 ONPE 算法的目标函数如下：

$$\begin{cases} a_1 = \underset{a}{\operatorname{argmin}} \dfrac{a^T XMX^T a}{a^T XX^T a} \\ a_k = \underset{a}{\operatorname{argmax}} \dfrac{a^T XMX^T a}{a^T XX^T a} \\ a_k^T a_1 = a_k^T a_2 = \cdots = a_k^T a_{k-1} = 0, a_k^T XX^T a_k = 1 \end{cases} \tag{42}$$

a_1可以通过求解$(\mathrm{X}X^T)^{-1}XMX^T$的最小特征对应的特征向量获得。

若已有 k-1 个正交向量$a_1, a_2, \ldots, a_{k-1}$，则第 k 正交向量满足以下条件：

$$\min \frac{a_k^T XMX^T a_K}{a_k^T XX^T a_K} \tag{43}$$

约束条件：$a_k^T a_1 = a_k^T a_2 = \cdots = a_k^T a_{k-1} = 0, a_k^T XX^T a_k = 1$。

利用拉格朗日乘子法得到如下方程：

$$C^{(k)} = a_k^T XMX^T a_K - \lambda(a_k^T XX^T a_K - 1) - \mu_1 a_k^T a_1 - \cdots - \mu_{k-1} a_k^T a_{k-1} \tag{44}$$

令$\frac{\partial \mathrm{C}^{(\mathrm{k})}}{\partial a_k} = 0$，则：

$$2XMX^T a_K - 2\lambda XX^T a_K - \mu_1 a_1 - \cdots - \mu_{k-1} a_{k-1} = 0 \tag{45}$$

得：

$$\lambda = \frac{a_k^T XMX^T a_K}{a_k^T XX^T a_K} \tag{46}$$

将(45)式两边同时左乘$a_j^T (XX^T)^{-1}, j = 1,2,\ldots,k-1$，得到 k-1 个式子：

$$2a_j^T (XX^T)^{-1} XMX^T a_K = \mu_1 a_j^T (XX^T)^{-1} a_1 + \cdots + \mu_{k-1} a_j^T (XX^T)^{-1} a_{k-1} \tag{47}$$

其中$j = 1,2,\ldots,k-1$。

为方便求解做如下定义：

$\mu^{(k-1)} = (\mu_1, \mu_2, \ldots, \mu_{k-1})^T$；

$A^{(k-1)} = (a_1, a_2, \ldots, a_{k-1})^T$；

$B^{(k-1)} = [B_{ij}^{(k-1)}] = [A^{(k-1)}]^T (XX^T)^{-1} A^{(k-1)}$；

所以就可以用矩阵形式表示这$\mathrm{k}-1$个方程了：

$$B^{(k-1)} \mu^{(k-1)} = 2\left[A^{(k-1)}\right]^T (XX^T)^{-1} XMX^T a_K \tag{48}$$

得：

$$\mu^{(k-1)} = 2\left[B^{(k-1)}\right]^{-1} \left[A^{(k-1)}\right]^T (XX^T)^{-1} XMX^T a_K \tag{49}$$

将(45)式两边同时左乘$(XX^T)^{-1}$，得：

$$2(XX^T)^{-1}XMX^Ta_K - 2\lambda a_K - (XX^T)^{-1}A^{(k-1)}\mu^{(k-1)} = 0 \tag{50}$$

通过(48)式和(50)式得：

$$\left\{I - (XX^T)^{-1}A^{(k-1)}\left[B^{(k-1)}\right]^{-1}\left[A^{(k-1)}\right]^T\right\}(XX^T)^{-1}XMX^Ta_K = \lambda a_K \tag{51}$$

令：

$$U^{(k)} = \left\{I - (XX^T)^{-1}A^{(k-1)}\left[B^{(k-1)}\right]^{-1}\left[A^{(k-1)}\right]^T\right\}(XX^T)^{-1}XMX^T \tag{52}$$

得特征方程：

$$U^{(k)}a_K = \lambda a_K \tag{53}$$

通过求解这个特征方程就能得到一组正交投影向量。

所以，ONPE 算法的正交投影向量就是特征方程$U^{(k)}a_k = \lambda a_k$的最大特征值所对应的特征向量。[6][33][35][36][37][38][40]

3.2.2 ONPE 算法流程

ONPE 算法的前两步与 NPE 算法一样，这里从第三步开始介绍：

3.计算正交投影向量。通过 5.2.1 中的算法计算出一组正交投影向量$\{a_1, a_2, \ldots, a_k\}$。

（1）通过求解$(XX^T)^{-1}XMX^T$的最小特征对应的特征向量获得a_1。

（2）通过求解(5.2.1.5)的最小特征对应的特征向量获得a_k。

4. 特征提取。

令通过 ONPE 算法得到的最佳投影矩阵为 d 维的W_{ONPE}，通过 PCA 算法得到的投影矩阵为W_{PCA}，则令$V = W_{PCA}W_{ONPE}$，则通过线性嵌入：

$$x \to y = V^Tx \tag{54}$$

可以得到对任意 x 的 d 维特征。

4 算法实现

在本节中，将分别运行 NPE、ONPE、IsoProjection 和 OIsoProjection 算法，并对比这些算法，分析这些算法的优缺点。

4.1 常用的人脸数据库介绍

使用公开人脸数据库对程序进行测评是评判算法好坏的主要依据。常用的人脸数据库有：FERET 人脸库，由美国国防部为研发自动人脸识别系统而创建的人脸库；Yale 人脸库和 Yale B 人脸库，由耶鲁大学计算视觉与控制中心创建；ORL 人脸库，由剑桥大学 AT&T 实验室创建；AR 人脸库，由西班牙巴塞罗那计算机视觉中心创建；XM2VTS 人脸库，由欧洲 ACTS 项目的研究计划 M2VTS 资助建立的身份认证资料数据库，是一个商业收费数据库；CMU PIE 库，是卡耐基梅隆大学创建。本文将在 ORL 人脸库和 Yale 人脸库上进行试验。

4.2 在 ORL 人脸库上进行实验

4.2.1 ORL 人脸库介绍

ORL 人脸库由剑桥大学 AT&T 实验室创建，该库由 40 个人，每人 10 张图像，共 400 张图像组成。图像均在不同光照、不同角度和不同表情下获得。每张图片的分辨率均为 112\times92 像素。该库的面部表情表情变化主要是笑/不笑和睁眼/闭眼，还有一些其他的面部细节。拍摄的是受试者的正面、垂直头像，允许倾斜或旋转。

图 3 OLE 人脸库

图 4-1 是 ORL 人脸库的示例图片。

人脸库下载：http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

4.2.2 实验结果

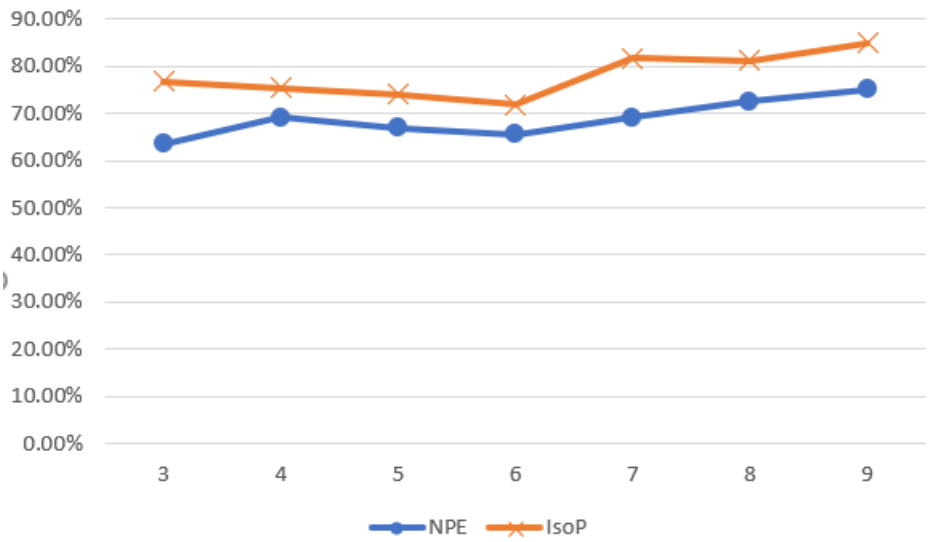

图 4 ORL 上的实验结果

4.3 分析

从上述实验结果可以看出，IsoProjection 算法的识别率要优于 NPE 算法，如果加入正交化实现 ONPE 算法和 OIsoProjection 算法，理论上识别率还会进一步提升。

5 结束语

毕业设计终于顺利完成，实验结果也比较符合预期。其实本文的结构顺序完全符合我的学习历程，在我刚开始接触人脸识别和流形学习的时候，先是查阅了很多介绍流形学习的文献，其中我觉得重要的东西都整理在了第一节中，然后我就开始学习一些经典的流形学习算法，比如 MDS 算法、LLE 算法、LPP 算法、

PCA 算法、NPE 算法、ISOMAP 算法、LGE 框架和 IsoProjection 算法等等，在我确定了毕业设计要研究什么的时候，我有在网上查阅一些如何将算法正交化核化的文献，在学习过程中，我认为我最大的收获是学习了一种思维模式，这给我的思想中种下了一颗种子。

流形学习其实是有章可循的，尽管其在过去几年取得了丰硕的成果，但因其数学理论基础较为复杂，以及学科间存在交叉融合，所以这也意味着流形学习还具有广阔的发展空间。

参考文献

[1] Deng Cai, Xiaofei He, and Jiawei Han. Isometric Projection[R]. Department of Computer Science, University of Illinois at Urbana-Champaign: Yahoo! Research Labs, 2006.

[2] 徐勇，范自柱，张大鹏. 基于稀疏算法的人脸识别[M]. 北京: 国防工业出版社，2014.

[3] 王庆军，张汝波，刘冠群. 一种应用于人脸识别的核正交等度规映射算法[J]. 光电子激光，2010(11): 1702-1705.

[4] 王庆军. 流形学习算法分析及应用研究[D]. 哈尔滨工程大学: 计算机科学与技术学院，2008.

[5] 王庆军. 基于流形学习子空间的人脸识别方法研究[D]. 哈尔滨工程大学: 计算机科学与技术学院，2011.

[6] 王庆军，张汝波，潘海为. 基于核正交局部判别嵌入的人脸识别[J]. 光电子激光，2010(9): 1386-1389.

[7] 王庆军，张汝波，楼宋江，吕海燕. 一种核正交局部敏感辨别分析算法[J]. 小型微型计算机系统，2009(11): 2268-2271.

[8] 王庆军，张汝波. 基于 LogGabor 和正交等度规映射的人脸识别[J]. 计算机科学，2011(2): 274-276+295.

[9] 古楠楠，樊明宇，王迪，韩志. 流形学习若干关键问题与算法研究[M]. 北京: 首都经济贸易大学出版社，2015.

[10] 秦鸿，李泰峰，郭亨艺，许毅. 人脸识别技术在图书馆的应用研究[N]. 大学

图书馆学报，2018(6).

[11] 徐蓉，姜峰，姚鸿勋. 流形学习概述[N]. 智能系统学报，2006(1).

[12] 高小方. 流形学习方法中的若干问题分析[N]. 计算机科学，2009(4).

[13] 王自强，钱旭，孔敏. 流形学习算法综述[N]. 计算机工程与应用，2008(35).

[14] 李波. 基于流形学习的特征提取方法及其应用研究[D]. 中国科学技术大学: 中国科学技术大学研究生院，2008.

[15] 赵连伟，罗四维，赵艳敞，刘蕴辉. 高维数据流形的低维嵌入及嵌入维数研究[D]. 软件学报，2005(16).

[17] 张军平. 流形学习及应用[D]. 中国科学院自动化研究所： 中国科学院研究生院，2003.

[18] 孟德宇,徐晨,徐宗本. 基于 Isomap 的流形结构重建方法[N]. 计算机学报，2010(3).

[19] 李小丽，薛清福. 几种流形学习算法的比较研究[N]. 电脑与信息技术，2009(3).

[20] 王靖. 流形学习的理论与方法研究[D]. 浙江大学: 理学院，2006.

[21] 程起才，王洪元，刘爱萍，冯燕. 基于 ISOMAP 的一种多流形学习算法[N]. 微电子学与计算机，2009(10).

[22] 詹宇斌. 流形学习理论与方法及其应用研究[D]. 国防科学技术大学: 国防科学技术大学研究生院，2011.

[23] 黄结. 基于正交局部敏感辨别分析的人脸识别方法研究[D]. 鲁东大学: 信息与电气工程学院，2013.

[24] 吴赣昌. 线性代数[M]. 北京: 中国人民大学出版社，2011.

[25] 陈江峰. 线性图嵌入算法及其应用[D]. 北京交通大学: 信号与信息处理，2012.

[26] 陈江峰,袁保宗. 直接线性图嵌入算法及其在人脸识别中的应用[N]. 电子与信息学报，2010(6).

[27] 严蔚敏，吴伟民. 数据结构（C 语言版）[M]. 北京: 清华大学出版社，2007.

[28] 郭鹤楠. 基于 NPE 和 LDCRF 的人体运动识别[D]. 吉林大学: 计算机科学与技术(生物信息学)，2011.

[29] 刘嘉敏，李连泽，罗甫林，刘亦哲，刘玉梅. 相关 NPE 算法的人脸识别研究

[N]. 计算机应用研究，2015(6).

[30] 胡凡君，张勤，李鹏，苗爱敏，邹勋，陈霍兴. 基于 NPE 算法的环网柜故障检测方法研究[N]. 自动化仪表，2017(10).

[31] 仝一君，王力. 基于 NPE 算法的语音特征提取应用研究[N]. 通信技术，2014(11).

[32] 左加阔. 基于流形学习算法的新生儿疼痛表情识别[D]. 南京邮电大学: 信号与信息处理，2011.

[33] 陶晓燕，姬红兵，景志宏. 一种用于人脸识别的正交邻域保护嵌入算法[N]. 西安电子科技大学学报（自然科学版），2008(6).

[34] 陶晓燕. 基于支持向量机和流形学习的分类方法研究[D]. 西安电子科技大学: 模式识别与智能系统，2008.

[35] 刘韵佳，赵荣珍，王雪冬. 基于 Schur 分解和正交邻域保持嵌入算法的故障数据集降维方法[N]. 中国机械工程，2017(21).

[36] 刘韵佳. 基于Schur-ONPE的转子故障数据集降维方法研究[D]. 兰州理工大学: 机械制造及其自动化，2017.

[37] 陈法法，杨晓青，陈保家，程珩，肖文荣. 基于正交邻域保持嵌入与多核相关向量机的滚动轴承早期故障诊断[N]. 计算机集成制造系统，2018(8).

[38] 李锋,汤宝平,董绍江. 基于正交邻域保持嵌入特征约简的故障诊断模型[N]. 仪器仪表学报，2011(3).

[39] 苗爱敏. 数据局部时空结构特征提取与故障检测方法[D]. 浙江大学，2014.

[40] 季云峰,冯立元,匡亮. 基于改进的有监督正交邻域保持嵌入的故障辨识[N]. 机械传动，2017(1).

[41] 陈江峰. 线性图嵌入算法及其应用[D]. 北京交通大学: 信号与信息处理，2012.

[42] 华校专，王正林. Python 大战机器学习: 数据科学家的第一个小目标[M]. 北京: 电子工业出版社，2017.

[43] 周志华. 机器学习[M]. 北京: 清华大学出版社，2016.

致 谢

时间仓促且自身水平不足，整篇论文肯定有缺陷或不足之处，恳请阅读此文的老师和同学，多予指正，不胜感激。

在实验室撰写论文和制作毕设期间，遇到了很多困难和阻碍，感谢我的班主任王庆军老师对我悉心教导。在这几个月间，我不仅收获了很多新的知识，最重要的是学会了一些新的思维方式，这为我以后的生活和工作提供了很多新的可能性。在向王老师请教的过程中，他那严谨的科研态度和精益求精的工作作风深深的影响着我。

在学习过程中，我不仅学习了人脸识别技术，更重要的是学会了很多处理和分析问题的能力，坚定了我的意志并且扩充了自己的只是领域，最终顺利完成本次毕业设计。

我之所以能够完成本次毕业设计，少不了老师、朋友和同学的帮助，在这里，衷心的感谢所有关心和帮助过我的人。

6 附录

6.1 主要算法的核心代码

详见中国浙江大学计算机科学学院的蔡登教授的个人主页：

http://www.cad.zju.edu.cn/home/dengcai/Data/DimensionReduction.html

6.1.1 IsoProjection 算法

```
if options.k <= 0    % Always supervised!
      Label = unique(options.gnd);
       nLabel = length(Label);
       G = zeros(nSmp,nSmp);
       for i=1:nLabel
            classIdx = find(options.gnd==Label(i));
            D = EuDist2(data(classIdx,:),[],1);
            G(classIdx,classIdx) = D;
       end
        maxD = max(max(G));
% effectively infinite distance
       INF = maxD*INFratio;
       D = INF*ones(nSmp,nSmp);
       for i=1:nLabel
            classIdx = find(options.gnd==Label(i));
            D(classIdx,classIdx) = G(classIdx,classIdx);
       end
Else
       switch lower(options.NeighborMode)
            case {lower('KNN')}
                 D = EuDist2(data);
```

```
        maxD = max(max(D));
    % effectively infinite distance
        INF = maxD*INFratio;
        [dump,iidx] = sort(D,2);
        iidx = iidx(:,(2+options.k):end);
        for i=1:nSmp
            D(i,iidx(i,:)) = 0;
        end
        D = max(D,D');
        D = sparse(D);
        D = dijkstra(D, 1:nSmp);
        D = reshape(D,nSmp*nSmp,1);
        infIdx = find(D==inf);
        D = reshape(D,nSmp,nSmp);
    case {lower('Supervised')}
        Label = unique(options.gnd);
        nLabel = length(Label);
        G = zeros(nSmp,nSmp);
        maxD = 0;
        for idx=1:nLabel
            classIdx = find(options.gnd==Label(idx));
            nSmpClass = length(classIdx);
            D = EuDist2(data(classIdx,:),[],1);
            if maxD < max(max(D))
                maxD = max(max(D));
            end
            if options.k >= nSmpClass
                G(classIdx,classIdx) = D;
            else
```

```
                    [dump,iidx] = sort(D,2);
                    iidx = iidx(:,(2+options.k):end);
                    for i=1:nSmpClass
                        D(i,iidx(i,:)) = 0;
                    end
                    D = max(D,D');
                    D = sparse(D);
                    D = dijkstra(D, 1:nSmpClass);
                    G(classIdx,classIdx) = D;
                end
            end
            % effectively infinite distance
            INF = maxD*INFratio;
            D = INF*ones(nSmp,nSmp);
            for i=1:nLabel
                classIdx = find(options.gnd==Label(i));
                D(classIdx,classIdx) = G(classIdx,classIdx);
            end
        end
    end
    S = D.^2;
    sumS = sum(S);
    H = sumS'*ones(1,nSmp)/nSmp;
    TauDg = -.5*(S - H - H' + sum(sumS)/(nSmp^2));
    TauDg = max(TauDg,TauDg');
    [eigvector,eigvalue]=LGE(TauDg, [], options, data);
    eigIdx = find(eigvalue < 1e-3);
    eigvalue (eigIdx) = [];
    eigvector(:,eigIdx) = [];
```

6.1.2 NPE 算法

```
if options.k <= 0    % Always supervised!
      W = zeros(nSmp,nSmp);
       for ii=1:nSmp
            idx = find(options.gnd==options.gnd(ii));
            idx(find(idx==ii)) = [];
           % shift ith pt to origin
           z=data(idx,:)-repmat(data(ii,:),length(idx),1);
            C = z*z';                          % local covariance
           % regularlization
            C = C + eye(size(C))*tol*trace(C);
            tW = C\ones(length(idx),1); % solve Cw=1
            tW = tW/sum(tW);    % enforce sum(w)=1
            W(idx,ii) = tW;
       end
       M = (eye(size(W)) - W);
       M = M*M';
       M = max(M,M');
       M = sparse(M);
Else
       switch lower(options.NeighborMode)
            case {lower('KNN')}
                  Distance = EuDist2(data,[],0);
                  [sorted,index] = sort(Distance,2);
                  neighborhood = index(:,2:(1+options.k));
            case {lower('Supervised')}
                   Label = unique(options.gnd);
                 nLabel = length(Label);
```

```
            neighborhood = zeros(nSmp,options.k);
            for idx=1:nLabel
                classIdx=find(options.gnd==Label(idx));
               Distance=EuDist2(data(classIdx,:),[],0);
                [sorted,index] = sort(Distance,2);
                neighborhood(classIdx,:) = classIdx(index(:,2:(1+options.k)));
            end
end
W = zeros(options.k,nSmp);
 for ii=1:nSmp
 % shift ith pt to origin
      z = data(neighborhood(ii,:),:)-repmat(data(ii,:),options.k,1);
      C = z*z';                          % local covariance
     %regularlization
      C = C + eye(size(C))*tol*trace(C);
      W(:,ii) = C\ones(options.k,1);% solve Cw=1
    % enforce sum(w)=1
    W(:,ii) = W(:,ii)/sum(W(:,ii));
end

M = sparse(1:nSmp,1:nSmp,ones(1,nSmp),nSmp,nSmp,4*options.k*nSmp);
for ii=1:nSmp
      w = W(:,ii);
      jj = neighborhood(ii,:)';
      M(ii,jj) = M(ii,jj) - w';
      M(jj,ii) = M(jj,ii) - w;
      M(jj,jj) = M(jj,jj) + w*w';
end
M = max(M,M');
```

```
    M = sparse(M);
end
M = -M;
for i=1:size(M,1)
    M(i,i) = M(i,i) + 1;
end
[evector, evalue] = LGE(M, [], options, data);
eIdx = find(evalue < 1e-10);
evalue (eIdx) = [];
evector(:,eIdx) = [];
```

6.1.3 LGE 算法

```
MAX_MATRIX_SIZE = 1600;
EIGVECTOR_RATIO = 0.1;
% SVD
if bPCA
    [U, S, V] = mySVD(data);
    [U, S, V]=CutonRatio(U,S,V,options);
    eigvalue_PCA = full(diag(S));
    if bD
        data = U*S;
        eigvector_PCA = V;
        DPrime = data'*D*data;
        DPrime = max(DPrime,DPrime');
    else
        data = U;
        eigvector_PCA = V*spdiags(eigvalue_PCA.^-
1,0,length(eigvalue_PCA),length(eigvalue_PCA));
    end
```

```
else
    if ~bChol
        if bD
            DPrime = data'*D*data;
        else
            DPrime = data'*data;
        end
        switch lower(options.ReguType)
            case {lower('Ridge')}
                if options.ReguAlpha > 0
                    for i=1:size(DPrime,1)
                        DPrime(i,i) = DPrime(i,i) + options.ReguAlpha;
                    end
                end
            case {lower('Tensor')}
                if options.ReguAlpha > 0
                    DPrime = DPrime +
options.ReguAlpha*options.regularizerR;
                end
            case {lower('Custom')}
                if options.ReguAlpha > 0
                    DPrime = DPrime +
options.ReguAlpha*options.regularizerR;
                end
        end
        DPrime = max(DPrime,DPrime');
    end
end
WPrime = data'*W*data;
```

```
WPrime = max(WPrime,WPrime');
% Generalized Eigen
dimMatrix = size(WPrime,2);
if ReducedDim > dimMatrix
    ReducedDim = dimMatrix;
end
if isfield(options,'bEigs')
    bEigs = options.bEigs;
else
    if (dimMatrix > MAX_MATRIX_SIZE) && (ReducedDim <
dimMatrix*EIGVECTOR_RATIO)
        bEigs = 1;
    else
        bEigs = 0;
    end
end
if bEigs
    %disp('use eigs to speed up!');
    opt = struct('disp',0);
    if bPCA && ~bD
        [evector, evalue] = eigs(WPrime,ReducedDim,'la',opt);
    else
        if bChol
            option.cholB = 1;
            [evector, evalue] = eigs(WPrime,R,ReducedDim,'la',opt);
        else
            [evector, evalue] = eigs(WPrime,DPrime,ReducedDim,'la',opt);
        end
    end
```

```
    evalue = diag(evalue);
else
    if bPCA && ~bD
        [eigvector, eigvalue] = eig(WPrime);
    else
        [eigvector, eigvalue] = eig(WPrime,DPrime);
    end
    eigvalue = diag(eigvalue);

    [junk, index] = sort(-evalue);
    evalue = eigvalue(index);
    evector = eigvector(:,index);

    if ReducedDim < size(eigvector,2)
        evector = eigvector(:, 1:ReducedDim);
        evalue = eigvalue(1:ReducedDim);
    end
end
if bPCA
    evector = eigvector_PCA*evector;
end
for i = 1:size(evector,2)
    evector(:,i) = evector(:,i)./norm(evector(:,i));
end
%function CutonRatio
function [U, S, V]=CutonRatio(U,S,V,options)
    if  ~isfield(options, 'PCARatio')
        options.PCARatio = 1;
    end
```

```
eigvalue_PCA = full(diag(S));
if options.PCARatio > 1
    idx = options.PCARatio;
    if idx < length(eigvalue_PCA)
        U = U(:,1:idx);
        V = V(:,1:idx);
        S = S(1:idx,1:idx);
    end
elseif options.PCARatio < 1
    sEig = sum(eigvalue_PCA);
    sEig = sEig*options.PCARatio;
    sNow = 0;
    for idx = 1:length(eigvalue_PCA)
        sNow = sNow + eigvalue_PCA(idx);
        if sNow >= sumEig
            break;
        end
    end
    U = U(:,1:idx);
    V = V(:,1:idx);
    S = S(1:idx,1:idx);
end
```