第1章 绪论

1.1 特征提取的背景与意义

在进入互联网时代后，各种各样的终端设备呈现几何级增加，如智能手机、平板电脑、监控设备、支付设备等。在 2016 年，Gartner 预估 2018 年的物联网设备连接量可能达到 84 亿。然而到 2018 的时候，Statista 的统计结果显示，实际上的连接设备高达 230 亿。如此快速增长的终端设备意味着数据量的急速增加，同时数据的类别多种多样，如文本、音频、图像、视频等。所以如何处理这些大量且繁杂的数据成为了各领域的研究热点，而特征提取在各领域研究中都发挥着重要的作用。

首先是针对文本和音频数据的自然语言处理领域，机器翻译、语音识别、文本分类等在实际生活中得到普遍应用。从原始数据中抽取出量化的特征是自然语言处理中的重要一步，但是提取的特征通常存在着高维度和高冗余的问题，因此特征提取成为了解决该问题的关键方法。计算机视觉领域则主要针对图像和视频等数据，该领域的研究正被广泛地应用于各种场景。如刷脸打卡、刷脸支付、安防监控等人脸识别的场景，核磁共振图像、CT 图像等医学图像处理的场景。这些实际场景中的应用十分注重速度性能，而特征提取则在提升速度性能方面起着关键作用。

单从机器学习的理论角度来说，大多数实际场景的应用可以归结为数据分类这一学习任务。数据分类流程主要包括预处理、特征提取、分类模型训练、分类模型测试等。预处理负责将数据尺寸进行统一及训练集与测试集的分割等，特征提取的主要作用是将数据的维度进行降低，分类模型训练和分类模型测试则是将特征提取后的数据运用到相应的分类模型中进行训练和预测。而这其中的特征提取是本文的研究重点，因为该部分涉及到机器学习的一个普遍问题“维数灾难”。所谓的“维数灾难”是指在高维空间中呈现的数据样本稀疏、距离计算困难等问题。如果直接对预处理后的 64×64 维大的图片进行距离的计算肯定会消耗大量的计算能力，那么对通过降维操作获得的低维样本进行处理则能节省大量的时间。能进行降维操作的根本原因在于，研究人员发现在对数据进行学习时，其规律往往只与其中的部分信息存在密切的关系。降维操作主要包含两种方式，一种是特征选择，另一种是特征提取，前者主要是保留已有特征中的重要特征而删除其余特征，而后者则是从原本的特征中找到新的较少的特征组合。

如果说数论是数学的皇冠，那么特征提取也可以称为机器学习上的皇冠了。在实际应用中，它能让设备在获取高维数据的同时，还能快速地完成计算与处理，

这便是特征提取的实际价值。在理论研究上，提供了一种解决“维数灾难”的途径，并为研究者提出更加复杂的计算方式提供了支撑，这便是特征提取的理论价值。所以进行特征提取的方法研究是十分有必要的，因为好的特征提取方法不仅能促进实际应用的发展，也能促进理论研究的创新，达到“一箭双雕”的作用。

1.2 特征提取的研究现状

1.2.1 特征提取方法的研究现状

特征提取中最典型的方法就是多维尺度分析(Multiple Dimensional Scaling，MDS)法。该方法起源于 1958 年，而该方法的广泛应用应该归功于克鲁斯卡尔，因为克鲁斯卡尔于 1978 年对该方法进行了改进并出版成书。现阶段的机器学习中提及该方法，指的应该是 2001 年的考克斯版本。作为一种线性降维方法，其主要目标是保持数据降维前后的距离不变。

线性降维方法中最具影响力的是主成分分析(Principal Component Analysis，PCA)法，卡尔·皮尔逊于 1901 年提出了该方法，然后该方法迅速扩展到各门学科中变成了不同的表现形式，如：线性代数中散度矩阵的奇异值分解、统计学中的因子分析、信号处理中的离散 KL 变换、图像分析中的 Hotelling 变换等。该方法的主要目标是让降维后的数据尽可能地分开。

从机器学习的角度上来说，上述两种方法属于非监督式的学习方法。为了利用数据原本的类别信息，Fisher 最早于 1936 年在二分类问题上提出了线性判别分析(Linear Discriminant Analysis，LDA)法。该方法的目标十分的简单，即降维后同类数据尽可能的近而非同类数据尽可能的远。该方法巧妙地将类别信息融合到进行优化的目标中，是一个典型的监督式学习方法。

多数线性化的降维方法都存在一个假设，即从高维空间到低维空间的函数映射是线性的。但是实际上的许多映射都是非线性的，一般化的解决方法是使用“核化”这个技巧。舍尔科普夫在 1998 年将核技巧和主成分分析相结合提出了核主成分分析(Kernelized PCA，KPCA)法。核主成分分析的主要部分仍然是 PCA。但是为了克服原始数据不能通过线性映射到低维空间的假设问题，将不再直接对原始数据进行 PCA。而是先通过核函数将原始数据映射到更高维的空间中，再对更高维空间中的数据进行 PCA 降维。因为往往原本线性不可分的数据在提高到更高维的空间后就变得线性可分了。

国内的研究学者同样对特征提取方法做出了许多贡献，2004 年杨健团队提出了将 KPCA 和 LDA 结合的 KFDA 方法。其算法过程是先将原始数据使用 KPCA 算法进行降维，然后对 KPCA 算法降维后的数据使用 LDA 方法进行二次的特征提取。2012 年文颖团队将差异向量和 KPCA 结合提出了 DV-KPCA 的方法，其

过程是先对原始数据进行施密特正交化获取公共向量，然后将公共向量和原始数据对比获得差异向量，最后对得到的差异向量使用 KPCA 算法获取降维后的特征。

1.2.2 流形学习方法的研究现状

特征提取中的一个重要研究部分就是流形学习，流形学习作为一种降维方法主要利用了拓扑流形的相关性质。因为流形的每个局部空间与欧式空间同胚，所以可以设法将局部的性质推广到全局中。流形中典型的例子就是地球，其本质是一个扭曲在三维空间中的二维流形。对于地球的每个局部位置来说，都可以视为一个欧氏空间并利用其相关性质进行计算。

等度量映射(Isometric Mapping，Isomap)算法是最典型的流形学习方法，该算法由乔什·特南鲍姆于 2000 年提出。其基本出发点是直接计算高维空间中的直线距离是不准确的，因为高维空间中的直线距离在低维流形上是不存在的。比如从北京到纽约的距离肯定不可能是在三维空间中的直线距离，因为这个直线距离是没有实际意义的，所以其具有实际意义的最近距离就是沿着地球表面的测地线距离。对于如何计算测地线距离，乔什团队联想到了最短路径问题。首先可以确定的是流形中的局部距离可以使用欧式距离计算，那么每个点都可以使用欧式距离找出其近邻点，并将非近邻的点设为不可达的状态，这样就构成了一个近邻连接图，图中任意两点的测地距离就可以通过图的最短路径算法来进行计算了。最后其进行降维的方法仍然是 MDS 算法，只不过作为算法输入的距离矩阵是通过上述方法获取的。

Isomap 算法思想的本质和 MDS 算法相同，是保持数据降维前后的距离不变。而另一个典型的流形学习算法局部线性嵌入(Locally Linear Embedding，LLE)法则希望原本高维空间中的线性关系在低维空间中得以保持。具体来说是指高维空间中的一个数据样本可以通过相邻的几个样本线性组合而成，而当降维到低维空间后，对应数据的线性组合仍然存在。LLE 算法由罗维斯和索尔在 2000 年提出，并且 Isomap 和 LLE 两篇算法是一同发表在 2000 年的《科学》期刊上的。可见 2000 年对于流形学习来说是一个十分重要的年份。

国内的研究人员同样对流形学习的研究做出了贡献。2009 年 Li 和其团队提出了一个基于无参数判别分析(Nonparametric Discriminant Analysis，NDA)的人脸识别框架。2012 年 Yi 提出了一个基于线性回归的重构判别分析(Reconstructive Discriminant Analysis，RDA)法。这两个算法都是通过最大化局部的类内和类间的散度来获取最佳的子空间进行降维的。2017 年 Qu 等人提出了一种对最短路径进行改进的 Isomap 算法，同时 Zhen 等人提出了一个将流形与全连接层相结合的算法。

目前来看无论是传统的特征提取方法还是与流形学习相结合的特征提取方法都存在一些问题：

(1) 一些方法往往只考虑了数据的特征，而没有结合数据的类别信息。但这些类别信息对于机器学习来说往往十分重要。

(2) 单纯地只考虑整体的性质，而忽略了局部的性质。

(3) 对于距离的度量比较单一，未验证其他的距离度量方式。

1.3 本文的主要研究内容及创新点

本文研究的主要内容是数据分类中的特征提取方法，侧重点是作为其分支的流形学习方法。首先介绍了基本的数据分类的流程，说明了特征提取在流程中的位置和作用。然后介绍了几种常用的特征提取的方法，从而知晓了这些算法的优缺点。最后根据了解的优缺点进行算法的结合与改进，提出了两个创新点：

(1) LDA 具有利用数据类别信息的监督式方法的优点，却不能利用数据的局部信息。而 LLE 利用了数据的局部信息，却是一种非监督式的方法。所以针对两个方法的优缺点提出了一种利用流形边距的方法。首先是利用 LDA 的思想定义类间、类内和总体的概念，然后结合 LLE 算法定义三者的散度矩阵，最后根据类间、类内定义了流形边距的概念。通过求解最大化流形边距和最小化总体的线性表示误差的优化问题，得到用于数据降维的转换矩阵。

(2) 考虑到除了使用流形边距来衡量类别间的距离外，还可以使用对数欧式距离来度量类与类之间的距离。因此提出了一种基于几何感知距离的方法。首先是定义类别点和总体的概念，构建对应的近邻图并计算其散度矩阵。然后根据每个类别点的散度矩阵使用对数欧式距离计算各类之间的距离总和。最后仍然是一个最优化的问题，只不过优化目标变成了最大化类间对数距离总和与最小化总体的线性表示误差。

1.4 本文的组织结构

本文主要对机器学习中数据分类的关键步骤特征提取进行研究，并与流形学习相结合。文章的内容大致分为五个部分并按下列方式组织：

第一章，绪论。介绍了特征提取方法的研究背景与意义，在对传统特征提取的研究现状进行说明的同时，还针对特征提取中的流形学习分支进行了描述。然后概括了本文的主要研究内容与创新点，最后列出了全文的组织结构。

第二章，特征提取的相关介绍。阐述了特征提取相关的概念，描述了数据分类与特征提取的关系，展示了数据分类的一般化流程，介绍了常用的几种特征提取方法，并对这些方法进行了分类。

第三章，基于流形边距的特征提取方法。结合 LLE 方法与 LDA 方法提出了

类别与类别之间的流形边距概念，明确最大化流形边距和最小化线性表示误差的目标。最后进行算法的实现和实验结果的分析。

第四章，基于几何感知距离的特征提取方法。在利用数据类别信息的情况下，使用了更加合适的对数欧式距离来度量类别流形间的距离。因此其优化目标变成了最大化类别间对数欧式距离总和与最小化线性表示误差。最后对算法进行实现及相应实验结果的分析。

第五章，总结与展望。对本文的研究内容和创新点进行了总结，并对特征提取和流形学习进一步的研究方向进行了展望。

1.5 本章小结

本章首先介绍了特征提取的应用背景及作用，然后指出了特征提取的实际意义和理论意义。对于特征提取的研究现状主要从线性和非线性两个角度上来讲解，提及了多维尺度分析、主成分分析、线性判别分析、核主成分分析、核线性判别分析等方法。对于作为特征提取分支的流形学习的研究现状，则主要描述了等度量映射和局部线性嵌入两个经典的流形学习算法。另外国内的研究人员同样对特征提取和流形学习做出了巨大的贡献，如差异向量结合核主成分分析的算法、无参数判别分析和重构判别分析等。同时引出本文的两个创新点，流形边距和对数欧式距离度量。最后就是对文章的组织结构的说明。

第2章 特征提取的相关介绍

2.1 特征提取与数据分类关系

在机器学习中利用已知的数据进行模型训练并预测的过程被称为一个学习任务。若预测的结果为离散值，此类学习任务被称为数据分类；若预测的为连续值，则为数据回归。本文的实验属于人脸识别领域的范畴，是一个对人脸数据集中的人脸图像进行类别预测的过程。其预测结果显然是离散值，因此属于数据分类的学习任务。特征提取是机器学习中为了解决“维数灾难”问题的一种降维操作，而降维操作是数据分类流程中的一个步骤。所以两者之间的关系可以狭义的理解为数据分类包含特征提取，但是特征提取又可以应用到各种其他的问题上。因此为了更好的阐述两者的关系，分别对两者的相关概念做了具体的介绍。

2.1.1 特征提取相关概念

首先需要明确本文的特征提取是指用于降低数据维度的“特征提取”，而并非对图像、文本等原始数据进行数据量化的“特征提取”。对图像和文本等进行量化的特征提取操作更准确称呼应该是特征抽取，是一个将任意数据转换为可用于机器学习的数字特征的操作。另一个容易混淆的概念就是同样用于降低数据维度的特征选择。两者的相同点是最后产生的效果是一样的，即减少数据集中的特征数目。特征选择是将输入的特征集合，根据评价准则选择出一组具有良好分类能力的特征子集。特征提取是对输入的特征通过变化或者映射的方法而产生新的特征集合。一条 n 维的数据$\{x_1, x_2, \ldots, x_n\}$分别通过特征选择和特征提取处理的区别如图 2.1。

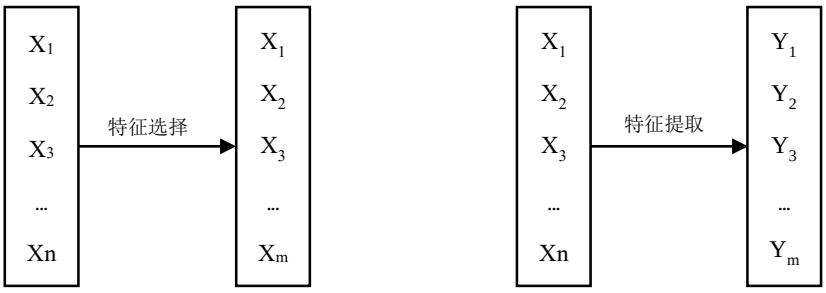

图2.1 特征选择和特征提取示意图

一般来说特征提取的过程是寻找一个变换矩阵 W 让原本高维空间中的数据

$X \in R^{D \times N}$ 转换到低维空间中成为数据 $Z \in R^{d \times N}$，其中 d 远小于 D，这样就达到了降低维度的目的。具体的表示如公式 2-1。

$$Z = W^T X \tag{2-1}$$

其中 W 属于 $R^{D \times d}$。

从矩阵分析的角度来说矩阵 W 包含了旋转变换和缩放变换，下面讨论不同的情况下的意义：

(1) W 为单位矩阵，则没有任何变换。

(2) W 为对角阵，则只进行缩放变换而不进行旋转的变换。

(3) W 为正交矩阵，则只进行旋转变换而不进行缩放变换。

(4) W 为方阵，则进行旋转和缩放两种变换，若此时的方阵不是满秩的，则具有降低维度的作用。

2.1.2 数据分类流程介绍

本小节介绍了机器学习中的数据分类流程，具体来说是本文对各种人脸数据集进行分类的实验流程。主要包括数据预处理、特征提取、训练分类模型、测试分类模型等步骤，如图 2.2。

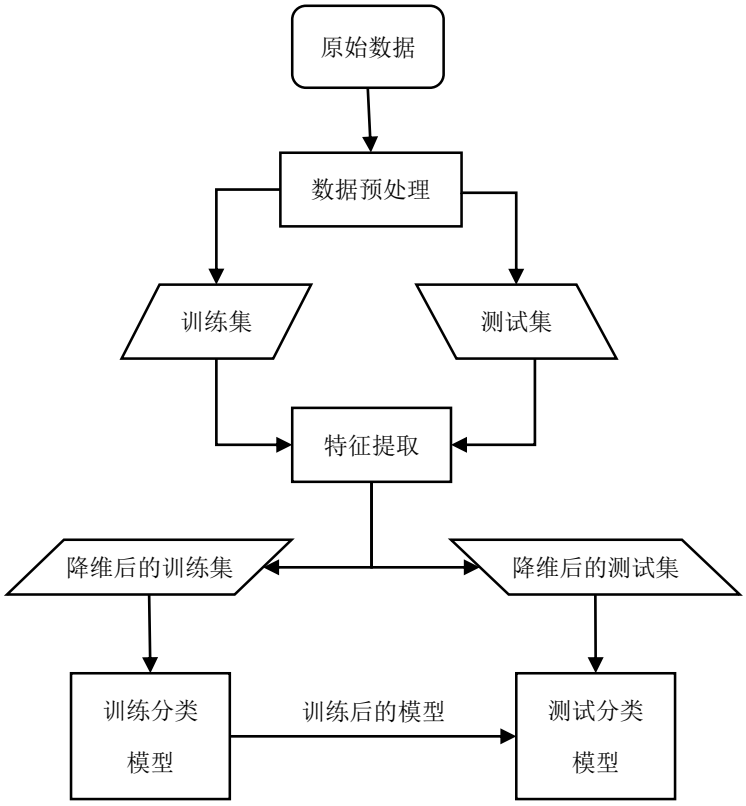

图2.2 数据分类流程图

数据预处理的过程主要包含两部分，一个是将图像数字化，另一个是将数据集分割。图像数字化首先需要将图片进行裁剪和缩放形成统一尺寸，然后将图片转换成灰度图的类型，最后将每张图片按行(列)优先转换为列向量并组合在一起成为数据矩阵。数据集分割是指将数据的每个类别按固定比例分割成为训练集和测试集两部分，如 100(10×10，10 种类别，每类 10 条)条数据按 40%训练集 60%测试集的比例进行分割，那么训练集是 40(10×4，10 种类别，每类 4 条)条数据，测试集是 60(10×6，10 种类别，每类 6 条)条数据。最后如果有必要可能需要对数据进行标准化的处理。

特征提取的过程按主次也分为两个部分。主要的部分是利用预处理分割的训练数据集，结合相应的特征提取算法得到公式 2-1 中的 W 矩阵和降维后的训练数据集。其中的特征提取算法就是本文主要研究的重点，除了后续章节会介绍的常用的算法外，还有本文提出的两个新算法。次要的部分则是使用获取的 W 矩阵根据公式 2-1 对测试数据集进行降维，从而得到降维后的测试数据集。

分类的过程则分为训练和测试两个部分，首先本文采用的是 k 近邻(k-Nearest Neighbor，kNN)分类方法。训练的部分是利用降维后的训练数据集对建立的分类模型进行训练，测试部分则是利用训练后的分类模型对测试数据集进行预测并进行相关评价。

k 近邻分类方法是一种常用的监督式分类方法，其工作原理简单来说就是“近朱者赤，近墨者黑”。给定需要预测的测试样本，根据某种距离度量计算该样本和训练数据集的距离，选择其中最小的 k 个数据对测试样本进行预测。如果是分类任务则使用“投票法”，回归任务使用“平均法”。因为 k 个近邻的距离不同，所以可以对投票法和平均法进行加权优化。k 值的确定十分重要，因为不同的 k 值可能导致不同的结果，图 2.3 是一个 k 近邻分类的示意图。

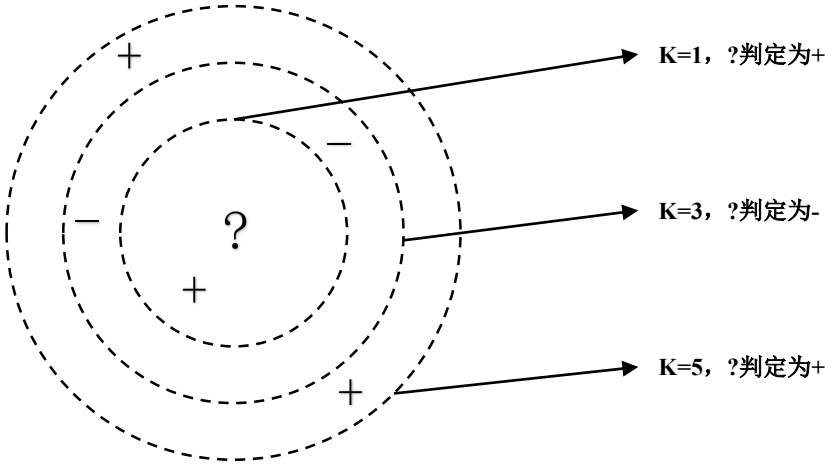

图2. 3 k 近邻分类示意图

2.2 特征提取与流形学习关系

特征提取是降维的两种方式之一，而流形学习是一种利用拓扑流形性质的从原始特征中生成新特征的降维方法，因此可以认为流形学习是特征提取的一个研究分支。上一节中已经说明了特征提取的概念，现在为了更好的说明两者的关系，需要对流形的概念进行简单的介绍。首先流形是指一个局部与欧式空间同胚的空间，也就是它的每个局部空间可以近视为欧式空间并利用相关性质，如使用欧式距离进行度量。这一特性就可以应用到降维中，当低维流形嵌入到高维空间中时，即使样本在高维空间的分布复杂，但是在局部上仍满足欧式空间的性质。这样可以先建立局部的映射关系，然后再将局部映射关系推广到全局。

高维空间的数据实际上是存在冗余的，并不需要那么多的维度来进行表示，可能只需要少量的维度就可以进行表示，那么这个低维的表示数据的空间就可以称为流形。反过来说就是，将这个低维的流形进行扭曲后就嵌入到高维空间中了，图 2.4 形象的展示了三维空间的数据展开为一个二维流形的过程。球体就是一

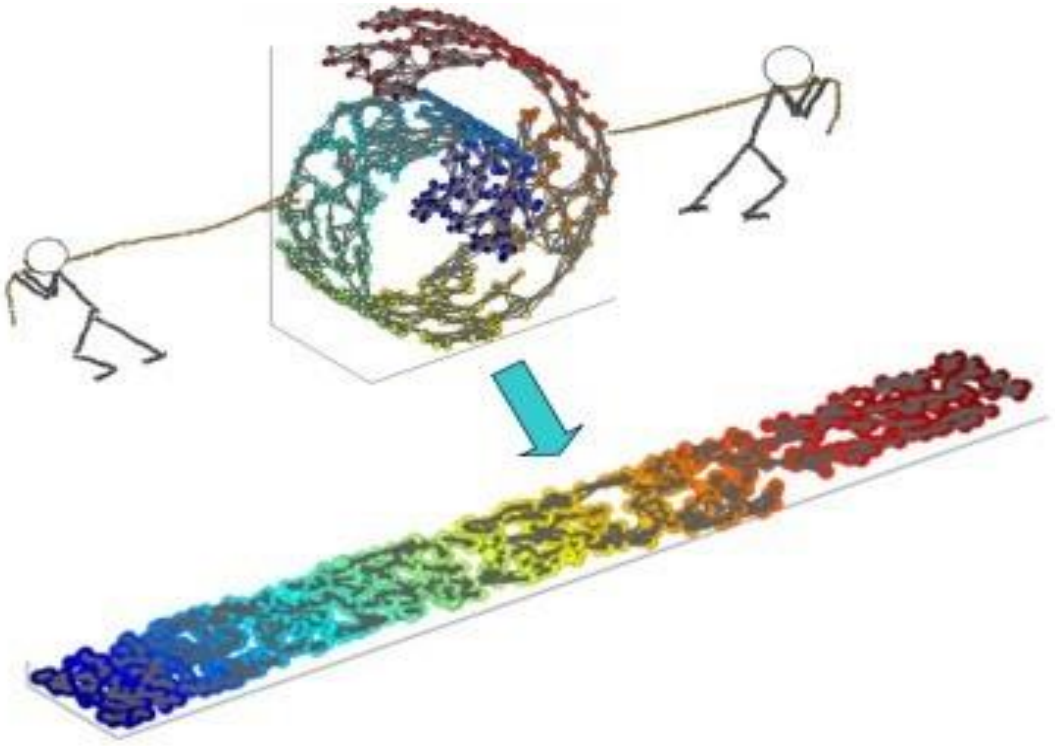

图2.4 流形展开示意图

个典型的例子，可以理解为一个二维的流形经过扭曲后形成了嵌入在三维空间中的球。球可以用公式 2-2 来表示，球上的每一个点都可以用一个三元组$[x, y, z]$来表示，但是实际上这些三元组可以只用两个变量 θ 和 φ 来表示，这也证明了存在一个二维的流形。

$$\begin{cases} x = x_0 + r\sin\theta\cos\varphi \\ y = y_0 + r\sin\theta\sin\varphi \\ z = z_0 + r\cos\varphi \end{cases} \tag{2-2}$$

2.3 几种常用特征提取的算法

特征提取的算法有许多种，根据不同的标准会形成不同的分类结果。从是否使用数据类别信息的角度可以分为有监督方法和无监督方法；从变换形式的角度可以分为线性方法和非线性方法；从数据结构类别的角度可以分为全局方法和局部方法。本文主要从线性和非线性的分类来介绍了几种常用的特征提取算法。

2.3.1 线性的特征提取算法

MDS 算法是一个经典的线性特征提取算法，其主要思想是保持原本数据集样本间的距离在降维后的低维空间中尽量不变。首先确定降维前的数据$X = \{x_1, x_2, \ldots, x_m\}$且$X \in R^{d\times m}$，目标的结果数据是降维后的数据$Z = \{z_1, z, \ldots, z_m\}$且$Z \in R^{d'\times m}$。为了求解矩阵 Z，可以令$B = Z^T Z \in R^{m\times m}$，对矩阵 B 进行特征分解有$B = V\Lambda V^T$,其中$\Lambda$为 d 个特征向量由大到小排列形成的对角矩阵，V是对应的特征向量构成的特征矩阵。可以选取其中的前d'个特征值构成新的对角矩阵$\tilde{\Lambda}$和特征矩阵$\tilde{V}$，那么$B = \tilde{V}\tilde{\Lambda}\tilde{V}^T = \left(\tilde{\Lambda}^{1/2}\tilde{V}^T\right)^T\left(\tilde{\Lambda}^{1/2}\tilde{V}^T\right)$，因为对角矩阵的转置不变。

$$Z = \tilde{\Lambda}^{1/2}\tilde{V}^T \in R^{d'\times m} \tag{2-3}$$

为了计算矩阵 B，需要根据原始数据 X 计算距离矩阵$D \in R^{m\times m}$,其中的第 i 行 j 列元素d_{ij}表示 x_i 和 x_j 之间的距离。由于降维前后的距离不变，故有$||z_i - z_j|| = d_{ij}$,且$b_{ij} = {z_i}^T z_j$。

$$d_{ij}^2 = ||z_i||^2 + ||z_j||^2 - 2{z_i}^T z_j = b_{ii} + b_{jj} - 2b_{ij} \tag{2-4}$$

由于会对数据进行中心化的处理，因此可以得到如下结果:

$$\sum_{i=1}^{m} d_{ij}^2 = tr(B) + mb_{jj} \tag{2-5}$$

$$\sum_{j=1}^{m} d_j^2 = tr(B) + mb_{ii} \tag{2-6}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}^2 = 2m\ \ tr(B) \tag{2-7}$$

同时规定三个值$d_{i\cdot}^2$、$d_{\cdot j}^2$、$d_{\cdot\cdot}^2$，其表示如下：

$$d_{i\cdot}^2 = \frac{1}{m}\sum_{j=1}^{m} d_{ij}^2 = \frac{1}{m}tr(B) + b_{ii} \tag{2-8}$$

$$d_{\cdot j}^2 = \frac{1}{m}\sum_{i=1}^{m} d_{ij}^2 = \frac{1}{m}tr(B) + b_{jj} \tag{2-9}$$

$$d_{\cdot\cdot}^2 = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}^2 = \frac{2}{m}tr(B) \tag{2-10}$$

结合 2-4、2-8、2-9、2-10 可以得到b_{ij}的计算公式 2-11。最后 MDS 的算法步骤如表 2.1。

$$b_{ij} = -\frac{1}{2}(d_{ij}^2 - d_{i\cdot}^2 - d_{\cdot j}^2 + d_{\cdot\cdot}^2) \tag{2-11}$$

表2. 1 MDS 算法步骤

输入：距离矩阵 D,其元素为原始数据 x_i 和 x_j 之间的距离；降维后的维数d'。

流程：

1. 根据公式 2-8 到 2-10 计算定义的$d_{i\cdot}^2$、$d_{\cdot j}^2$、$d_{\cdot\cdot}^2$；

2. 根据公式 2-11 计算矩阵 B；

3. 对矩阵 B 进行特征分解；

4. 选取前d'大的特征值构成对角矩阵$\tilde{\Lambda}$和特征向量$\tilde{V}$，根据公式 2-3 得到 Z；

输出：降维后的矩阵 Z。

PCA 算法则是另一个常用的线性特征提取方法。其目标有两个，一个是最近重构性，另一个是最大可分性,两者的思想都与“超平面”这个概念相关。为此需要确定“超平面”的概念，超平面是指进行降维后的低维空间的基向量构成的一个“平面”。二维空间中是一条直线，三维空间中是一个平面，更高维的空间中则称之为“超平面”。最近重构性是指原始数据到这个超平面的距离尽可能近。最大可分性是指降维后数据尽可能地分开，即原始数据投影到该超平面后的点尽可能地分开。因为这两个目标实际上是一个等价的推导，所以这里仅从最大可分性来进行介绍。首先原始的数据为$X \in R^{d\times m}$,包含 m 条 d 维的数据。所以一条数

据x_i通过转换矩阵投影后的数据为$W^T x_i$，那么为了达到最大可分性的目的，需要投影后的数据的方差最大，图 2.5 进行了解释。

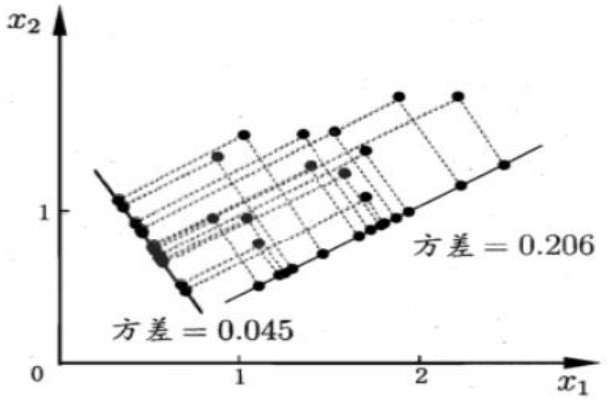

图2.5 PCA 最大可分性示意图

由于数据一般都会进行中心化的处理，那么方差为$\sum_{i=1}^{m} W^T x_i x_i^T W$。进行矩阵化的表示后得到优化目标如公式 2-12:

$$\begin{cases} \max\limits_{W} \ \mathrm{tr}(W^T X X^T W) \\ \text{s.t.} \quad W^T W = I \end{cases} \tag{2-12}$$

解决最优化问题的常用方法是拉格朗日乘子法，对公式 2-12 进行拉格朗日乘子法得到公式 2-13:

$$XX^T w = \lambda w \tag{2-13}$$

现在只需要对XX^T进行特征值分解，得到 d 个特征值后进行排序，最后选取前d'大特征值对应的特征向量构成投影矩阵W'。具体的算法步骤如表 2.2。

表2.2 PCA 算法步骤

输入： 原始数据$X = \{x_1, x_2, \ldots, x_m\}$；低维空间的维数$d'$。

流程：

1. 对原始数据进行中心化处理；

2. 计算协方差矩阵XX^T；

3. 对协方差矩阵XX^T进行特征分解；

4. 选取前d'大特征值对应的特征向量构成投影矩阵W';

输出： 投影矩阵W'。

最后介绍的特征提取算法是 LDA 算法，该算法是一个监督式的方法，因为该方法使用了数据的类别信息。该算法的优化目标结合类别信息来描述是让降维后的同类点尽可能地近而非同类之间尽可能地远。图 2-6 展示了使用不同的投影矩阵将二维的数据投影到直线上的结果，其中左边的图投影后的数据中相同类别的点分布比较散并且不同类别的交界处还存在部分混杂，而右边的图显示的效果则是相同的点分布比较集中并且不同类别间隔明显。那么肯定是右边的图的效果更符合 LDA 算法的目标，并且降维后的数据能更好的用于分类。

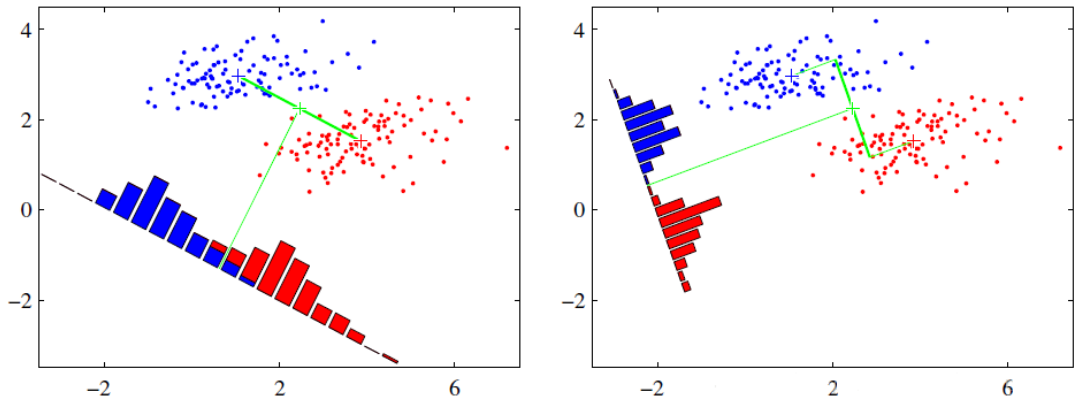

图2. 6 LDA 效果示意图

因为 LDA 结合了数据的类别信息，为了更好的进行说明先从简单的二分类问题进行切入，最后再推广到多分类的问题。原始的带有类别信息的数据集X = $\{(x_i, y_i)\}_{i=1}^{m}$,其中x_i表示第 i 条数据的 n 维特征，而$y_i \in \{0,1\}$表示第 i 条数据的类别。用X_c，μ_c，E_c分别表示类别c $\in \{0,1\}$的特征向量集合、平均值向量和协方差矩阵。算法目标的第一部分是降维后非同类的点尽可能地远，其含义就是投影后的类别的中心点$w^T\mu_c$的距离最大化如公式 2-14，其中 w 表示投影矩阵。算法目标的第二部分是降维后同类点尽可能地近，其含义就是投影后的类别的协方差$w^T E_c w$的总和最小化如公式 2-15。

$$\max ||w^T\mu_0 - w^T\mu_1||_2^2 \tag{2-14}$$

$$\min(w^T E_0 w + w^T E_0 w) \tag{2-15}$$

将两者结合统一进行表示如公式 2-16。其中定义类内散度矩阵$S_w = E_0 + E_1$，类间散度矩阵$S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T$,因此 J(w)可以使用散度矩阵表示为$\frac{w^T S_b w}{w^T S_w w}$。

$$\arg\max_w \mathrm{J}(w) = \frac{\left||w^T\mu_0 - w^T\mu_1|\right|_2^2}{w^T E_0 w + w^T E_0 w} = \frac{w^T(\mu_0-\mu_1)(\mu_0-\mu_1)^T w}{w^T(E_0+E_1)w} \tag{2-16}$$

另外为了更好的进行统一，会对投影矩阵进行一定的限制即$w^T\mathrm{w}=1$,求解这个最优化问题的方法同样是拉格朗日乘子法，最后可以得到公式 2-17。该公式的求解和 PCA 方法一样变成了特征值求解的问题。

$$S_w^{-1}S_b w=\lambda w \tag{2-17}$$

在应用到多分类的问题时，需要注意类内散度矩阵S_w和类间散度矩阵S_b的计算，其具体定义如公式 2-18 和 2-19。其中 N 表示类别个数，m_i表示第 i 类别的样本数目，μ表示总的均值向量。

$$S_w=\sum_{i=1}^{N}\sum_{x\in X_i}(x-\mu_i)(x-\mu_i)^T \tag{2-18}$$

$$S_b=\sum_{i=1}^{N}m_i(\mu_i-\mu)(\mu_i-\mu)^T \tag{2-19}$$

具体的 LDA 算法描述如表 2.3。

表2. 3 LDA 算法步骤

输入：原始数据$\mathrm{X}=\{(x_i,y_i)\}_i^m$；$y_i\in\{1,2,\dots,N\}$；低维空间的维数$d'$。

流程：

1. 根据公式 2-18 计算类内散度矩阵S_w；

2. 根据公式 2-19 计算类间散度矩阵S_b；

3. 计算矩阵$S_w^{-1}S_b$，并进行特征值分解，得到特征值λ及对应的特征向量 w；

4. 选取前d'大的特征值对应的特征向量构成投影矩阵W';

5. 计算投影后的数据$z_i=W'^T x_i$。;

输出：投影后的数据$\mathrm{Z}=\{(z_i,y_i)\}_i^m$。

2. 3. 2 非线性的特征提取算法

对于非线性的问题最常用的方法就是核技巧，将核化的技巧与常用的线性特征提取方法 PCA 相结合就形成了常用的非线性特征提取方法 KPCA 算法。其主要思想是将原本线性不可分的数据映射到高维空间后，由于其在高维空间中是线性可分的便可以采用线性的 PCA 算法进行降维。首先确定原始数据为$\mathrm{X}=\{x_1,x_2,\dots,x_m\}\in R^{d\times m}$,映射的高维空间的数据为$\mathrm{Z}=\{z_1,z_2,\dots,z_m\}\in R^{D\times m}$。对于高维空间中的数据 Z 结合 PCA 算法中的公式 2-15 可得$\mathrm{Z}Z^T w=\lambda w$,假设通过映

射Φ将数据 X 转换到 Z 即$\mathrm{Z}=\Phi(\mathrm{X})$，结合两者可得$\Phi(\mathrm{X})\Phi(X)^{T}w=\lambda w$。可以令 w 表示为$\mathrm{w}=\mathrm{Z}\alpha=\Phi(\mathrm{X})\alpha$其中$\alpha=\frac{Z^{T}w}{\lambda}$,最后将$\Phi(X)^{T}\Phi(\mathrm{X})\Phi(X)^{T}\Phi(\mathrm{X})\alpha=\lambda\Phi(X)^{T}\Phi(\mathrm{X})\alpha$进行化简可得$\Phi(X)^{T}\Phi(\mathrm{X})\alpha=\lambda\alpha$。之所以化简为这种形式是由于核技巧的原因，因为尽管映射Φ无法进行确定，但是其内积$\Phi(X)^{T}\Phi(\mathrm{X})$可以通过核函数k(X)进行计算，从而得到核矩阵 K，最终进行特征分解的形式如公式 2-20。

$$\mathrm{K}\alpha=\lambda\alpha \tag{2-20}$$

核函数的主要作用是不用直接计算到高维的映射，而是使用函数值表示内积的值，这样能避免大量的计算。核函数的充分条件是 mercer 定理，mercer 定理是指任何半正定的函数都可以作为核函数。所谓的半正定函数是指，如果$\mathrm{f}(x_i,x_j)$构成的矩阵是半正定的矩阵，那么函数 f 就是核函数。常用的核函数包括线性核函数、多项式核函数、高斯核函数等，具体形式如下:

$$\mathrm{k}(\mathrm{x},\mathrm{y})=x^{T}y+c \tag{2-21}$$

$$\mathrm{k}(\mathrm{x},\mathrm{y})=(\alpha x^{T}y+c)^{d} \tag{2-22}$$

$$\mathrm{k}(\mathrm{x},\mathrm{y})=\exp(-\frac{||x-y||^{2}}{2\sigma^{2}}) \tag{2-23}$$

最后 KPCA 算法的具体流程如表 2.4。

表2. 4 KPCA 算法步骤

输入：原始数据$\mathrm{X}=\{x_1,x_2,\dots,x_m\}\in R^{d\times m}$；低维空间的维数$d'$。

流程：

1. 选取核函数 k；

2. 利用核函数$\mathrm{k}(x_i,x_j)$计算核矩阵 K；

3. 结合公式 2-20 进行特征值分解；

4. 选取前d'大的特征值对应的特征向量构成投影矩阵α;

5. 计算投影后的数据$Z=W^{T}\Phi(\mathrm{X})=\alpha^{T}\Phi(\mathrm{X})^{T}\Phi(\mathrm{X})=\alpha^{T}K$；

输出：投影后的数据Z。

Isomap 算法利用流形学习中的局部同胚的特性，结合线性方法中的 MDS 算法进行降维。其主要的关键点在于样本点之间的距离不是使用传统的 MDS 的欧式距离进行表示，而是使用构建的图的最短路径进行表示。其流程如表 2.5：

表2.5 Isomap 算法步骤

输入：原始数据X $= \{x_1, x_2, \ldots, x_m\} \in R^{d\times m}$；近邻点数 k；低维空间的维数$d'$。

流程：

1. 计算点x_i与其他点的距离，排序后选取前 k 小的点作为近邻点；

2. 重复步骤 1 确定 m 个点的近邻并设置其欧式距离，非近邻点的距离设置为无穷大；

3. 结合前两步构建的图，利用最短路径算法计算任意两点之间的距离 $\mathrm{dist}(x_i, x_j)$，生成距离矩阵 Dist；

4. 将计算的距离矩阵 Dist 作为 MDS 算法的输入;

5. 返回 MDS 算法的输出；

输出：降维后的数据 Z。

另一个基于流形学习的 LLE 方法认为高维数据的局部特性在降维后同样存在并保持相同。具体来说就是对于样本点x_i,其可以用相邻的局部点x_j, x_k, x_l进行线性表示如公式 2-24，参数 w 就是所谓的局部特性，其保持相同则是指降维后

$$x_i = w_{ij}x_j + w_{ik}x_k + w_{il}x_l \tag{2-24}$$

对应的点仍然使用相同的参数 w 重构。对于参数 w 的确定，则是点x_i与重构后点间距离的最小化问题，其具体表示如公式 2-25，其中Q_i表示样本点x_i用于重构的近邻点的下标集合。若令$C_{jk} = \left(x_i - x_j\right)^T (x_i - x_k)$,则公式 2-25 求解的参数 w

$$\begin{cases} \min\limits_{w_1, w_2, \ldots w_m} \sum_{i=1}^{m} ||x_i - \sum_{j\in Q_i} w_{ij}x_j||_2^2 \\ s.t. \quad \sum_{j\in Q_i} w_{ij} = 1 \end{cases} \tag{2-25}$$

如公式 2-26。假设降维后的点用Z $= \{z_1, z_2, \ldots z_m\} \in R^{d'\times m}$表示，因为进行重构的

$$w_{ij} = \frac{\sum_{k\in Q_i} C_{jk}^{-1}}{\sum_{l,s\in Q_i} C_{ls}^{-1}} \tag{2-26}$$

参数 w 保持不变就有公式 2-27，其与公式 2-25 形式相同但是求解的目标则不相

$$\min_{z_1,z_2,\dots z_m} \sum_{i=1}^{m} ||z_i - \sum_{j\in Q_i} w_{ij} z_j||_2^2 \tag{2-27}$$

同,公式 2-25 中 x 已知求解 w 而公式 2-27 中 w 已知求解 z。

公式 2-27 的矩阵化表示如公式 2-28，其中矩阵 M 为公式 2-29。

$$\begin{cases} \min\limits_{Z} tr(ZMZ^T) \\ s.t. \quad ZZ^T = I \end{cases} \tag{2-28}$$

$$\mathrm{M} = (I - W)^T (I - W) \tag{2-29}$$

对于公式 2-28 的求解同样可以使用特征值分解，将矩阵 M 进行特征值分解后，前d'个特征值对应的特征向量构成的矩阵即为Z^T。LLE 算法的具体步骤如下表 2.6。

表2. 6 LLE 算法步骤

输入：原始数据$\mathrm{X} = \{x_1, x_2, \dots, x_m\} \in R^{d\times m}$；近邻点数 k；低维空间的维数$d'$。

流程：

1. 计算点x_i与其他点的距离，排序后选取前 k 小的点作为近邻点；

2. 利用公式 2-25 求解w_{ij},对于非近邻的点 w 则为 0；

3. 利用公式 2-29 得到矩阵 M；

4. 对矩阵 M 进行特征值分解;

5. 选取前d'个特征值对应的特征向量；

输出：降维后的数据 Z。

2. 4 本章小结

本章介绍的主要内容是降维方法中的特征提取。首先介绍特征提取和数据分类之间的关系，因此阐述了特征提取的概念和数据分类的流程；然后说明了流形学习相关的概念并表明了流形学习和特征提取之间的关系；最后详细介绍了几种常用的特征提取的算法，并按线性和非线性进行了分类。线性的特征提取的方法包括保持距离不变的 MDS 算法、保持最大可分性的 PCA 算法、利用类别信息的 LDA 算法。非线性的特征提取的方法则主要介绍了三种，核技巧和 PCA 结合的 KPCA 算法、利用最短路径与 MDS 相结合的 Isomap 算法、保持局部特性不变的 LLE 算法。

第3章 基于流形边距的特征提取方法

3.1 引言

近些年来，提出了许多基于 LLE 的监督式算法用于处理数据分类问题。其中最普遍的方法是将监督式的 LDA 算法与 LLE 算法相结合，另外一部分则是使用数据的类别信息构建数据局部近邻图。但是在构建 k 近邻图的时候，会存在非同类点之间的距离比同类点之间的距离更短的情况，这种错误的情况会导致在进行判别分析时错误地选择近邻点。解决这个问题的方法主要有两种，一种是对数据点之间的距离进行调整，另一种是选取近邻点时只从同类点中进行选取。假设位于一个流形上的点都是来自相同的类型，那么相应的一些其他的流形上的点也是相关类别点构建而成的，这样能使用构建的 k 近邻图表征流形内部的数据。但是这样会忽略了由流形之间的数据构成的 k 近邻图，即没有使用类别与类别之间的关系。基于以上描述可以构建流形内图和流形间图，并且结合这两个图提出的流形边距度量可以全局地衡量不同流形间的距离。最后可以提出一个不考虑流形类别信息的全局流形图用于衡量数据的局部性。相较于非监督式的 LLE 算法而言，本章提出的局部线性表示流形边距(LLRMM)算法结合了流形类别信息构建的流形间图和流形内图。在流形间图中，任何结点及其 k 近邻必须属于不同的流形，因此流形间图表示不同流形间的距离。在流形内图中，任何结点及其 k 近邻点则属于相同的流形。构建的两个图与定义的全局流形边距存在密切的关系。

为了更好的说明提出的 LLRMM 算法，首先从简单的二分类问题着手。在算法示意图 3.1 中，存在两个不同类型的流形 M1 和 M2。对于流形 M1 中的一个

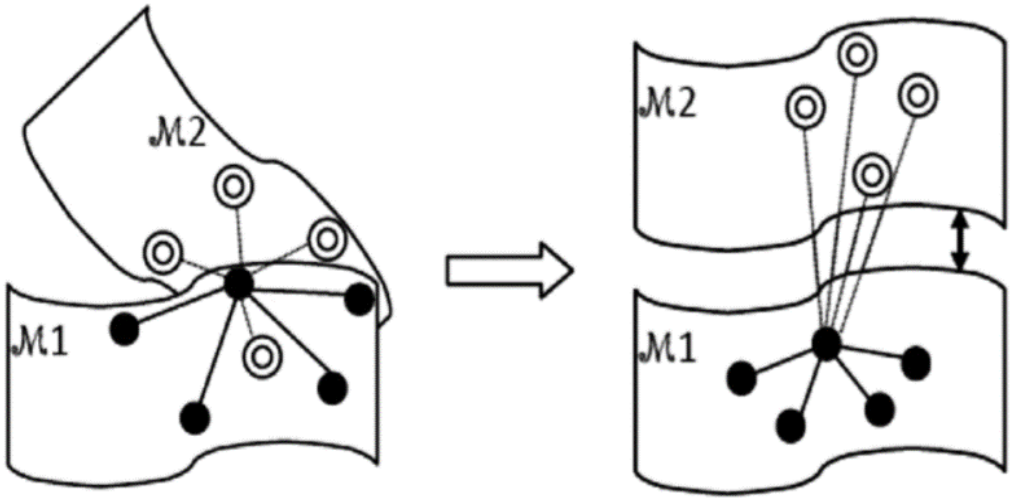

图3.1 LLRMM 算法示意图

样本点，其构建流形内图的其余四个最近邻点选自于同类别的 M1 流形中，与此同时，可以选择不同类别的 M2 流形中的距离该点最近的四个点一同构成流形间图。从图 3.1 的左边图可以发现，两个流形 M1、M2 的数据在高维空间中互相混合无法很好的进行分离，因此为了对两个流形的数据进行分离，需要找到一个低维的子空间达到最大化流形间的边距的目的，正如右边的图展示的情况。

3.2 基于最大流形边距的算法

本节主要对 LLRMM 算法从三个方面进行了详细的描述。首先是局部线性表示的阐述，因为该算法是一个基于流形学习的算法，而对于流形的局部同胚性这一性质而言，其具体化的表现就是局部线性表示；第二点则是本算法的重点，利用流形内图和流形间图来构建流形边距的概念；最后则是借鉴 LDA 算法的思想，将流形边距和全局线性表示结合构成瑞利熵的形式进行最优化问题的求解。

3.2.1 局部线性表示

首先是流形内图的构建，构建的过程中会涉及流形的类别信息和局部几何性质。对于类别信息来说是指图中任何一个点和它的最近邻点都是来自同一个流形即是同一个类别。局部几何性质是说该点可以由它的最近邻点组合而成，换句话说就是能用近邻点线性表示该点，更进一步就是求解表示误差最小化的优化问题并获取表示权值 w。该优化问题的目标函数就是表示误差如公式 3-1，其中$X_j(j \in \{1,2,\dots k\})$表示$X_i$的 k 个同类流形的近邻点。由于线性表示的权值有单位化的统一限制即$\sum_{j=1}^{k}(W_w)_{ij}=1$，所以公式 3-1 可以重写为公式 3-2。

$$\varepsilon((W_w)_i)=\min_{W_w}||X_i-\sum_{j=1}^{k}(W_w)_{ij}X_j||^2 \tag{3-1}$$

$$\begin{aligned}\varepsilon((W_w)_i)&=\min_{W_w}\left\|\sum_{j=1}^{k}(W_w)_{ij}X_i-\sum_{j=1}^{k}(W_w)_{ij}X_j\right\|^2\\&=\min_{W_w}\left\|\sum_{j=1}^{k}(W_w)_{ij}(X_i-X_j)\right\|^2\end{aligned} \tag{3-2}$$

这里几乎和 LLE 一致，可以令$G_{jt}=(X_i-X_j)(X_i-X_t)$,则公式 3-2 可以重新表示为公式 3-3。并且可以利用拉格朗日乘子法进行最优化问题的求解获取线性

$$\varepsilon((W_w)_i)=\min_{W_w}\left\{\sum_{j=1}^{k}(W_w)_{ij}(X_i-X_j)\cdot\sum_{t=1}^{k}(W_w)_{it}(X_i-X_t)\right\}$$

$$= \min_{W_w} \sum_{j=1}^{k} \sum_{t=1}^{k} (W_w)_{ij} (W_w)_{it} G_{jt} \tag{3-3}$$

表示的权值W_w。进行拉格朗日求解的过程主要分为两步，第一步是拉格朗日函数的构建如公式 3-4，第二步则是对求解参数计算偏导，并且偏导数为零的点为

$$\mathrm{L} = \sum_{j=1}^{k} \sum_{t=1}^{k} (W_w)_{ij} (W_w)_{it} G_{jt} - \lambda(\sum_{j=1}^{k} (W_w)_{ij} - 1) \tag{3-4}$$

极值点，即是要求解的值。公式 3-4 中对W_w计算偏导，并且偏导数等于零即$\frac{\delta L}{\delta W_w} = 0$,最后解的结果如公式 3-5。

$$(W_w)_i = \frac{\sum_{t=1}^{k} G_{jt}^{-1}}{\sum_{m=1}^{k} \sum_{l=1}^{k} G_{lm}^{-1}} \tag{3-5}$$

公式 3-5 得到的是X_i在流形内图中 k 个近邻点的权值，相应的非近邻点就不占比重其权值为 0，其具体表示如公式 3-6。其中$\mathrm{WithinN}(X_i)$表示X_i在流形内图中近邻点的集合。这样流形内图中的任何一个点都存在一个相对应的权值向量，最后所有数据点的向量组合在一起即构成了流形内图的权值矩阵W_w。

$$(W_w)_i = \begin{cases} \frac{\sum_{t=1}^{k} G_{jt}^{-1}}{\sum_{m=1}^{k} \sum_{l=1}^{k} G_{lm}^{-1}} & X_j \in WithinN(X_i) \\ 0 & otherwise \end{cases} \tag{3-6}$$

同样的，可以对流形间图使用与流形内图相同的流程获取权值矩阵W_b,其不同点在于近邻点的不同，其近邻点是与该点不同流形的点。具体的每个点的权值表示如公式 3-7，其中$\mathrm{BetweenN}(X_i)$表示X_i在流形间图中近邻点的集合。

$$(W_b)_i = \begin{cases} \frac{\sum_{t=1}^{k} G_{jt}^{-1}}{\sum_{m=1}^{k} \sum_{l=1}^{k} G_{lm}^{-1}} & X_j \in \mathrm{BetweenN}(X_i) \\ 0 & otherwise \end{cases} \tag{3-7}$$

最后会求解一个全局流形图的权值矩阵W_t,其流程仍然没有改变，只是近邻点的选取与类别没有关系而是从所有的数据中进行选择。具体的权值表示如公式 3-8，其中$\mathrm{TotalN}(X_i)$,表示X_i在全局的数据中的近邻点集合。

$$(W_t)_i = \begin{cases} \frac{\sum_{t=1}^{k} G_{jt}^{-1}}{\sum_{m=1}^{k} \sum_{l=1}^{k} G_{lm}^{-1}} & X_j \in \mathrm{TotalN}(X_i) \\ 0 & otherwise \end{cases} \tag{3-8}$$

3.2.2 流形边距定义

在上一小节中定义了三个流形图，分别是流形内图、流形间图和流形总图，并且还确定了图中点用局部线性表示的权值及构成的权值矩阵。通过这些权值矩阵能很好的表示流形内数据、流形间数据与总体数据的特征。本小节基于三种流形图定义了对应的三种散度矩阵，分别是流形内图散度矩阵、流形间图散度矩阵和流形总图散度矩阵。

在公式 3-1 中，是通过最小化一个点和其同流形近邻点的线性表示来获取权值的。通过二次型可以将公式 3-1 重新表示为 3-9，其中$U_w = (\mathrm{I} - W_w)(\mathrm{I} - W_w)^T$。

$$\begin{aligned}\varepsilon((W_w)_i) &= \min_{W_w} \left\| \sum_{j=1}^{k} (W_w)_{ij} X_i - \sum_{j=1}^{k} (W_w)_{ij} X_j \right\|^2 \\ &= \min_{W_w} tr\left\{ (X_i - \sum_{j=1}^{k} (W_w)_{ij} X_j)(X_i - \sum_{j=1}^{k} (W_w)_{ij} X_j)^T \right\} \\ &= \min_{W_w} tr\left\{ \sum_{i,j} (U_w)_{ij} (X_i X_j) \right\} \end{aligned} \tag{3-9}$$

如果将该公式应用到所有的数据点，则可以用矩阵的形式进行表示如公式 3-10。

$$\varepsilon(W_w) = \min_{W_w} tr(X U_w X^T) \tag{3-10}$$

公式 3-10 的目标是通过求解受约束的最优化问题来找到最小化表示误差的权值，因此该公式可以被用来表示流形内图数据的紧密度，并可以定义对应的流形内图散度矩阵S_w，其具体形式如公式 3-11。最后与流形内图定义的散度矩阵相类似地，可以分别定义流形间图散度矩阵S_b和流形总图散度矩阵S_t,其对应的具体形式如公式 3-12 和 3-13，其中$U_b = (\mathrm{I} - W_b)(\mathrm{I} - W_b)^T$、$U_t = (\mathrm{I} - W_t)(\mathrm{I} - W_t)^T$。

$$S_w = \mathrm{X} U_w X^T \tag{3-11}$$

$$S_b = \mathrm{X} U_b X^T \tag{3-12}$$

$$S_t = \mathrm{X} U_t X^T \tag{3-13}$$

有了上述的三个散度矩阵，可以更好说明流形边距的概念。通常来说边距的概念就是外边界与内边界的距离，用同心圆来描述就是外圆的半径与内圆半径的差，从流形的角度来说就是流形M_i上的点与流形M_j的距离减去流形M_i的内部距离，这样定义的S_M就可以衡量不同流形间的分离程度，如公式 3-14。

$$S_M = \sum_{i,j} d(M_i, M_j) - \sum_i tr(M_i) \tag{3-14}$$

对于两个流形之间的距离$\sum_{i,j} d(M_i, M_j)$，其实可以使用流形M_i中距离流形M_j最短的点X_i的距离表示，因此$\sum_{i,j} d(M_i, M_j) = \sum_{i,j}\{\min d(X_i, M_j)\}$。这样就很容易地联想到了流形间图中的一些概念，可以使用点X_i在流形M_j上近邻点的线性组合表示流形M_j。这样两者的距离可以表示为公式 3-15,其本质就是流形间图的散度矩阵S_b。

$$\sum_{i,j} d(M_i, M_j) = \sum_{i,j} \left\| X_i - \sum_{j=1}^{k} (W_b)_{ij} X_j \right\|^2 \tag{3-15}$$

而对于流形内部的距离其实与公式 3-9 相关，因为进行最小化的表示误差$\varepsilon((W_w)_i)$就是该点与流形内部其他点的距离即流形内部的距离，其本质形式就是流形内图的散度矩阵S_w。最后流形边距的定义如公式 3-16。

$$S_M = S_b - S_w = X(U_b - U_w)X^T \tag{3-16}$$

3.2.3 最优化问题求解

在各种各样的流形学习方法中，最终的目的都是寻找一个子空间并且在该空间中不同流形上的数据更容易区分。在 LLRMM 算法中，其目标具体化就是让定义的流形边距在低维子空间中达到最大即$\max\ \mathrm{tr}\{Y(U_b - U_w)Y^T\}$。相较于流形内图散度矩阵来说，提出的流形总图的散度矩阵能更好的表示所有流形的固有特性，而其最终的目标是低维子空间中的表示误差最小即$\min\ \mathrm{tr}\{YU_tY^T\}$。

基于上述的分析，为了求解低维子空间需要满足上述的两个目标函数。具体来说就是求解如公式 3-17 所示的一个多目标优化问题。而为了求解这个多目标

$$\begin{cases} \max tr\{Y(U_b - U_w)Y^T\} \\ \min tr\{YU_tY^T\} \end{cases} \tag{3-17}$$

优化的问题，需要找到一个函数能很好地同时满足多个目标的要求，进而转换为单目标优化问题，最后这个函数如公式 3-18。

$$\max \frac{tr\{Y(U_b - U_w)Y^T\}}{tr\{YU_tY^T\}} \tag{3-18}$$

通常来说，传统的流形学习方法会遭遇样本外问题，因此会引用一个原始数

据和嵌入数据间的线性变换如Y = A^TX，并且这个线性变换会有正交的限制即A$A^T = I$，所以公式 3-18 最后表示如公式 3-19。这个优化的目标是找到一个线性

$$\begin{cases} max\frac{tr\{A^T\mathrm{X}(U_b-U_w)X^TA\}}{tr\{A^T\mathrm{X}U_tX^TA\}} = max\frac{tr(A^TS_MA)}{tr(A^TS_tA)} \\ s.t. \quad A^TA = I \end{cases} \tag{3-19}$$

变化既能最大化流形边距又能最小化全局流形的紧密性。

求解这个最优化问题同样使用拉格朗日乘子法，最后得到的特征值分解方程如公式 3-20。最后转换矩阵 A 是由特征分解的前 d 个特征值对应的特征向量构

$$S_MA_i = \lambda_iS_tA_i \tag{3-20}$$

成。转换矩阵不仅能应用在训练集的数据上，还能应用在测试集的数据上用于获取其低维的嵌入。最后 LLRMM 算法的具体流程如表 3.1。

表3.1 LLRMM 算法步骤

输入：原始数据X = $\{x_1, x_2, \dots, x_n\} \in R^{D\times n}$；数据类别C = $[C_1, C_2, \dots, C_C]$近邻点数 k；低维空间的维数d。

流程：

1. 通过合适的 k 值构建流形间图、流形内图和流形总图；

2. 通过公式 3-6、3-7、3-8 来计算相应的权值矩阵W_w、W_b、W_t

3. 计算对应的格拉姆矩阵$U_w = (\mathrm{I} - W_w)(\mathrm{I} - W_w)^T$、$U_b = (\mathrm{I} - W_b)(\mathrm{I} - W_b)^T$、$U_t = (\mathrm{I} - W_t)(\mathrm{I} - W_t)^T$；

4. 计算相关联的散度矩阵$S_w = XU_wX^T$、$S_b = XU_bX^T$、$S_t = XU_tX^T$；

5. 计算流形边距$S_M = S_b - S_w$；

6. 求解特征分解方程$S_M\mathrm{A} = \lambda S_tA$;

7. 对特征值进行从大到小的排序并将前 d 个特征值对应的特征向量组合成变换矩阵 A，通过Y = A^TX求解获取降维后的数据；

输出：变换矩阵 A 和降维后的数据 Y。

实验中的具体流程如图 3.2。

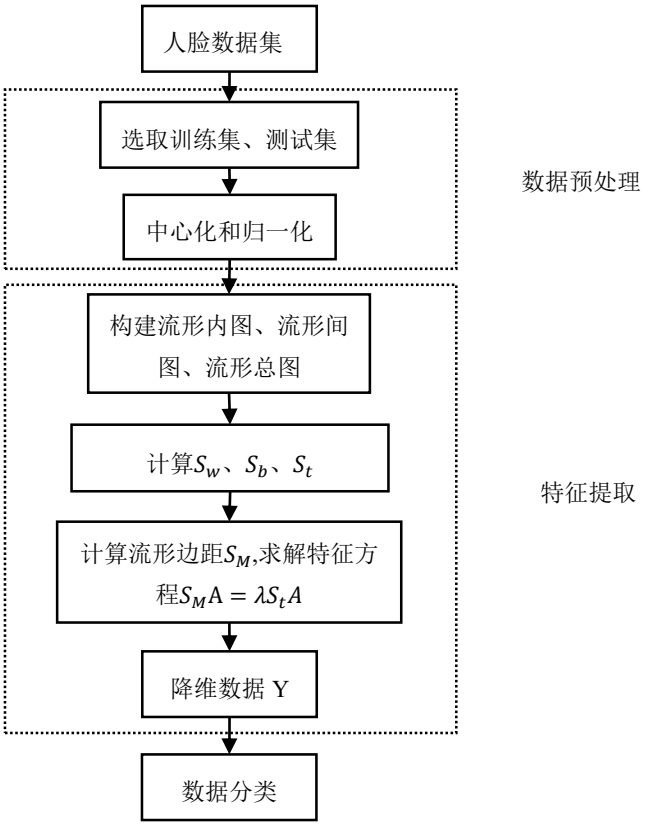

图3.2 LLRMM 算法实验流程

3.3 实验与分析

本节主要通过实验来验证 LLRMM 算法的性能，并且选取了传统的利用核函数方法 KPCA，三个流形学习方法 NDA、RDA、DMML 及一个度量学习方法 LMNN 来进行比较。对于 RDA 和 LLRMM 方法而言是 Fisher 形式的目标函数，因此会存在小样本尺寸问题，所以需要提前进行主成分分析，通过对原始数据的维数约简来避免问题。对于 KPCA、NDA、RDA、DMML 和 LLRMM 来说都是数据的特征提取方法即只进行了数据的降维处理，最后需要使用最近邻分类器对上述算法提取的特征进行分类。而对于用于比较的 LMNN 方法而言，因为 LMNN 本身就是一个分类器，所以不需要添加最近邻分类器即可进行比较。最后这些方法会应用到 AR、CMU PIE、Yale、YaleB 和 LFW 等人脸数据集中，并根据实验分别展示识别的性能。

3.3.1 AR 人脸数据集实验

在 AR 人脸数据集中有 4000 张彩色图片，其中包含 70 名男性和 56 名女性，并且通过改变面部表情、光照条件和遮挡等方式形成不同的正面人脸图。其中有

120 个人的图片被选中，包含 65 名男性和 55 名女性，并且拍摄于两个不同的时段，所以数据集被分成两个部分。每个部分中的每个人有 13 张彩色图片，这些图片可以转换为灰度图。在本实验中，会从总计 120 个人的每个人图集中选取 14 张灰度图，并且两个部分中各选 7 张。每张图片会被统一到 64×64 的像素大小。AR 数据子集中一个人的图集如图 3.3 所示。

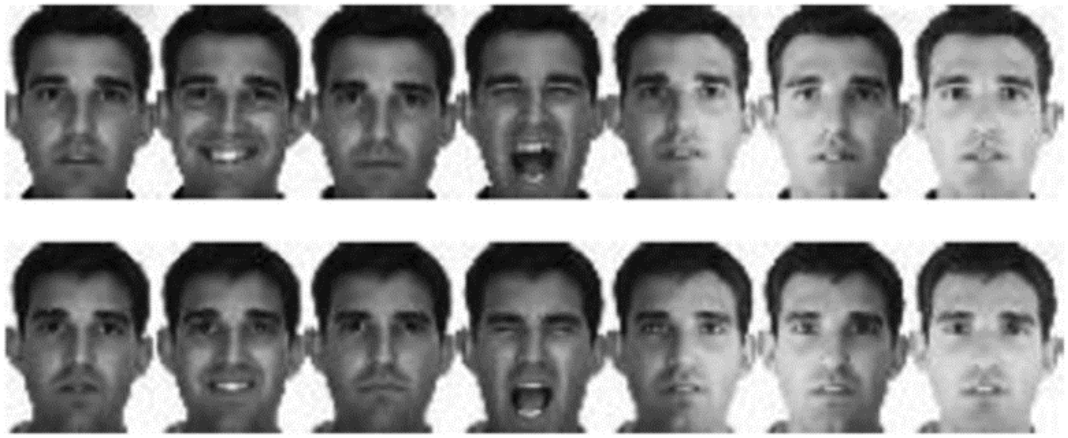

图3.3 AR 数据集的个人图集

在本实验中会测试训练集样本数目的影响，因此会分别选择 AR 数据子集中每个人的 5、6、7、8 张图片构成训练集时，而对应剩余的 9、8、7、6 张图片就会构成测试集。当使用近邻分类器进行分类时，会将 k 设置为 4、5、6 和 7 来进行比较。在进行试验时，会随机地选择 5、6、7、8 张图片 10 次，最后得到的平均识别率和标准偏差显示在表 3.2 中，从表中可以发现相同的样本数目时，LLRMM 算法的识别率更高一点。

表3.2 AR 数据集实验结果

	5 trains	6 trains	7 trains	8 trains
DMML+NN	85.01±1.24	87.35±1.48	87.56±1.62	90.43±1.35
KPCA+NN	90.36±1.53	91.76±1.97	92.54±1.28	93.52±1.01
NDA+NN	92.45±1.06	93.21±1.12	94.03±0.89	95.33±1.47
RDA+NN	92.68±1.74	93.99±2.10	94.89±1.02	96.21±0.72
LMNN	84.62±1.49	87.82±1.43	90.29±0.74	92.17±1.29
LLRMM+NN	95.54±1.32	96.44±1.22	97.21±0.99	98.01±0.78

3.3.2 CMU PIE 人脸数据集实验

CMU PIE 的人脸图像数据集被广泛使用，特别是用于对姿势，照明和表情的评估。在 CMU PIE 的人脸数据集中，包含 68 个人的 41368 幅人脸图像。在不同的姿势、照明和表情的条件下，通过 13 个同步摄像机进行 21 次拍摄来捕获人脸图像。在本实验中，会为每个人选择了 170 张灰度人脸图像，一起构成实验用的 CMU PIE 人脸子集。此外，每张图像的大小调整为 32×32。示例图像如图 3.4 所示。

图3.4 CMU PIE 人脸数据集的个人图集

在实验中，会从 CPU PIE 子集的每个类别中选择 60、70、80 和 90 张图像作为训练集，并以相对应剩余的 110、100、90 和 80 张图片作为测试集。构建流形内图、流形间图和总流形图时，k 值设置为 20。表 3.3 中显示的是实验统计结果，包括 CMU PIE 人脸子集中的平均识别率和相应的标准偏差。会随机地选择 60、70、80 和 90 张个人图像作为训练集，进行 10 次重复实验。从表 3.3 中可以得出，在相同数量的训练样本的情况下，LLRMM 的平均识别率要大于 KPCA，NDA，RDA，DMML 和 LMNN。

表3.3 CMU PIE 数据集实验结果

	60 trains	70 trains	80 trains	90 trains
DMML+NN	88.23±2.12	89.79±1.74	90.72±1.22	91.03±0.99
KPCA+NN	90.02±1.66	91.57±1.25	92.63±1.18	93.26±0.87
NDA+NN	91.34±1.35	92.76±1.62	93.29±1.37	94.08±1.35
RDA+NN	91.92±1.54	93.42±1.53	94.37±0.98	95.01±1.06
LMNN	92.29±0.26	93.70±0.34	94.99±0.20	95.24±0.17
LLRMM+NN	93.24±1.68	94.38±1.02	95.48±0.77	95.99±1.52

3.3.3 Yale 人脸数据集实验

Yale 人脸数据库是由耶鲁大学计算视觉与控制中心收集并构建的。在耶鲁人脸数据集中，有 15 个人包含 165 张图片，这些图片具有光照条件，面部表情以及戴或不戴眼镜等变化，因此每个人分别有 11 张图像。图 3.5 显示了来自 Yale 人脸数据集的某个个人图集，这些图像已被裁剪为 64×64 像素大小。

图3.5 Yale 数据集的个人图集

对耶鲁人脸数据集进行了 10 次重复实验，以获取平均识别率和相应的标准偏差。对于每种特征提取方法，将人员训练数据的数量分别设置为 4、5 和 6，其余的则作为测试样本。同时，当使用 k 近邻准则构造相应的图时，k 值分别设置为 3、4 和 5。表 3.4 分别显示了不同训练样本数目的 KPCA + NN、NDA + NN、RDA + NN、DMML + NN、LMNN 和 LLRMM + NN 算法的平均识别率和相应的标准偏差，其中 LLRMM 算法获得了最佳结果。

表3.4 Yale 数据集实验结果

	4 trains	5trains	6 trains
DMML+NN	81.35±1.64	86.38±1.43	88.94±1.86
KPCA+NN	82.73±1.57	90.12±1.25	92.76±0.89
NDA+NN	84.78±1.02	91.56±1.78	93.87±2.21
RDA+NN	87.76±0.92	92.18±1.56	94.59±1.36
LMNN	87.86±2.14	89.86±3.47	91.36±2.20
LLRMM+NN	89.14±0.85	93.33±1.36	95.73±1.11

3.3.4 YaleB 人脸数据集实验

YaleB 人脸数据集是对原始 Yale 人脸数据集的扩展，该数据集在 9 种姿势和 64 种光照条件下收集了 28 个人的 16128 个图像。在本实验中选择了其中的 38 个分组，同时选用每组在不同光照下的 64 个正面图像，这些图像一同构成了 YaleB 的数据子集。子集中的每个图像都被裁剪为 32×32 像素大小，子集中的一个分组的部分图像如图 3.6 所示。

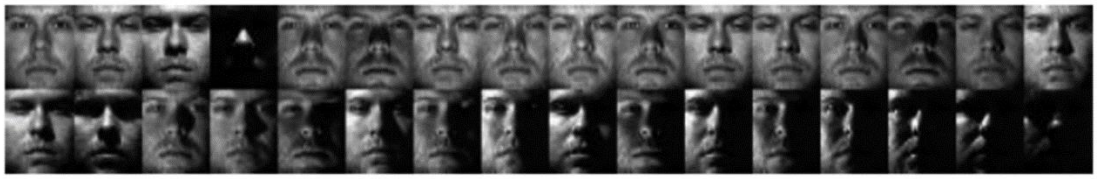

图3.6 YaleB 数据集的个人图集

进行实验时，会分别选择每组的 20、30 和 40 张图像作为训练样本，其余部分的 44、34 和 24 个图像作为测试样本。此外，在确定 k 最近邻时，k 值设置为 12。对重复 10 次的实验进行统计，其结果如表 3.5 所示。从表中可以发现，无论每个类别选择多少训练样本，提出的结合最近邻分类器的 LLRMM 算法的性能也更加优异于其他方法，如 DMML + NN、KPCA + NN、NDA + NN、RDA + NN 和 LMNN 分类器。

表3.5 YaleB 数据集实验结果

	20 trains	30 trains	40 trains
DMML+NN	72.13±1.78	86.75±2.31	89.57±2.13
KPCA+NN	73.24±2.56	88.38±2.78	90.35±1.72
NDA+NN	76.89±1.98	89.78±1.96	91.48±1.64
RDA+NN	78.87±1.35	90.88±1.84	92.53±1.47
LMNN	79.73±0.76	91.38±1.11	93.51±2.01
LLRMM+NN	82.12±2.34	92.57±1.04	95.27±1.45

3.3.5 LFW 人脸数据集实验

为了对不受约束的图像进行身份验证和面部识别的研究，带有类别信息的人脸图片集 LFW 被创建。其中包含来自 1680 个人的在不受约束条件下的 13,000 张以上的面部图像。在本实验中，使用的子集包含来自 86 个人的 1251 幅图像，每个类别只有 10–20 幅图像。每个人脸图像均手动裁剪为 32×32 的大小。LFW 子集中一个人的一些面部图像如图 3.7 所示。

图3.7 LFW 数据集的个人图集

在实验中，会随机选择每个人的 6、7 和 8 张图像作为训练样本并且剩余的图片作为测试集。另外选择 6、7 和 8 作为样本集数目时对应的 k 值会被设置为 5、6 和 7。最后进行 10 次重复的实验后，表 3.6 展示了统计的实验结果。由于 LFW 数据集对于图像分类来说是一个非常具有挑战性的数据集，因此表中各种方法的识别率不是很高。但是仍然可以发现 LLRMM 方法的平均识别率比其他方法更高。

表3.6 LFW 数据集实验结果

	6 trains	7 trains	8 trains
DMML+NN	29.88±1.73	32.26±1.45	33.81±1.95
KPCA+NN	31.52±0.94	34.45±1.76	35.27±2.18
NDA+NN	32.83±1.54	35.03±1.99	36.28±1.96
RDA+NN	33.36±1.94	36.30±1.58	37.21±1.43
LMNN	34.56±1.25	36.24±2.12	37.56±2.02
LLRMM+NN	36.58±1.34	38.12±1.72	40.27±1.57

3.3.6 结果分析

根据以上实验结果，可以发现数据规模对数据的识别性能有影响。在实验中，采用了 AR，CMU PIE，YaleB 和 LFW 等大型数据集以及 Yale 等小型数据集进行测试。在 AR，CMU PIE，YaleB 和 LFW 上的实验结果表明，LLRMM + NN 优于 DMML + NN，KPCA + NN，NDA + NN，RDA + NN 和 LMNN。对于大数据集而言，构建流形间图和流形内图时会用到更多的数据点。因此可以很好地发现流形的局部性，并充分利用监督信息，这有助于数据分类。对于 Yale 数据集来说，尽管在选择相同数量的训练样本时，LLRMM + NN 的平均识别率仍比使用 DMML + NN，KPCA + NN，NDA + NN，RDA + NN 和 LMNN 的平均识别率大。但是 LLRMM+NN 的方法没有比 DMML + NN，KPCA + NN，NDA + NN，RDA + NN 大多少，原因在于 LLRMM，DMML，NDA 和 RDA 是基于流形学习的方法，在很大程度上取决于 k 最近邻图的构造，如果训练样本太小而无法挖掘流形结构，则相应的方法将无法获得更好的效果。

3.4 本章小结

本章主要对提出的基于流形边距的特征提取方法进行了介绍。首先是引言部分，先从基于 LLE 和 LDA 结合的方法抛出构建 k 近邻图时存在的问题，再从解决问题的手段中引出流形内图、流形间图及流形边距的概念，最后提出 LLRMM 算法。第二节则是对 LLRMM 算法的具体介绍，从局部线性表示、流形边距的定义、最优化问题求解三个方面循序渐进的说明算法的原理与实现。最后则是实验与分析部分，从 AR、CMU PIE、Yale、YaleB、LFW 五个数据集分别进行了实验说明，先是数据集本身信息的介绍，然后是实验中参数的说明，最后是实验结果的描述与分析。

第4章 基于几何感知距离的特征提取方法

4.1 引言

现在的流形学习方法中，为了利用数据的类别信息往往都会构建类内与类间这两种近邻图。如边距 Fisher 分析(marginal Fisher analysis, MFA)算法和判别多流形学习(discriminant multi-manifold learning, DMML)算法会通过构建这两种近邻图来表示类别内部的紧密度和类别之间的分离度。但是这些方法仅用这两个图来利用类别信息可能是不够充分的，因此一种将每个类别都看作一个点并构建对应的类别近邻图的想法应运而生。同时需要通过某种距离度量来衡量类别间的分离度，最后选取了对数欧式距离作为距离度量，因为该距离是黎曼测度，比传统的欧式距离可能更适合流形学习。

4.2 基于几何感知距离的算法

因为基于类别信息构建的每一个类别近邻图能很好的表现类别内部的信息，同时将每一个类别看成一个点，并用对数欧式距离可以很好地度量类别间的距离。所以本章提出了这种基于几何感知距离的算法，命名为对数欧式距离度量学习(LEDML)算法。首先定义类别图的散度并计算类别间的度量距离，然后和 LLRMM 算法一样结合全局的线性信息，这样就能充分地考虑类别间的信息和全局的信息来提高判别能力。下面将分成三部分来具体介绍提出的 LEDML 算法。

4.2.1 类内散度

本小节主要说明了每个类别点的近邻图构建和散度的定义。对于高维数据 $X = \{X_1, X_2, \dots, X_n\}$来说其包含 n 个样本，并且这些数据分成 C 个类别$\{Y_1, Y_2, \dots, Y_c\}$。这样会有 C 个类别点及构建的 C 个类别图与对应的 C 个散度矩阵。假设某个类别中的任意一条样本为X_i,那么在该类别图中存在 k 个近邻点$X_j(j = 1,2, \dots k)$。为了体现其局部的性质需要使用线性表示的技巧，其进行误差表示的优化目标函数如公式 4-1。其中X_j是X_i的 k 个同类别近邻点，其中的权值$(W_c)_{ij}$也存在单位化的

$$\varepsilon((W_c)_i) = \min_{W_c} ||X_i - \sum_{j=1}^{k}(W_c)_{ij}X_j||^2 \tag{4-1}$$

限制即$\sum_{j=1}^{k}(W_c)_{ij} = 1$，最后该优化问题通过拉格朗日乘子法后得到的权值解如

公式 4-2。其中$G_{jt}=(X_i-X_j)(X_i-X_t)$，$\mathrm{ClassN}(X_i)$表示同类别的 k 个近邻点，

$$(W_c)_i=\begin{cases}\frac{\sum_{t=1}^{k}G_{jt}^{-1}}{\sum_{m=1}^{k}\sum_{l=1}^{k}G_{lm}^{-1}} & X_j\in ClassN(X_i)\\ 0 & otherwise\end{cases} \tag{4-2}$$

otherwise 表示同类别但非近邻的点。其形式和计算与第三章的局部线性表示大同小异，其不同点在于第三章中最后计算的权值矩阵只有一个，而此处的则包含有多个权值矩阵。实际上整体的权值矩阵可以由这些分散的权值矩阵组合而来，之所以进行分开表示是为了便于计算每个点的散度矩阵。对于公式 4-1 使用二次型重新表示后如公式 4-3。其中$U_c=(\mathrm{I}-W_c)(\mathrm{I}-W_c)^T$,当适用于所有该类别的点

$$\varepsilon((W_c)_i)=\min_{W_c} tr\{\textstyle\sum_{i,j}(U_c)_{ij}(X_iX_j)\} \tag{4-3}$$

后可以表示为公式 4-4。W_c是由公式 4-2 计算的值构成的权值矩阵，最后该类别

$$\varepsilon(W_c)=\min_{W_c} tr(X_cU_wX_c{}^T) \tag{4-4}$$

的散度矩阵如公式 4-5。该散度表示了一个类别点的性质，这样的 C 个散度矩阵

$$S_c=X_c(\mathrm{I}-W_c)(\mathrm{I}-W_c)^TX_c^T \tag{4-5}$$

可以用来进行类间距离的计算。

4.2.2 类间距离度量

对于类别之间的距离度量如图 4.1 所示，其中包含三个类别流形M_1、M_2、M_3。

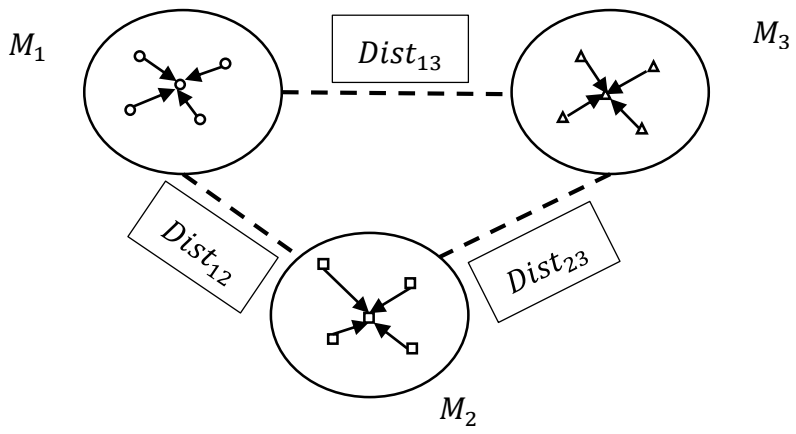

图4.1 类间距离示意图

这样就构成了上一小节中描述的三个类别点，同时可以计算出其对应的散度矩阵S_1、S_2、S_3，最后则是类别间的距离$Dist_{12}$、$Dist_{13}$、$Dist_{23}$的计算。

机器学习中的距离度量有许多种，有些是从几何学中的定义而来，有些是从统计学中的定义而来。首先是常用的欧式距离，假设点$\mathrm{X}=(X_1,X_2,\dots,X_n)$和点$\mathrm{Y}=(Y_1,Y_2,\dots,Y_n)$,其欧式空间的距离如公式 4-6。而欧式距离其实是闵氏距离中的

$$\mathrm{d}(\mathrm{X},\mathrm{Y})=\|X-Y\|_2=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \tag{4-6}$$

一员，是闵氏距离中 p 为 2 的情况，闵氏距离的表示如公式 4-7。从统计学中而

$$\mathrm{d}(\mathrm{X},\mathrm{Y})=\|X-Y\|_p=(\sum_{i=1}^{n}|x_i-y_i|^{\mathrm{p}})^{\frac{1}{p}} \tag{4-7}$$

来的经典距离则是马氏距离，假设有 m 个样本点$X_1,X_2,\dots,X_m$,其协方差矩阵为 S，那么样本点X_i和X_j之间的马氏距离如公式 4-8。

$$\mathrm{d}(X_i,X_j)=\sqrt{(X_i-X_j)^T S^{-1}(X_i-X_j)} \tag{4-8}$$

但是欧式距离是适用欧式空间的，对于流形中的距离度量可能不太适用。提及流形中的距离可能会联想到黎曼测度，因此需要先了解黎曼流形，黎曼流形属于微分流形，对于其中每个点 x 的切空间都定义了点积，而且其数值随 x 平滑地改变。它容许定义弧线长度、角度、面积、体积、曲率、函数梯度及向量域的散度等概念。当一个流形 M 被赋予合适的黎曼测度 d 时,这个流形 M 就成为一个黎曼流形，而黎曼测度就是流形上每个点切空间的点积。黎曼测度通常经过对数映射把流形上的点映射到切空间上，再在切空间上通过定义的点积来计算距离。常用的黎曼测度有对数欧式距离(Log-Euclidean)和仿射不变黎曼测度(AIRM)，其具体表示如公式 4-9 和 4-10。

$$\mathrm{d}(\mathrm{X},\mathrm{Y})=\|\log(X)-\log(Y)\|_F^2 \tag{4-9}$$

$$\mathrm{d}(\mathrm{X},\mathrm{Y})=\left\|\log(X^{-\frac{1}{2}}YX^{-\frac{1}{2}}\right\|_F \tag{4-10}$$

最后由于每个类别点的散度矩阵S_c是对称正定性，因此可以采用对数欧式距离进行类别间的距离表示，进行求和后的结果如公式 4-11。其中S_i和S_j表示 C 个类别点中的两个点的散度矩阵，并且 i 和 j 不相等。

$$\mathrm{D} = \sum_{i,j}\left\|\log(S_i) - \log(S_j)\right\|_F^2 \tag{4-11}$$

4.2.3 最优化问题求解

在特征提取方法中，通常是提出一个目标来进行判别分析，并且会将这个目标转换为一个对应的优化问题，最后求解优化问题来获取用于降低维度的转换矩阵。首先 LEDML 方法的目标包含两部分，一个部分是类间距离的最大化，这样有利于进行判别分析和分类等，第二部分则是全局表示误差的最小化，这部分则是和上一章的 LLRMM 方法一样，保留了 LLE 算法的目的。

如果降维后的数据为$\mathrm{Y} = (Y_1, Y_2, \ldots, Y_3)$,用转换矩阵 A 和原始数据 X 可以表示为$\mathrm{Y} = A^T X$。那么在低维数据中的类间距离可以表示为公式 4-12。而对于低维数

$$\begin{aligned} D' &= \sum_{i,j}\left\|\log(S_i) - \log\left(S_j\right)\right\|_F^2 \\ &= \sum_{i,j}\left\|\log(Y_i(I-W_i)(I-W_i)^T Y_i^T) - \log(Y_j(I-W_j)(I-W_j)^T Y_j^T)\right\|_F^2 \\ &= A^T\left(\sum_{i,j}\left\|\log(X_i(I-W_i)(I-W_i)^T X_i^T) - \log(X_j(I-W_j)(I-W_j)^T X_j^T)\right\|_F^2\right)A \\ &= A^T DA \end{aligned} \tag{4-12}$$

据中的全局表示误差在第三章中也有过描述，最后得到的结果如公式 4-13。其中

$$\varepsilon = \mathrm{Y}U_t Y^T = A^T X(I-W_t)(I-W_t)^T X^T A \tag{4-13}$$

W_t为利用所有数据进行线性表示得到的权值矩阵。对两者进行最大化和最小化时其目标函数为公式 4-14,这是一个多目标优化的问题，因此需要变成单目标优

$$\begin{cases} \max tr\{A^T \sum_{i,j}\left\|\log(S_i) - \log(S_j)\right\|_F^2 A\} \\ \min tr\{A^T X(I-W_t)(I-W_t)^T X^T A\} \end{cases} \tag{4-14}$$

化问题，与 LDA 一样使用瑞利熵的形式，最后得到的目标函数如公式 4-15。最

$$\begin{cases} \max \dfrac{tr\{A^T \sum_{i,j}\left\|\log(S_i)-\log(S_j)\right\|_F^2 A\}}{tr\{A^T X(I-W_t)(I-W_t)^T X^T\}} \\ s.t. \quad A^T A = I \end{cases} \tag{4-15}$$

后通过拉格朗日乘子法得到特征分解方程如公式 4-16。进行特征分解后，将得到

$$DA_i = \lambda_i S_t A_i \tag{4-16}$$

的特征值进行排序并选取前 d 大的值，其对应的 d 个特征向量即可构成转换矩阵 A。具体的算法流程如表 4.1。

表4.1 LEDML 算法步骤

输入：原始数据$\mathrm{X}=\{x_1,x_2,\ldots,x_n\}\in R^{D\times n}$；数据类别$\mathrm{C}=[C_1,C_2,\ldots,C_C]$近邻点数 k；低维空间的维数d。

流程：

1. 通过合适的 k 值为每个类别的数据构建一个近邻图，以及总的近邻图；

2. 通过公式 4-2 来计算每个类别的权值矩阵W_c,以及计算全局的W_t;

3. 计算每个类别对应的格拉姆矩阵$U_C=(\mathrm{I}-W_c)(\mathrm{I}-W_c)^T$和全局的格拉姆矩阵$U_t=(\mathrm{I}-W_t)(\mathrm{I}-W_t)^T$；

4. 计算相关联的每个类别的散度矩阵$S_c=\mathrm{X}_cU_cX_c^T$和$S_t=XU_tX^T$;

5. 通过公式 4-11 计算类间距离总和 D；

6. 求解特征分解方程 $\mathrm{DA}=\lambda S_tA$;

7. 对特征值进行从大到小的排序并将前 d 个特征值对应的特征向量组合成变换矩阵 A，通过$\mathrm{Y}=A^TX$求解获取降维后的数据；

输出：变换矩阵 A 和降维后的数据 Y。

其进行实验的算法流程如图 4.2。

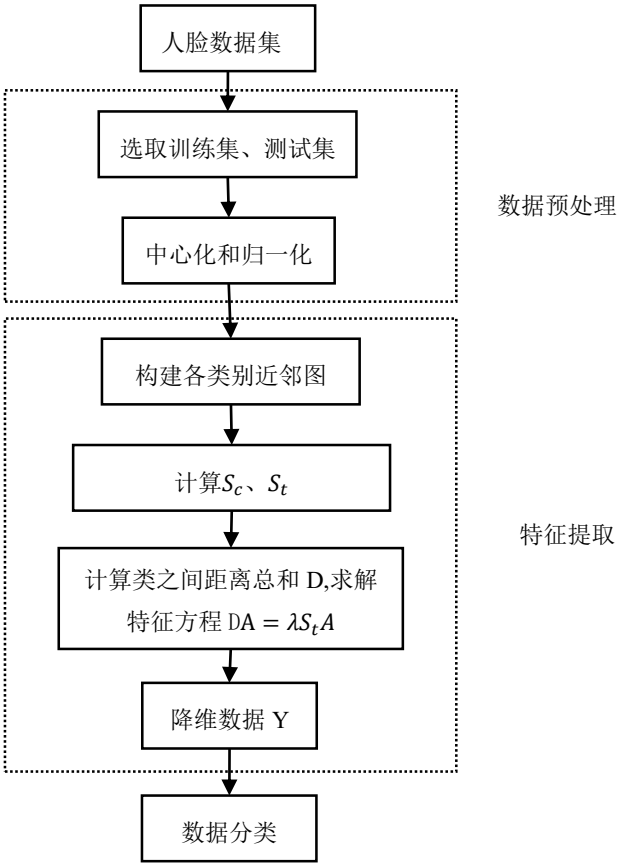

图4.2 LEDML 算法实验流程

4.3 实验结果与分析

本节中既对基于对数欧式距离度量学习方法进行了自身参数的比较，同时也与其他降维方法进行了结果比较。自身对比主要包括不同维度间的比较、欧氏距离与对数欧式距离间的比较。而用于比较的其他降维方法包括核主成分分析(KPCA)、线性判别分析(LDA)和局部线性嵌入(LLE)，KPCA 和 LLE 方法是非监督式方法，可以说明类别信息的作用，选择 LDA 则是体现线性方法和非线性方法的区别。实验使用的数据集包括 Yale、CMU PIE 和 FERET，其中 CMU PIE 和 FERET 分别选择了其中的 15 个类别和 20 个类别的数据构成子集进行实验。

表 4.2 中显示了各个数据集进行实验的属性和参数。“每类训练样本数量”表示从数据集的每个类别中选出相应数量的样本构成训练集，对应的剩余部分则构成测试集。如 Yale 数据中 4、5、6 训练集对应的测试集为 7、6、5。表格中的 kw 参数表示算法中每个类别点构建近邻图时的近邻参数 k，对应的 kt 则表示构建全局近邻图时的近邻参数 k。另外为了减少实验的随机性，会随机的选取训练样本 10 次，将 10 次实验运行的平均准确率作为参考。

表4.2 各数据集实验参数

	Yale	CMU PIE	FERET
人数类别	15	15	20
每类样本数量	11	170	7
每类训练样本数量	4、5、6	50、60、70	3、4、5
kw	3	5	2
kt	6	25	4

对于 FERET 数据集，其整个数据库有 1400 张人脸图像，每个类别包含不同角度、不同光照、不同表情等条件下的 7 张图片。FERET 子集中一个人的一些面部图像如图 4.3 所示。

图4.3 FERET 数据集个人图集

4.3.1 维度对比实验

首先是实验自身对比中的维度对比。在进行特征提取时，最后低维数据的维度确定十分重要。维度选取过低可能导致数据原本的特性丢失而降低识别率，维度过高可能导致识别速度降低，甚至存在保留数据原本无用的特性而降低识别率的可能。本实验主要从 5、10、15、20、25、30 这 6 个维数进行了实验对比，并将各数据集的平均准确率通过折线图的形式进行了展示，具体结果如图 4.4、图 4.5 和图 4.6。

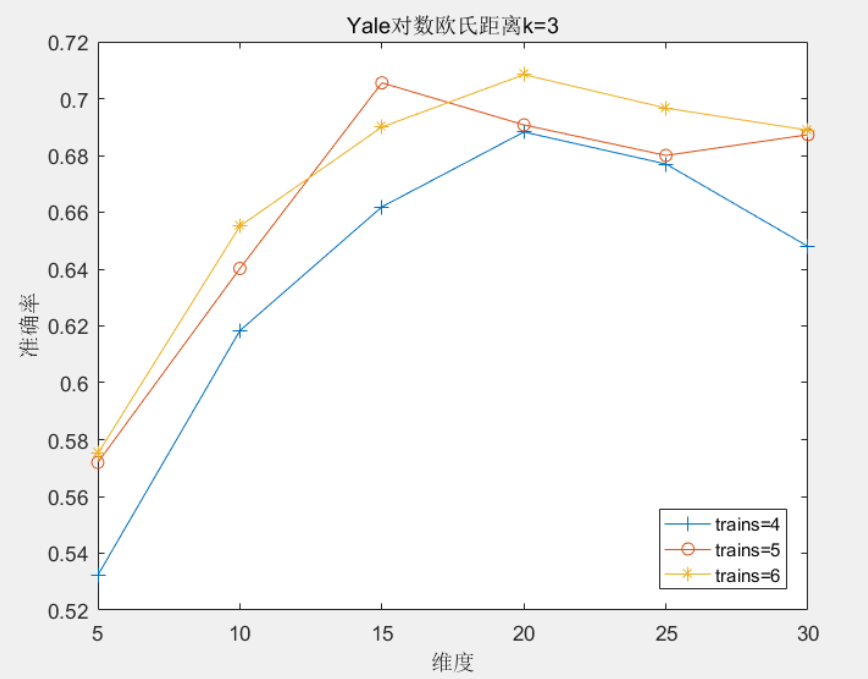

图4.4 Yale 数据集的维数对比

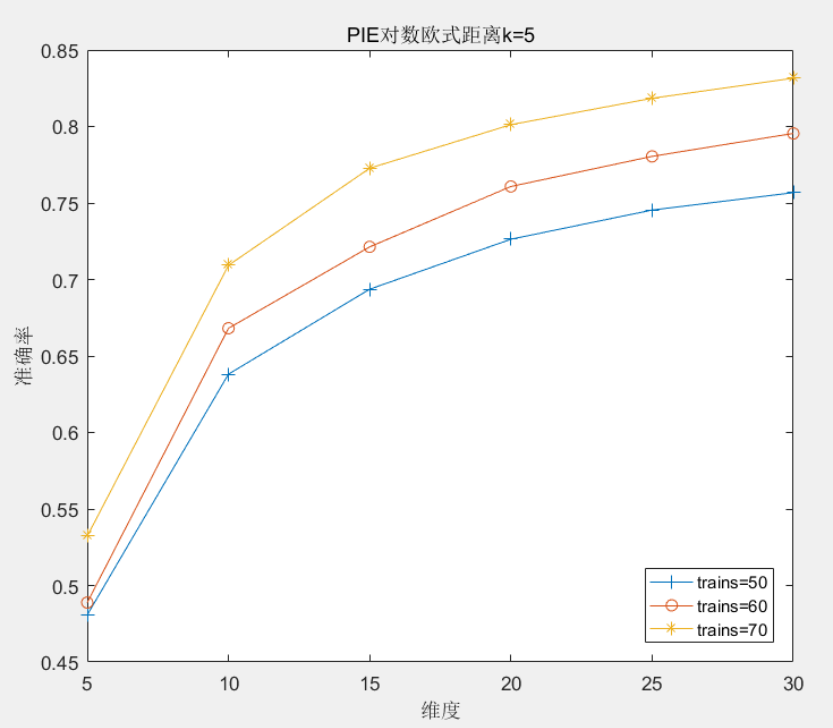

图4.5 PIE 数据集的维数对比

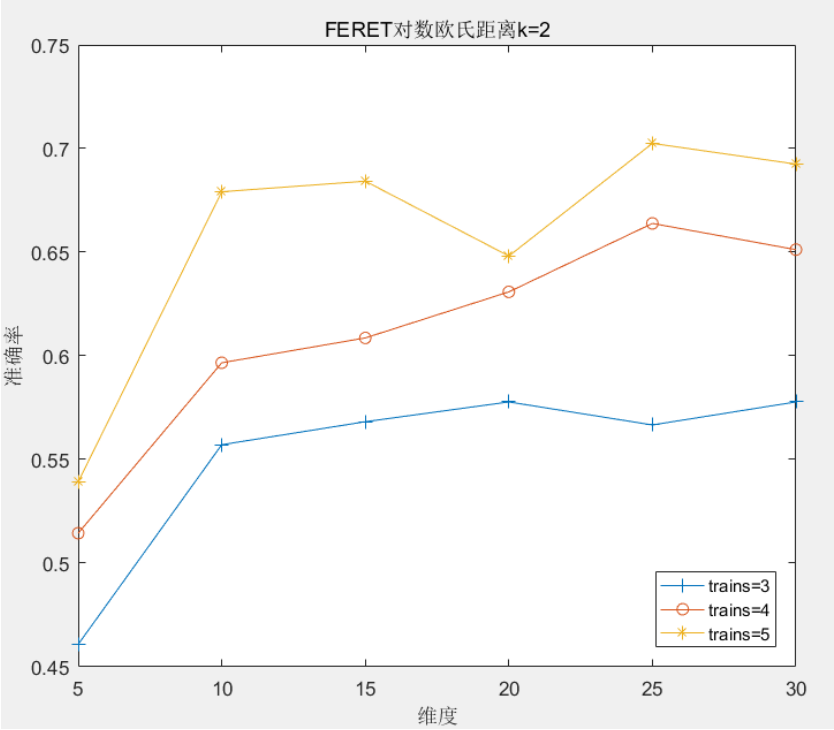

图4.6 FERET 数据集的维数对比

三幅图中的准确率基本是递增的，但是维度较小时的增长较快，然后趋近平缓。说明过低的维度丢失了数据特性导致识别率不高，而到达某个临界值后继续增加的维数对于准确率没有明显贡献，甚至可能导致负作用如 Yale 数据集的结果。

4.3.2 距离对比实验

本小节是自身实验对比中的距离度量对比，主要是欧式距离和对数欧式距离间的比较。为了方便对比，在对 10 次实验结果进行平均的基础上，也对每类训练样本数 trains 的结果进行了平均。最后通过柱状统计图的形式展示了实验结果，如图 4.7、图 4.8 和图 4.9。

三幅图中对数欧式距离和欧式距离的识别率相差较小，在某些维数的情况下对数欧式距离的准确率稍好，这说明在进行距离的度量时选用黎曼测度的对数欧式距离是可行的，这为其他算法进行距离度量时提供了一个不错的选择方向。

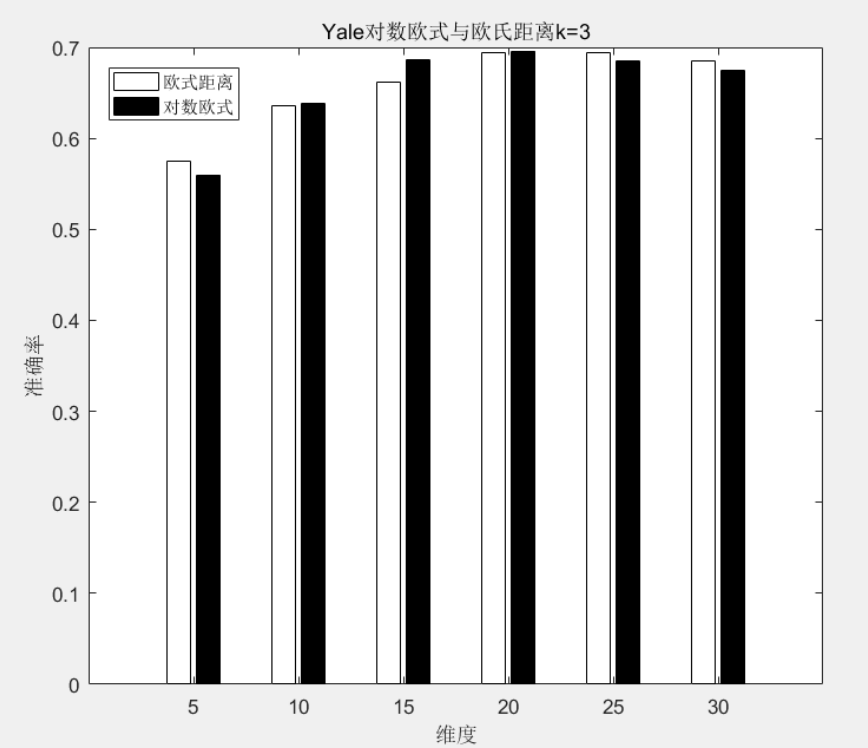

图4.7 Yale 数据集的距离对比

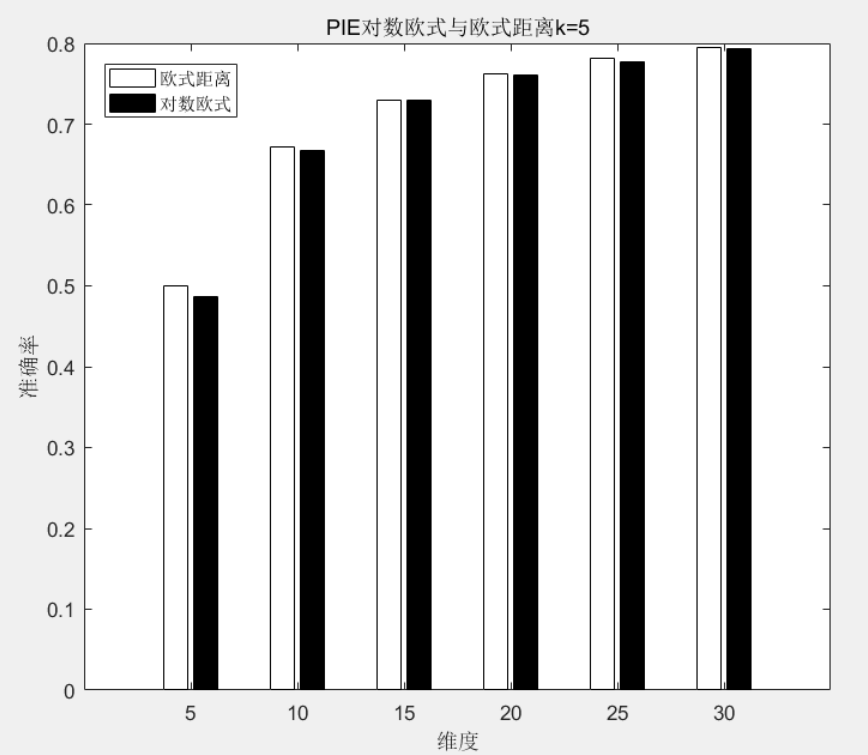

图4.8 PIE 数据集的距离对比

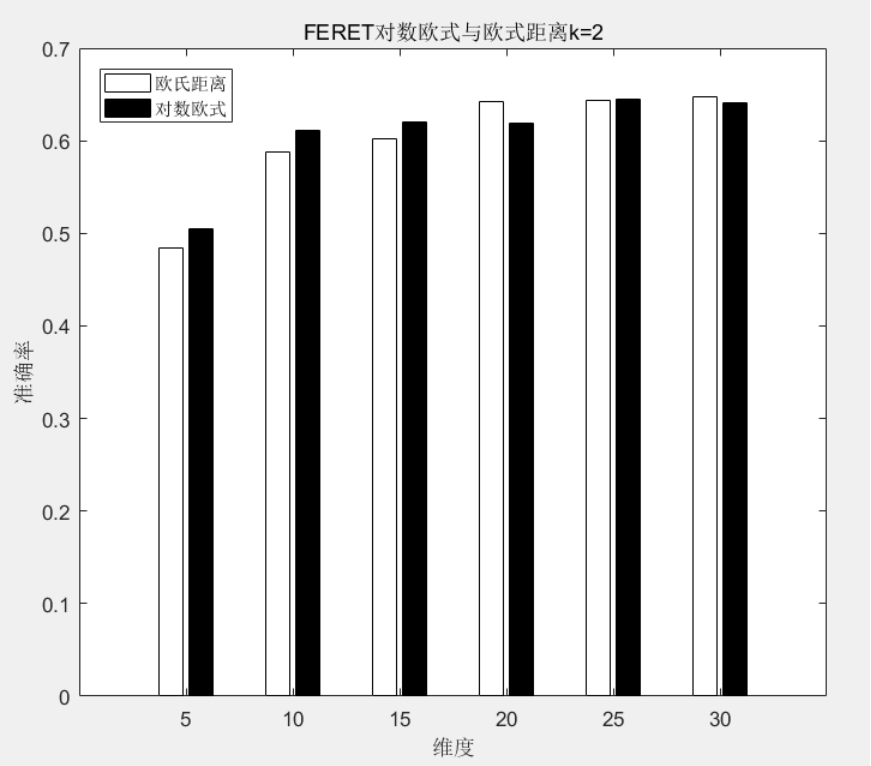

图4.9 FERET 的距离对比

4.3.3 降维方法对比试验

最后是基于对数欧式距离度量学习方法同其他常用降维方法之间的对比。KPCA、LDA、LLE 等降维方法同样在 Yale、CMU PIE、FERET 数据集上进行实验，其训练集和测试集的分割方式与上两节实验相同。最后实验结果如表 4.3。

表4.3 各降维实验对比结果

		KPCA+NN	LDA+NN	LLE+NN	LEDML+NN
Yale	4 trains	62.22±3.12	74.41±2.76	60.86±1.56	67.70±3.21
	5 trains	65.15±3.44	74.30±3.60	60.23±3.62	68.00±2.57
	6 trains	65.82±3.57	76.84±3.60	63.48±2.60	69.67±4.49
PIE	50 trains	54.00±1.46	82.85±1.51	67.60±1.40	74.95±1.01
	60 trains	57.50±1.41	85.95±2.06	69.71±1.63	79.84±0.81
	70 trains	59.70±1.48	88.12±1.01	74.10±1.80	83.25±1.72
FERET	3 trains	55.54±2.87	64.96±5.04	61.50±5.80	56.67±4.54
	4 trains	64.00±5.12	69.50±2.78	65.94±5.21	66.39±4.31
	5 trains	63.17±5.10	76.25±5.80	66.92±6.75	70.25±5.33

从表格中首先可以发现，训练样本越多识别的效果越好，这与之前折线图显示的结果基本相符。这很容易解释，因为越多的训练集数据获取的信息也越多，最后训练的模型效果就更好。另外发现本章提出的 LEDML 方法与 LDA 方法相较于其他方法有更好的效果，可能与利用数据类别信息有关。

4.3.4 结果分析

根据以上实验结果。从实验自身对比可以发现，低维数据的维数确定十分重要，并非越大越好，在接近阈值时其增长趋于平缓，超过后甚至可能准确率降低。另外，利用对数欧式距离作为距离度量与传统欧式距离相比也有不错表现，可以作为一种选择方案。最后基于对数欧式距离的度量学习方法较其他的无监督式方法有更好的表现，原因在于其充分利用数据的类别信息和选用了恰当的距离度量。但是比同样为监督式方法的 LDA 略差，可能是构建近邻图的 k 值不是最佳值，无法充分发掘数据的局部信息及流形结构。

4.4 本章小结

本章主要对提出的基于对数欧式距离的度量学习方法进行了介绍。首先是引言部分，从 MFA 和 DMML 方法中类间近邻图和类内近邻图的想法，进一步提出每个类别构建近邻图作为一个类别点的概念，并用对数欧式距离进行类别点间的距离衡量，最后提出 LEDML 算法。第二节则是对 LEDML 算法的具体介绍，从类别点的类内散度、类别点间的距离度量、最优化问题求解三个方面说明算法的原理与实现。最后则是实验与分析部分，从 Yale、CMU PIE、FERET 三个数据集分别进行了实验。先是低维维数和距离度量的两个自身对比实验，然后是将 LEDML 同 KPCA、LDA、LLE 等其他降维方法进行对比的实验，最后是实验结果的总结与分析。

第5章 总结与展望

在互联网时代，数据不仅存在数量大的特性，还存在高维度的特性，这样进行处理时就会存在“维数灾难”问题。因此维数约简成为了机器学习的研究热点，从不同的角度会提出不同的降维方法，但是都可以归结为特征选择和特征提取两大类。流形由于其局部与欧氏空间同胚的性质，能发现高维空间中存在的低维空间，所以被广泛的应用于降维问题中。最终在机器学习中形成了流形学习这一分支。然而流形学习只发展了短短的二三十年，必然会存在一些问题，如样本外点的学习、类别信息的使用、合适距离的选取等问题。针对这些问题，研究人员做了许多尝试。本文为了充分使用数据的类别信息，提出了一种基于流形边距的局部线性表示法，其思想主要是 LDA 和 LLE 的结合，同时定义了流形边距的概念。另外为了选取合适的距离度量，结合了黎曼流形中的黎曼测度，提出了一种基于对数欧式距离的度量学习方法。

5.1 本文主要工作

针对两种提出的方法，本文进行了如下工作：

(1) 首先说明了特征提取的背景与意义，并介绍了特征提取和流形学习相关的研究现状。从线性和非线性的角度描述了常用的特征提取方法的思想和步骤，其中包括一些经典的流形学习的方法。在分析对比了这些算法的优缺点后，提出本文改进后的方法并进行了阐述。

(2) 针对类别信息使用的问题，本文提出了基于流形边距的局部线性表示法。该方法首先借鉴 LDA 的思想确定了流形内、流形间、全局流形的概念，同时分别按照 LLE 算法的思路进行处理，创建了流形内图、流形间图和流形总图并计算了相应的散度矩阵。基于这些类别信息的数据，LLRMM 算法定义了流形边距的概念，其表示了流形内部数据与非该流形数据的分离程度。为了用于数据降维，需要在最大化流形边距的同时，还要最小化全局线性表示误差。统一来说就是数据在全局上尽可能地紧密而本流形与非本流形间尽可能地分开。最后 LLRMM 算法主要通过两种手段来充分利用类别信息，第一种是和 LDA 算法一样利用类别信息对数据进行初步区分，第二种则通过定义的流形边距二次利用类别信息，因此本方法比其他的利用类别信息的方法有更好的效果。

(3) 早先提出的方法没有将每个类别单独看待，并且在一些距离度量上没有进行合理地选择。因此本文提出了基于对数欧式距离的度量学习算法。首先 LEDML 方法也使用了数据的类别信息，而与其他类似方法的最大不同点在于，该方法将每个类别单独看成一个类别点，并不是使用一个全局的类别内图，这样为衡量类别间的分离度进行了铺垫。通过类似点与点之间的距离定义，可以通过

类别点间的距离来量化分离程度。而在具体的计算中，会用每个类别图的散度矩阵作为该点的表示，并使用黎曼测度的对数欧式距离来进行距离计算。基于此，其优化目标是类别间的对数欧式距离总和最大化和全局线性表示误差最小化。最后由于对数欧式距离相比于欧式距离更适合流形数据的情况，因此该方法可能有更好的判别效果。

5.2 未来工作的展望

维数约简与特征提取的研究是机器学习的关键，因此相关的研究经历了长期的发展。而作为其研究分支的流形学习是一股新鲜的血液，没有经历长期的发展到达成熟的阶段。这表明一些已有的流形学习方法或多或少的会存在一些问题，所以为了充分发挥流形学习的价值，需要研究者不断地改良方法继续努力。现在，在本文已做的研究基础上提出了一些可以改良和进步的地方：

(1) 本文提出的流形边距的概念和用对数欧式距离进行类别间距离度量的想法，也可以应用于其他的同样利用类别信息的流形学习方法。

(2) 本文提出的两种方法都是基于线性表示的，对于其中近邻图的 k 值选取可能需要更多的验证。另外黎曼测度也不仅仅有对数欧式距离这一种，还有仿射不变的黎曼测度和 Bregman 散度等，因此需要进一步的对比试验。

(3) 与许多其他的理论研究一样，流形学习也存在人工数据集上表现良好，但真实数据集上表现不理想的情况。因此为了让流形学习更具有实际应用价值，能更好的进行发展。需要进行从理论到实际应用的研究，如真实数据集与理论数据集的差异研究，加快算法运行速度的研究等。