参赛密码 _____

（由组委会填写）

第十二届“中关村青联杯”全国研究生数学建模竞赛

学　　校	上海交通大学
参赛队号	**10248091**
队员姓名	**1.**杨晓宇
	2.张　哲
	3.甄　诚

参赛密码 _____

（由组委会填写）

第十二届“中关村青联杯”全国研究生数学建模竞赛

题 目 数据的多流形结构分析

摘 要：

进入信息时代，人们生活的各个方面都离不开数据的处理，对大规模数据的分析与处理在科学研究领域占据着越来越重要的地位。基于流形学习的聚类算法，是当前计算机模式识别领域的新思路。本文通过引入基于流形学习的聚类方法来解决复杂数据集的聚类问题，主要采用了 PCA,SSC,SMMC,mum-Cluster,LSC 等算法以及其相应的改进算法解决了一系列不同数据特征、不同实际场景的聚类问题，其中包括独立与非独立子空间聚类问题、多流形聚类问题以及多种实际问题，并且在参数调节过程中，提出了一种基于 PID 的参数调节算法，这在本文的问题解决过程中起到了至关重要的作用。

在问题一中，我们采用了 PCA 算法获得了一个基本解，并从中得到启发，采用了基于流形学习的 SSC 算法来进行数据的聚类，通过调节 SSC 的参数，得到合理的相似度矩阵，通过 K-means 聚类算法得到了高精度的聚类结果，分类结果为前 40 个点和最后 60 个点为一类，中间 100 个点分为一类。

在问题二中，我们采用一种算法统一解决四种不同类型的数据，引入了 SMMC 算法，其核心思想是：从相似性矩阵的角度出发，充分利用流形采样点所内含的自然局部集合结构信息来辅助构造更合适的相似性矩阵并而发现正确的流形聚类。相对而言，题目中的四个数据集可以利用基本局部几何结构构造辅助信息来得到正确的结果，但 SMMC 受参数影响明显，参数的合理调节确实为一

个难点，参数的好坏直接影响到分类结果的好坏，我们根据 SMMC 算法的经验公式，创新性地提出一种基于 PID 思想的参数调节策略，能够快速合理的逼近所需要的参数组合，得到所需要的聚类结果。同时，考虑到聚类算法的效能，我们在文中针对 SMMC 算法做了时间复杂度的分析。

在问题三中，首先 3(a)是一个大样本量的线性交叉的数据集合，针对其数据的特点我们引入了多流形聚类方法（mum-Cluster）方法。它直接从整个数据中找出流形交叠的部分并拆开不同的流形结构,从而构造出更忠实于流形结构的近邻图来实现混合流形聚类，通过 PID 参数调节策略，得出了理想的分类结果，在 3(b)运动分割问题中，我们引入了加权 SSC 算法来进行运动分割，将视频序列分成多个时空区域。这种以高斯相似度作为权重的算法很好的解决了 SSC 算法出现的局部子块分类不准确的问题，结果在全部 31 帧数据中都能准确的聚类数据，实现了运动分割的预期效果。分类结果为 1 到 138 个数据属于同一流形，为背景信息；139 到 214 个数据点属于同一流形，为公共汽车；215 到 297 个数据点属于同一流形，为小汽车。针对 3(c)中的人脸识别问题，我们首先采用了经典的 PCA 算法来进行启发式聚类，但是对于光照不同的特征提取不是很准确，导致了聚类结果的精度不高。因此，我们采用了 2DIMPCA 算法来进行了人脸识别，达到了完全精确的聚类效果。同时，我们也采用了 RPCA 先进行图像的预处理后，在采用 SSC 算法进行人脸识别，也可以达到同样的效果，分类结果为 1 到 5,11 到 15 属于一类，为外国人。6 到 10,16 到 20 属于一类，为中国人。我们对 4 种算法的识别效率和运行时间进行了比较，发现 2DIMPCA 算法可以又快又准处理不同光照下的人脸视频问题。

在问题四中，问题 4(a)是一个大样本的数据集，我们根据问题二的求解经验，选择采用了 SMMC 算法进行了启发式聚类，但是基本的 SMMC 算法定义的自然局部集合结构信息来辅助构造的相似性矩阵并不能满足圆台两侧的聚类要求，所以，我们通过改变 SMMC 算法的局部集合构造信息，从点的信息转变成小的子块的信息，从而使相似性矩阵能更好的反映数据的特点，使侧面的数据能够精确的关联，经过改进后的 SMMC 算法，在我们 PID 参数调试策略的指导下，可以较为精确的完成数据集的分类。问题 4(b)，我们根据数据的特点采用了 LSC 算法来进行聚类，其算法思想是：既考虑局部近邻信息又充分考虑流形结构数据所内含的额外的结构信息来指导近邻点的选取，以尽量地从同一个潜在流形上选取近邻点而不是整个欧氏空间。这样的流形结构在理论上可以达到我们的聚类的目的，从聚类结果上看，LSC 算法基本实现了聚类要求。

本文有以下的创新点，首先，我们引入了 SSC 算法去解决高维数据，然后，我们引入了 SMMC 算法去解决聚类问题，采用了 SSC+RPCA 算法和 2DIMPCA 解决了有噪声（光照不同）情况下的人脸识别问题，并对算法进行对比。引入了 mum-Cluster 算法去解决高密度有汇聚数据的聚类问题。更为重要的是，我们根据实际的经验结合算法理论提出了一套基于 PID 思想的参数调节策略，很好的解决了 SMMC 等算法的调节问题。针对 SMMC 所依赖的流形结构，我们采用了定义新的流形解决了自然流形无法聚类的问题，最后我们针对特殊的复杂的数据采用了 LSC 算法去定义一个近邻点的流形，可以有效的解决特殊的分类问题。

关键词：流形学习、稀疏子空间聚类(SSC)、多流形聚类方法(SMMC)、PID 策略、二维广义主成分分析(2DIMPCA)、局部与结构一致性方法(LSC)

目　录

一. 背景回顾

1.1 问题引出

在过去的几十年,随着人类社会的发展,电子计算机和各种数据采集工具 (如摄像头、传感器等)不断地得到普及并融入人们的日常生活中。随之而来的是,从多个数据源得到的多种形态的数据不断地成指数级的爆炸,人们已经能够在不分时间和地域的情况下,方便地获取各种数据和信息。如何对这些海量的观测数据进行压缩、存储、阅读、分析、处理,从它们中学习和发现某些内在的规律性,进而探讨隐藏在大千世界纷繁复杂的观察表象背后的事物本质,成为人们迫切想知道和亟需解决的问题。

随着世纪年代第一台电子计算机的诞生,经过几十年的不断积累和努力,电子计算机和各种数据采集工具(如摄像头、传感器等)不断地得到普及并扮演越来越重要的角色,人类社会逐渐进入信息时代。时至今日,人类社会已经做到了“不分时间和地域,可以方便地获得数据和信息”。随之而来的是,表征复杂模式的数据资源“正如在春季看到紫罗兰处处开放一样”爆炸性地增长(如网络数据、生物数据、图像数据和经济金融数据等),人们逐渐被数据和信息所 “淹没”。这些不断涌现的数据往往表现出数据量大、维数高、结构非线性化以及不为人的感知单独处理等主要特点,同时它们提供了所观察现象或未知事物的全面、完整、细致的信息,有利于揭示隐藏在事物和现象的纷繁复杂表象下的内在规律和事物本质。但是由于理论发展的滞后以及现实技术条件的限制(如计算机的处理能力、存储空间等),我们虽然被数据和信息所“淹没”,却缺乏足够的知识,对信息的利用能力没有得到显著提高。数据的膨胀给数据分析和处理带来了前所未有的困难和挑战:如何阅读和分析数据,从众多影响因素中快速有效地挖掘出隐含在其内部的本征信息和内在规律性,提取人们所需要的有价值的信息。为了能“不分时间和地域,可以有效地利用数据和信息”,首先需要对现实世界中获取的复杂海量数据进行分析和处理。当使用数学模型对数据资源进行描述时,往往可以用多变量组成的向量形式进行表示,在统计处理中通常称之为高维数据。现实世界中的海量高维数据根据人们能否获得先验信息可以分为两类,一类是可以通过人工判读或数据源获得部分先验标签信息(label)的数据,另一类是没有任何类别标签信息的数据。相应地,数据分析和处理的方法可以分为对应有部分标签数据的模式分类和对应无标签数据的聚类分析。

一个已经被证实的合理的数据分析思路就是流形学习(manifold learning),它从流形(manifold)的角度来重新把握数据的内部结构,通过对离散数据集合的分析来探求嵌入在高维数据空间中本征低维流形的不同表现形式,寻求事物产生的内在规律性,得到数据所蕴含的与人类认知一致的本征结构信息。从流形的观点来对数据进行学习和分析,并将其应用于模式分类和聚类分析任务中,实际上早已存在于人们的研究思路中。但是,更明确地指出这样的观点,并引发这一领域迅速发展的工作是 2000 年《science》上的连续三篇文章。首先,Seung 和 Lee[1]在神经生理学上的研究发现,整个神经细胞群的触发率可以由少量变量组成的函数来描述,如眼的角度和头的方向等,这隐含地表明神经元群体活动性是由其内在的低维结构所控制。从而,他们认为感知通常是以流形方式存在的。随后,Tenenbaum 等人[2]以及 Roweis 和 Saul[3]分别提出了等距特征映射(Iosmetric Feature Map,Isomap)和局部线性嵌入(Locally Linear Embedding,LLE)两个不同的方法来具体处理流形数据,揭示数据内在的低维流形结构,并将其成功地应用于脸谱图像数据和文档类

数据的处理中。

1.2 流形学习简述

流形学习(manifold learning)的基本前提假设是高维观测数据位于或近似位于低维空间的一个可以被人类所感知的线性或非线性的流形(manifold)上,其目的是要挖掘数据的内在规律性和数据的分布形式,从观测的现象中寻找事物的本质信息和内在规律性。

所谓流形,通俗地说,就是各种维数的曲线或曲面等几何对象的总称。它来源于黎曼几何中“流形”的概念,被定义为一个满足局部欧几里得属性的拓扑空间。其严格的数学定义如下:

定义 1.1(陈维桓[4]):设 χ 是一个 Hausdorff 空间,若对任意一点 $x \in \chi$,都有 x 在 χ 中的一个邻域 U 同胚 d 于维欧氏空间的一个开集,则称 χ 是一个 d 维流形(或拓扑流形)。

流形学习假设高维观测数据采样于一个潜在的低维流形上,通过某种显示或隐式的映射关系学习出此假设存在的流形并将原始数据从周围观测空间(ambient space)投影到一个低维嵌入空间(embedding space),在这个空间内保持原始数据的某些全局或局部的几何属性和内在结构。流形学习是一种从一组观测数据中推导其产生式模型的过程,可以形式化的描述为:

定义 1.2:给定一组高维观测数据 $\chi = \left\{ \boldsymbol{x}_i \in \Re^D, i = 1, \ldots, N \right\}$ (其中 D 为观测数据的维数, N 为样本容量),假设其来自于(或近似来自于)某一本征维数为 d ($d < D$,并且通常 $d \ll D$)的低维光滑流形上 $\Omega^d \subseteq \Re^d$。流形学习的目标就是探求这些高维数据 $\boldsymbol{x}_i$ 的低维嵌入表示 $\boldsymbol{y}_i \in \Re^D$,即寻找一个从观测空间到低维嵌入空间的嵌入映射 f,使得 $\boldsymbol{y}_i = f\left(\boldsymbol{x}_i\right)$,以及其一对一的重构映射 f^{-1},使得 $\boldsymbol{x}_i = f^{-1}\left(\boldsymbol{y}_i\right)$。同时,这一映射函数 f 应该满足某些限制条件以尽量地保持原始流形数据的全局或局部几何结构关系。

1.3 流形学习方法的分类

从方法论上讲,流形学习方法可以根据不同的准则进行不同的分类[5]:

1.根据流形的结构特征

(1)线性方法:线性方法用于对线性流形的学习,其映射函数是线性函数并且通常具有显式的表达形式,例如:主成分分析(PCA)和多维尺度变换 (MDS)等;

(2)非线性方法:当流形具有非线性的结构特征时,通常采用非线性学习方法。这时映射函数 f 是非线性映射,但通常没有显式的表达形式,而仅仅通过隐含映射的方式给出原始高维数据的低维嵌入表示,例如:等距特征映射(Iosmap)和局部线性嵌入(LLE)等。

2.根据模型的结构特征

(1)全局方法:全局方法通常采用全体数据来建模,关注并描述数据的整体特征,例如:PCA 采用全体数据的协方差来刻画数据不同维数之间的统计相关性和整体差异性;

(2)局部方法:由于流形在每一点的局部近邻和欧氏空间的一个开集同胚,局部方法通常采用局部近邻数据来建模,关注并描述数据的局部结构特征,例如:LLE采用局部 K 近邻或 ε 近邻数据来刻画数据之间的局部线性重构关系;

(3)局部模型的全局排列方法:局部模型的全局排列方法结合了全局方法和局部方法的思想,它们基于全局非线性结构在局部是线性的假设,先进行局部线性模型的拟合再进行全局坐标的对齐。例如:全局协调方法[6](Global Coordination)将全局非线性流形分割为一些小的局部线性子块,然后通过一定的策略将这些子块全局排列在一起,得到数据的一致的低维流形表示。

3.根据优化目标函数的性质

(1)基于凸目标函数(convex)的方法:凸目标函数的优化具有全局最优解,不会陷入局部极值,例如:LLE 方法;

(2)基于非凸目标函数(nonconvex)的方法:非凸目标函数的优化过程则会陷入局部极值,例如:概率主成分分析方法[7](Probabilistic Principal Component Analysis,PPCA)和上述基于局部模型的全局排列方法。

1.4 稀疏子空间聚类与低秩子空间聚类

子空间聚类，又称为子空间分割，假设数据分布于若干个低维子空间的并，是将数据按某种方式分类到其所属的子空间的过程。通过子空间聚类，可以将来自同一子空间中的数据归为一类，由同类数据又可以提取对应子空间的相关性质。假设数据矩阵为 $\boldsymbol{X}=[\boldsymbol{x_1},\boldsymbol{x_2},\ldots,\boldsymbol{x_N}]\in\boldsymbol{R}^{M\times N}$，来自 n 个不同线性子空间的并 $S=\bigcup_{i=1}^{n}S_i$，$(n\geq 1)$，子空间 S_i，$(i=1,\ldots,n)$ 的维数为 $0<d_i<M$，对应的基矩阵为 $\boldsymbol{B}_i=\left[\boldsymbol{b}_{1_i},\boldsymbol{b}_{2_i},\ldots,\boldsymbol{b}_{d_i}\right]\in\boldsymbol{R}^{M\times d}$，则子空间 S_i 可以表示为

$$S_i=\left\{\boldsymbol{y}\in\boldsymbol{R}^M:\boldsymbol{y}=\boldsymbol{B}_i\boldsymbol{a}=\sum_{j=1_i}^{d_i}\boldsymbol{b}_j a_j\right\},i=1,\ldots,n \qquad (1\text{-}1)$$

其中，$\boldsymbol{a}\in\boldsymbol{R}^{d_i}$ 为 $\boldsymbol{y}$ 在低维子空间中的表示系数。

子空间聚类就是将数据矩阵 $\boldsymbol{X}$ 中属于子空间 S_i 的数据分为一类，得到数据的低维表示，并由此得到子空间 S_i 的维数、基矩阵。

基于谱聚类的方法在近几年较为流行，这类方法首先定义一个关于样本点相互关系的图，然后利用 Normalized Cut[8]等谱聚类方法（其输入是一个反应样本关系的相似度矩阵，矩阵的第 i 行 j 列的数值越大说明第 i 个样本和第 j 个样本的关系越密切，如果能将同类样本的相似度构造的较大，不同类的较小，这类方法一般都能得到正确的分类结果）得到分割结果。代表性的基于谱聚类的子空间分割方法包括低秩表示[9]和稀疏表示[10]等，下面对这两种方法的做个简单介绍。

稀疏子空间聚类（SSC）

稀疏子空间聚类方法，是对子空间表示系数进行稀疏约束的一类子空间聚类方法。子空间聚类的最终结果是将同一子空间的数据归为一类。在子空间相互独立的情况下，属于某一子空间的数据只由这个子空间的基的线性组合生成，而在其他子空间中的表示系数为零。这样高维数据的表示系数就具有稀疏的特性。同一子空间中的数据，因为都仅在这一子空间中有非零的表示系数，表现为相同的稀疏特性，通过对表示系数稀疏约束的求解，突出了数据表示系数的这种稀疏特性，进而为数据的正确聚类提供支持。

低秩子空间聚类（LRR）

通过对子空间表示系数矩阵的研究，有些学者在求解子空间表示系数矩阵时，引入核范数(一个矩阵的核范数是指矩阵的所有奇异值的加和)约束，希望通过系数矩阵的低秩要求得到更好的数据的子空间表示。文章[9]给出了低秩表示模型的闭解且理论上保证了当子空间独立且数据采样充分的情况时，低秩表示可以得到块对角的解。这个结论基本保证了低秩表示方法在解决独立子空间分割问题的有效性。

二．问题一的建模与求解

2.1 问题重述

当子空间独立时，子空间聚类问题相对容易。附件一中 1.mat 中有一组高维数据（.mat 所存矩阵的每列为一个数据点，以下各题均如此），它采样于两个独立的子空间。请将该组数据分成两类。

问题一中数据为200×100数据，为 200 个 100 维的数据，它采样于两个独立的子空间，数据特点为数据量较小，维数较高，为典型的数据采用子空间聚类人容易，采用最基本的流行学习的方法即可进行处理，采用 SSC 算法求解该问题。下面简单介绍稀疏子空间聚类（SSC）和 K-means 聚类方法。

2.2 模型建构与方法分析

2.2.1 稀疏子空间聚类

2009 年, Elhamifar 等基于一维稀疏性提出了稀疏子空间聚类(Sparse subspace clustering, SSC) 方法, SSC 模型考虑线性子空间相互独立的情形，设数据矩阵为 $\boldsymbol{X}=[\boldsymbol{x_1},\boldsymbol{x_2},\ldots,\boldsymbol{x}_{\mathrm{N}}]=\left[\boldsymbol{X}_1,\boldsymbol{X}_2,\ldots\boldsymbol{X}_N\right]\Gamma\in\boldsymbol{R}^{M\times N}$, 其中 $\boldsymbol{X}_i\in\boldsymbol{R}^{M\times n_i}$ 来自子空间 S_i，$\sum_{i=1}^{n}n_i=N,i=1,\ldots,n$。$\Gamma\in\boldsymbol{R}^{N\times N}$ 是列向量的一个排列，即数据矩阵 $\boldsymbol{X}$ 通过特定的列变换可以将同一子空间的数据排列在一块。 $\boldsymbol{X}$ 的列向量 $\boldsymbol{x}$ 的稀疏子空间聚类表示为

$$\min_{a}\|\boldsymbol{a}\|_1 \quad s.t.\ \boldsymbol{x}=\tilde{\boldsymbol{X}}a\ , \tag{2-1}$$

其中， $\tilde{\boldsymbol{X}}$ 为 $\boldsymbol{X}$ 为中去掉列向量 $\boldsymbol{x}$ 的矩阵。式（2-1）如有下定理成立。

定理 2.1 设 $\boldsymbol{X}\in\boldsymbol{R}^{M\times N}$ 的列向量来自 n 个线性子空间 S_i 的并， S_i 之间相互独立且 $\boldsymbol{X}=\left[\boldsymbol{X}_1,\boldsymbol{X}_2,\ldots\boldsymbol{X}_N\right]\Gamma\in\boldsymbol{R}^{M\times N}$。令 $\boldsymbol{x}$ 为第 i 个子空间中的点，则式（2-1）的解

$$\boldsymbol{a}=\Gamma^{-1}\left[\boldsymbol{a}_1^T,\boldsymbol{a}_2^T,\ldots\boldsymbol{a}_n^T\right]^T\in\boldsymbol{R}^N \tag{2-2}$$

具有块稀疏结构，即 $\boldsymbol{a}_i\neq0$ 且 $\boldsymbol{a}_j=0$， $j\neq i$。当 Γ 为单位阵时，由式（2-1）得到的系数矩阵具有块稀疏的结构，这是分块对角阵在稀疏约束下表现出来的性质。

式（2-1）是一个 l_1 优化问题，可利用凸规划方法求解，SSC 模型利用 Lasso 方法求解数据的子空间表示系数。对于式（2-1），为了避免出现平凡解，实际计算时，字典 $\boldsymbol{X}$ 中将去掉当前计算的列向量 $\boldsymbol{x}$，对应系数取零，反映到系数矩阵则

为系数矩阵的对角线元素为零。将模型写为矩阵形式则得到

$$\min_{A}\|\boldsymbol{A}\|_i \quad s.t.\ \boldsymbol{X}=\boldsymbol{X}\boldsymbol{A}, diag\left(\boldsymbol{A}\right)=0, \tag{2-3}$$

其中$\|\boldsymbol{A}\|_1=\sum_i\sum_j\left|A_{i,j}\right|$ 通过求解上述模型，得到 $\boldsymbol{X}$ 的子空间表示系数矩阵 $\boldsymbol{A}$，由此可以构造图G的邻接矩阵$\boldsymbol{W}$ 。由于无向图的邻接矩阵具有对称性，故邻接矩阵为

$$\boldsymbol{W}=\left|\boldsymbol{A}\right|+\left|\boldsymbol{A}^T\right|, \tag{2-4}$$

其中，$\left|\boldsymbol{A}\right|$表示 $\boldsymbol{A}$ 中每个元素取绝对值。邻接矩阵$\boldsymbol{W}$ 即可得图的拉普拉斯矩阵 $\boldsymbol{L}$，再利用 K-means 算法可以得到最终的聚类结果。综上，SSC 模型的算法如下表 2-1：

表 2-1 稀疏子空间聚类算法

算法 1：稀疏子空间聚类（SSC）
输入：数据矩阵 $\boldsymbol{X}$，子空间个数 n 1. 对数据矩阵每个列向量建立式 (2-1)，并利用 Lasso 方法求解； 2. 构造无向赋权图G ，并根据式 (2-4)构造图的邻接矩阵； 3. 由图的邻接矩阵构造图的拉普拉斯矩阵L ，并利用 K-means 算法得到子空间聚类结果。

文献[11]中还讨论了仿射子空间情形下的子空间聚类问题，相对于模型(2-3)，稀疏的仿射子空间聚类模型，增加了每一列系数之和为 1 的约束，其余的表示系数求解和谱聚类过程没有本质差别。

为了适应数据中含有噪声和野点的情况，[11]对模型 (2-3)进行了改进，增加噪声项和野点项的约束，以减少噪声和野点对表示系数的影响。模型的矩阵形式可表示为

$$\begin{aligned}&\min_{\boldsymbol{A},\boldsymbol{E}_1}\|\boldsymbol{A}\|_1+\|\boldsymbol{E}_1\|_1+\lambda\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{A}-\boldsymbol{E}_1\|_F\\&s.t.\ \boldsymbol{X}=\boldsymbol{X}\boldsymbol{A}+\boldsymbol{E}_1+\boldsymbol{E}_2, diag\left(\boldsymbol{A}\right)=0\end{aligned} \tag{2-5}$$

其中，$\boldsymbol{E}_1$表示野点产生的误差，$\boldsymbol{E}_2$表示噪声，$\lambda>0$这里假设野点的误差也是稀疏的，噪声的误差用 F-范数度量。

SSC 模型利用l_1范数度量系数的稀疏性，而稀疏表示理论最早是利用l_0范数作为数据稀疏性的度量。l_0范数约束下的稀疏表示问题的求解往往是 N-P 难的，而在一定的条件下，可以用l_1范数约束近似l_0范数约束。故在许多稀疏表示的场合，利用l_1范数进行稀疏性的度量。

2.2.2 K-means 聚类算法

K-means 聚类是著名的划分聚类算法，由于简洁和效率成为所有聚类算法中最广泛使用的聚类算法。

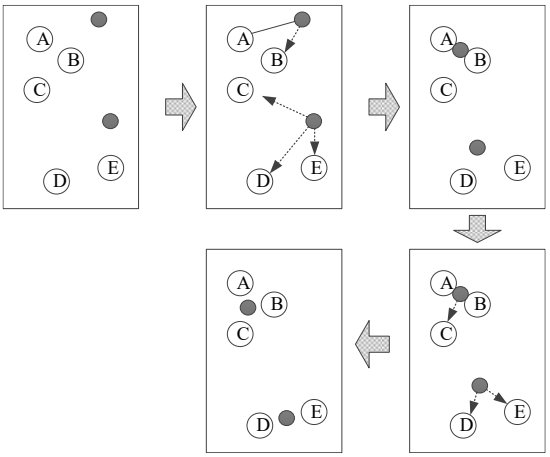


图 2-1 K-means 聚类示意图

在 K-means 聚类算法中，用一个聚类中心来代表一个簇。算法基本过程是先生成 K 个初始聚类中心，并把样本集中的样本按照最小距离原则分配到最近的聚类核心中去。然后计算每个聚类中的样本均值作为新的聚类中心。重复该过程直到聚类中心不再发生变化。具体的聚类算法描述如下表 2：

表 2-2 K-means 算法

算法 2：K-means 算法	
1	随机生成 K 聚类中心, $\left(C_1, C_2, \ldots, C_k\right)$
2	对于每一个样本点
3	计算距离 with $d\left(x_i, x_j\right)=\sqrt{\sum_{k=1}^{d}\left(x_{ik}-x_{jk}\right)^2}$
4	根据 $\sum_{i=1}^{n}\min_{j\in\{1,\ldots,k\}}\left\|x_i-C_j\right\|^2$ 将每一个点分配到聚类中心
5	如果$\left(C_1, C_2, \ldots, C_k\right)$改变转到 4

2.3 问题求解

本题目中需要分类的数据位 200 个 100 维的数据点，属于典型的高维聚类问题，对于这样的问题，采用通常的 PCA 降维的方法，其效果往往不是非常的理想，往往很难达到精确分类的效果，下面是降低到二维时的数据分布图。

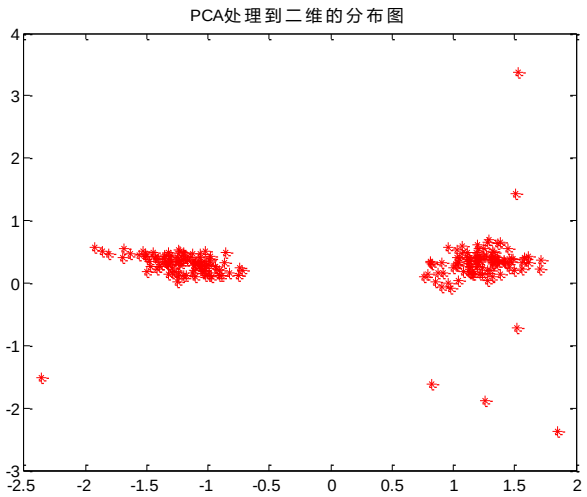


图 2-2 PCA 处理得到的二维分布图

我们从图 2-2 可以发现，PCA 提取特征后可以将数据点分为两类，效果也在合理的范围内，但是针对这样的一个高维度的问题，我们可以采用提取高维空间中的子空间来进行分类，这样可以达到更加精确的分类效果。

稀疏子空间聚类是一类新型的高效的分类方法，其突出了稀疏表示的优势，又引入了流形学习的思想，这种基于谱聚类的子空间聚类方法，假设高维空间中数据本质上属于某个低维子空间，能够在低维子空间中进行线性表示，反过来，高维度数据的低维表示能够揭示数据所在的本质子空间，有利于数据聚类，从上面的 PCA 特征降低维度后的结果可以看出，在本次的数据集种可以有效的寻找到这样的一个低维子空间，所以采用 SSC（稀疏子空间聚类）的方法可以有效的解决本题中的分类问题。但是设定合适的参数也是一个非常重要的步骤，我们首先采用默认的正则化系数来进行聚类，此时 lambda = 0.05，其分类结果如图 2-3：

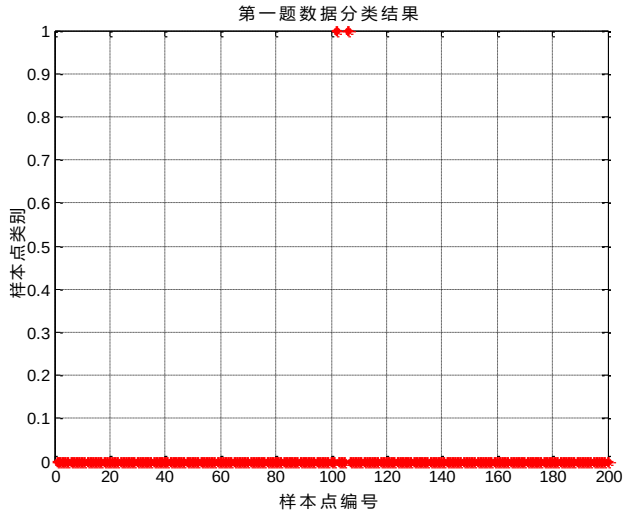


图 2-3 问题一 lambda=0.05 SSC 分类结果

显然这样的分类的结果达不到要求，经过测试后，我们同样采用 Lasso 最优准则，正则化系数选择设定为 0.01，在宽松系数约束，不限制维度的情况下进行处理，输入数据后进行子空间表示得到相应的相似度矩阵，其特征值分布矩阵如下图 2-4 所示：

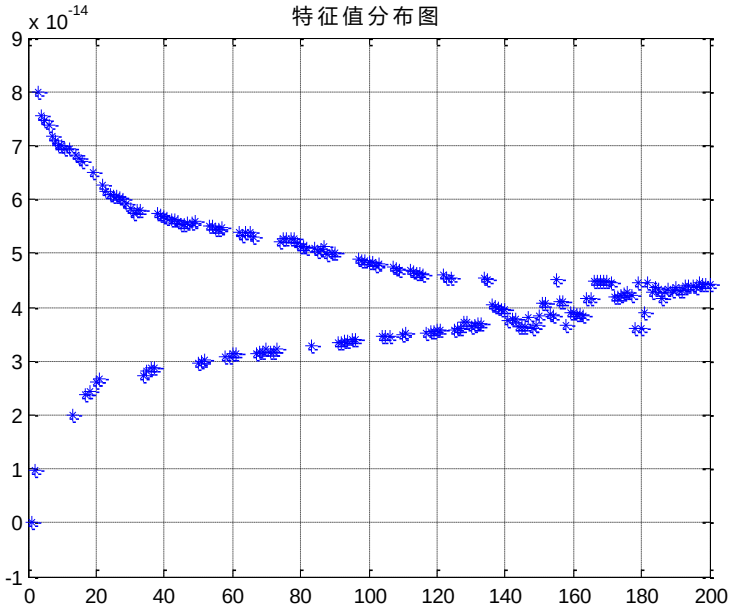


图 2-4 问题一特征值分布

得到相似度矩阵后，我们后面采用 k-means 这样的经典聚类算法来进行聚类，聚成两类，选取特征矩阵中最小的两个特征值所对应的特征向量来进行聚类处理，使用数字 0 和 1 分别代表不同的类别，其分类的结果如图 2-5 所示：

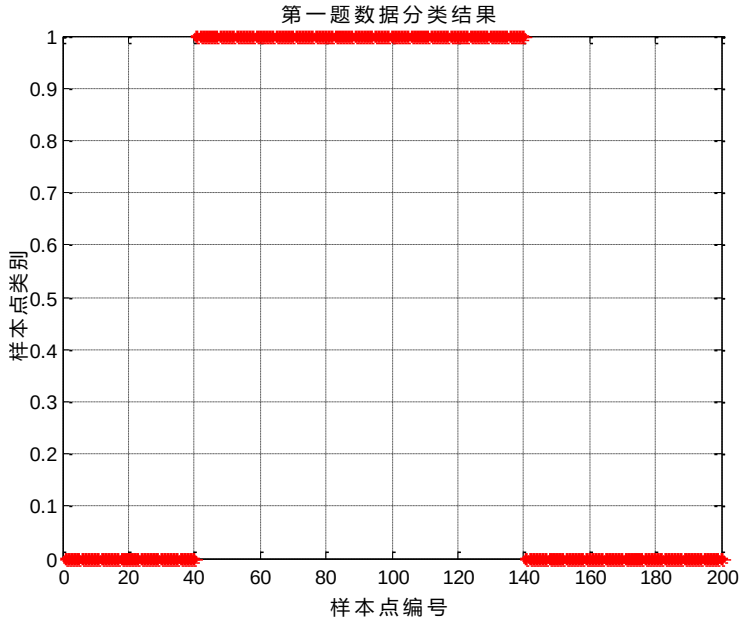


图 2-5 问题一最终 SSC 分类结果

从上面的分别结果我们可以看出，两类的数量之比值为 100:100，非常好的完成了这样的一个高维分布问题.从上图中可以看出前 40 和后 60 个数据为一组，中间 100 一个数据为一组。可以看出采用 SSC+K-means 成功将其分类。SSC 这样的一种算法很好的完成了分类的任务，这样的结果显示，我们可以看出，针对本类问题的数据特点应该选择较小的正则化系数，这样的设定可以直观的理解为是数据的特征突出，也就实现了更好分类数据的目的啊。为了进行鲁棒性分析，我们编程设计进行了 100 次蒙特卡洛实验，其结果显示在我们严格推倒出的参数

的情况下，SSC 算法对于此类问题有很好的鲁棒性。
根据题目要求给出标签表如表 2-3 所示。

表 2-3 问题结果标签表

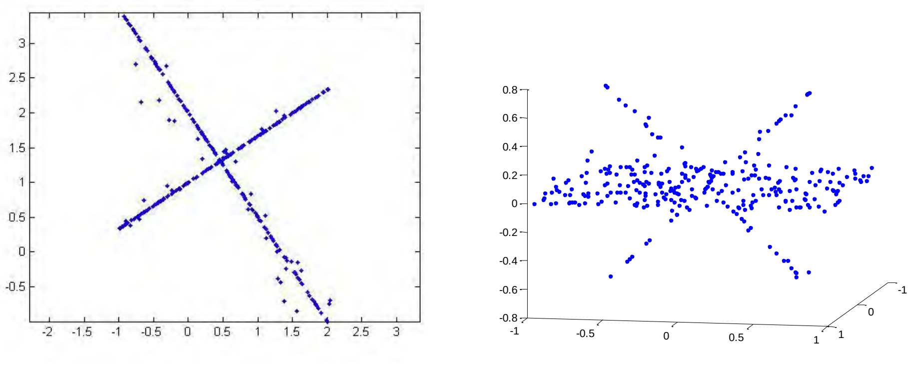
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表格中标号为 0 和 1 分别为两组数据，如表格所示，综上所述，SSC 算法+K-means 算法可以完成本题中的分类任务，也反映出了流形学习聚类算法在分类此类问题中特有的优势。

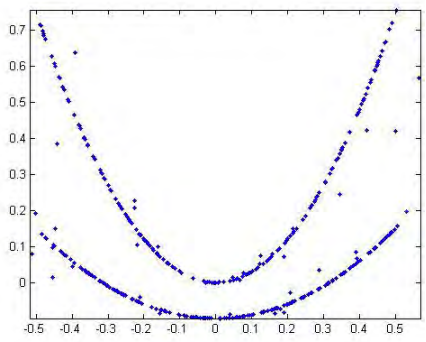
三. 问题二的建模与求解

3.1 问题重述

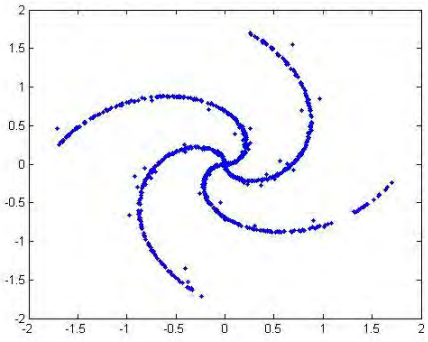
请处理附件二中四个低维空间中的子空间聚类问题和多流形聚类问题，如图 1 所示。图 3-1(a)为两条交点不在原点且互相垂直的两条直线，请将其分为两类；图 3-1(b)为一个平面和两条直线，这是一个不满足独立子空间的关系的例子，请将其分为三类。图 3-1(c)为两条不相交的二次曲线，请将其分为两类。图 3-1(d)为两条相交的螺旋线，请将其分为两类。



(a) (b)



(c)



(d)

图 3-1 问题二原始图

分析题目：

（a）为两条相互垂直不过原点的直线，2×340 数据，为 340 个 2 维数据，两条直线垂直，因此其中的点来自两个独立子空间，但是不过原点，数据不满足稀疏特性，同时为低维数据，此问题为独立子空间聚类问题；

（b）为两条直线和一个平面，3×300 数据，为 300 个 3 维数据，数据来自三个不独立的子空间，数据过原点，所以具有稀疏特性，也为低维数据，此问题为不独立子空间的聚类问题；

（c）为两条不相交曲线，2×400 数据，为 400 个 2 维数据，两条曲线分别属于两个流形，曲线不想交，两个流形相对独立，为独立多流形聚类问题；

（d）为两条相交曲线，2×988 数据，为 988 个 2 维数据，两条曲线分别属于两个流形，但曲线相交，两个流形不独立，为不独立多流形聚类问题。

我们选取一个普适的方法来解决问题二，使其既能解决子空间聚类问题，也能解决多流形聚类问题，无论是否独立。由于 SMMC 用来检测或分组数据中的低维流形结构十分有效。决定采用 SMMC 方法来进行，下面对 SMMC 方法进行介绍并提供了一套有效的参数调试方法。

3.2 多流形聚类方法（SMMC）

多流形聚类方法属于混合流形聚类中的一种，简单介绍混合流形聚类。

线性流形聚类方法由于其自身的线性特性能够很好地对具有线性结构的数据进行分组,但却不能很好地分组具有非线性结构的数据。现有的非线性流形聚类方法能在一定程度上很好地分组非线性结构数据,但却局限于处理良分离的非线性结构或相互交叠的非线性结构。一个自然的理论问题和现实需求是如何对更一般情况下的数据进行分组,即数据中既包括线性结构又包括非线性结构、既有良分离的结构又有相互交叠的结构。这就是混合流形聚类问题。

完整的混合流形聚类问题可以形式地描述为[12]:

定义 3.1：给定来自于 k 个不同潜在流形(线性/非线性) $\Omega_j \subseteq \mathbb{R}^D (j=1,\ldots,k,$ 第 j 个流形的本征维数为 d_j，$0<d_j<D$ 且可能互不相同)的共 N 个高维数据采样 $\chi=\{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=1}^N$。它们是无组织的,即不知道哪个采样点位于哪个潜在流形上。此外,其中某些流形是相互良分离的,而某些流形是相互交叠的。混合流形聚类的目的是:

1.确定潜在流形的数目及其本征维数 $d_j, j=1,\ldots,k$;

2.将给定的数据采样划分到其所属的潜在流形。

多流形聚类方法(Spectral Multi-Manifold Clustering,简记为 SMMC)来实现混合流形聚类。它的基本思想是:从相似性矩阵的角度出发,充分利用流形采样点所内含的自然的局部几何结构信息来辅助构造更合适的相似性矩阵并进而发现正确的流形聚类。

3.2.1 相似性矩阵

我们的相似性矩阵构造基于下述事实:

(1)尽管数据在全局上位于或近似位于光滑的非线性流形上,局部地,每个数据点和它的近邻点位于流形的一个局部线性块上[3,14];

(2)每个数据点的局部切空间提供了非线性流形局部几何结构的优良低维线性近似[15];

(3)在不同流形聚类的相交区域,来自于同一个流形聚类的数据点有相似的局部切空间而来自不同流形聚类的数据点其切空间是不同的。因此,我们可以利用数据点所内含的局部几何结构信息来辅助构造更合适的相似性矩阵 $\boldsymbol{W}$.

值得指出的是,只有当下面的两个条件同时满足时,我们才能够断定两个数据点是来自同一个流形聚类的:它们相互靠近同时具有相似的局部切空间。两个反例是:

(1)数据点和垂直的仿射子空间上的数据点有相似的局部切空间但它们相互远离;

(2)曲面和垂直仿射子空间的相交区域附近的点相互靠近但它们有不同的局部切空间。

因此,我们在构造相似性矩阵时,既要考虑数据点之间的欧氏距离关系 $q_{ij}=q\left(\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|\right)$ (称为局部相似性 local similarity),),又要考虑数据点局部切空间之间的相似性 p_{ij} (称为结构相似性,structural similarity)。这两个相似性融合在一起来决定最后的相似性权值:

$$w_{ij}=f\left(p_{ij},q_{ij}\right) \tag{3-1}$$

其中是 f 一个合适的融合函数。为了使得构造出的相似性矩阵具有前面分析中所期望的性质, f 应该是关于数据点间欧氏距离的一个单调递减函数同时是局部切空间之间相似性的单调递增函数。

下面给出方法中所采用的函数 p,q 和 f 的具体形式。

假设数据点 $\boldsymbol{x}_i\left(1,\ldots,N\right)$ 处的局部切空间为 Θ_i ,则两个数据点 x_i 和 x_j 的局部切空间之间的结构相似性可以定义为:

$$p_{ij}=p\left(\Theta_i,\Theta_j\right)=\left(\prod_{l=1}^{d}\cos\left(\theta_l\right)\right)^{o} \tag{3-2}$$

在（3-2）中， $o\in N^{+}$ 是一个可调参数。 $0\le\theta_1\le,\ldots,\le\theta_d\le\pi/2$ 是两个切空间 Θ_i 和 Θ_j 之间的主角度[16],递归地定义为:

$$\cos\left(\theta_1\right)=\max_{\substack{u_1\in\Theta_i,v_1\in\Theta_j\\ \|u_1\|=\|v_1\|=1}}\boldsymbol{u}_1^T\boldsymbol{v}_1\ ) \tag{3-3}$$

$$\cos(\theta_l)=\max_{\substack{u_l\in\Theta_i,v_l\in\Theta_j\\ \|u_l\|=\|v_l\|=1}} \boldsymbol{u}_l^T\boldsymbol{v}_l \quad l=2,\ldots,d \tag{3-4}$$

其中， $\boldsymbol{u}_l^T\boldsymbol{u}_1=0,\boldsymbol{v}_l^T\boldsymbol{v}_1=0,i=1,\ldots,l-1$

数据点 x_i 和 x_j 之间的局部相似性简单地定义为：

$$q_{ij}=\begin{cases}1, if \quad \boldsymbol{x}_i\in Knn(\boldsymbol{x}_j) \quad or \quad \boldsymbol{x}_j\in Knn(\boldsymbol{x}_i)\\ 0, otherwise\end{cases} \tag{3-5}$$

其中 $Knn(x)$ 代表 x 的 K 个近邻数据点。换句话说,局部相似性要求我们在构造近邻图时采用 K -近邻图而不能采用完全图将所有的数据点都通过边连接起来。最后函数将这两个函数 p 和 q 简单的乘在一起得到相似性权值:

$$w_{ij}=p_{ij}q_{ij}=\begin{cases}\left(\prod_{l=1}^{d}\cos(\theta_l)\right)^{o}, if \quad \boldsymbol{x}_i\in Knn(\boldsymbol{x}_j) \quad or \quad \boldsymbol{x}_j\in Knn(\boldsymbol{x}_i)\\ 0 \qquad\qquad\qquad , otherwise\end{cases} \tag{3-6}$$

容易验证,式(3-6)中定义的相似性权值具有所期望的性质,即来自不同聚类流形的数据点之间有相对低的权值。其原因是:（1）当来自不同流形的两个数据点相互远离时,根据(3-6)其相似性权值为0；（2）当来自不同流形的两个数据点靠近流形的相交区域时,它们有不同的局部切空间结构,因此当调节参数 o 足够大时,也可以使得相似性权值相对低。因此,当谱方法应用于上述定义的相似性矩阵 $\boldsymbol{W}$ 时,可以期望得到更好的性能。

上述过程中我们假设每个数据点的局部切空间是已知的,一个未解决的问题是如何有效地近似或估计这些局部切空间，我们将在下一小节中详细讨论这个问题。

3.2.2 局部切空间

传统上,每个数据点的局部切空间可以通过给定样本点附近的局部近邻点来估计[15.17]。具体地说,给定样本点 $\boldsymbol{x}$ 和它在欧氏空间度量下的 n 个近邻点 $N(\boldsymbol{x})=\{\boldsymbol{x}^1,\ldots,\boldsymbol{x}^n\}$，$\boldsymbol{x}$ 附近的局部几何信息内含在该点处的局部采样协方差矩阵 $\sum_{\boldsymbol{x}}$ 中：

$$\sum\nolimits_x=\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}^i-\boldsymbol{u}_x)(\boldsymbol{x}^i-\boldsymbol{u}_x)^T \tag{3-7}$$

其中 $\boldsymbol{u}_x=1/n\sum_{i=1}^{n}\boldsymbol{x}^i$.

样本点 x 处的局部切空间 Θ_x 由 d 的最大 d 个奇异值 $\sum_{\boldsymbol{x}}$ 对应的左奇异向量给出。即,假设 $\sum_{\boldsymbol{x}}$ 的奇异值分解为:

$$\sum\nolimits_x=\begin{bmatrix}\boldsymbol{U}_d & \tilde{\boldsymbol{U}}_d\end{bmatrix}\begin{bmatrix}\Sigma_d & 0\\ 0 & \tilde{\Sigma}_d\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_d & \boldsymbol{V}_d\end{bmatrix}^T \tag{3-8}$$

其中 $\begin{bmatrix}\boldsymbol{U}_d & \tilde{\boldsymbol{U}}_d\end{bmatrix}\in\Re^{D\times D}$ 是正交矩阵并且 $\boldsymbol{U}_d\in\Re^{D\times d}$，则有：

$$\Theta_x=span(\boldsymbol{U}_d) \tag{3-9}$$

不幸的是,当两个数据点 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 非常靠近时,即使他们来自于不同的流形,根据估计出的局部切空间 Θ_x 和 Θ_y 也非常相似。其原因在于,在这种情况下 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 的基

于欧氏距离度量的局部近邻 $N(\boldsymbol{x})$ 和 $N(\boldsymbol{y})$ 会严重地交叠在一起,从而导致了相似的局部协方差矩阵 $\sum_x$ 和 $\sum_y$。因此,这种传统的局部切空间估计方法不能用于混合流形建模。下面,我们将给出一个快速有效的方法来逼近每个数据点附近的局部切空间。

我们的基本思想基于如下事实:

(1)全局非线性流形在局部能被一系列局部线性流形很好的逼近[3,14];

(2)主成分分析器[13]可以有效地穿过相交线性流形;

(3)被同一个线性分析器逼近的数据点通常具有相似的局部切空间并且这些切空间可以被局部分析器的主子空间很好地近似。

因此,我们可以训练一系列局部线性分析器来逼近潜在的流形,然后估计每个给定数据点的局部切空间为其相应局部分析器的主子空间。

具体地说,我们训练 M 个混合概率主成分分析器[13] (Mixture of Probabilistic Principle Component Analyzers,MPPCA)来估计局部切空间,其中每个分析器由模型参数 $\theta_m=\{\boldsymbol{\mu}_m,\boldsymbol{V}_m,\sigma_m^2\}, m=1,\ldots,M$ 刻画,其中 $\boldsymbol{\mu}_m\in\Re^D$ $\boldsymbol{V}_m\in\Re^{D\times d}$,而 σ_m^2 是一个标量。需要指出的是, M 是用于逼近所有潜在的线性或非线性流形的局部线性子模型的个数。在第 m 个分析器模型下,一个 D 维的观测数据向量 $\boldsymbol{x}$ 通过下式对应一个 d 相应的维潜在向量 $\boldsymbol{y}$:

$$x=\boldsymbol{V}_m y+\boldsymbol{\mu}_m+\varepsilon_m \tag{3-10}$$

其中 $\boldsymbol{\mu}_m$ 是数据的均值向量,潜在变量 $\boldsymbol{y}$ 和噪声 ε_m 分别是高斯分布 $y\sim \mathrm{N}(0,I)$ 和 $\varepsilon_m\sim \mathrm{N}(0,\sigma_m^2 I)$。在此模型下,的边际分布为:

$$p(\boldsymbol{x}\mid m)=(2\pi)^{-D/2}\left|\boldsymbol{C}_m\right|^{-1/2}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_m)^T\boldsymbol{C}_m^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_m)\right\} \tag{3-11}$$

其中模型协方差为:

$$\boldsymbol{C}_m=\sigma_m^2\boldsymbol{I}+\boldsymbol{V}_m\boldsymbol{V}_m^T \tag{3-12}$$

模型参数 $\boldsymbol{\mu}_m$, $\boldsymbol{V}_m$, σ_m^2 可以通过利用 EM 算法最大化观测数据 $X=\{\boldsymbol{x}_i,i=1,\ldots,N\}$ 的对数似然来得到:

$$L=\sum_{i=1}^{N}\ln\left\{\sum_{m=1}^{M}\pi_m p(\boldsymbol{x}_i\mid m)\right\} \tag{3-13}$$

其中 π_m 是混合比例,满足条件 $\pi_m>0$ 和 $\sum_{m=1}^{M}\pi_m$。具体地说,学习的主要过程为:

E-step 利用当前模型参数 $\theta_m=\{\boldsymbol{\mu}_m,\boldsymbol{V}_m,\sigma_m^2\}$ 计算:

$$R_{im}=\frac{\pi_m p(\boldsymbol{x}_i\mid m)}{\sum_{m=1}^{M}\pi_m p(\boldsymbol{x}_i\mid m)} \tag{3-14}$$

$$\pi_m^{new}=\frac{1}{N}\sum_{i=1}^{N}R_{im} \tag{3-15}$$

$$\mu_m^{new}=\frac{\sum_{i=1}^{N}R_{im}\boldsymbol{x}_i}{\sum_{i=1}^{N}R_{im}} \tag{3-16}$$

M-step 重新估计参数 $\boldsymbol{V}_m$ 和 σ_m^2 为

$$\boldsymbol{V}_m^{new}=\boldsymbol{S}_m\boldsymbol{V}_m\left(\sigma_m^2\boldsymbol{I}+\boldsymbol{T}_m^{-1}\boldsymbol{V}_m^T\boldsymbol{S}_m\boldsymbol{V}_m\right)^{-1} \tag{3-17}$$

$$\left(\sigma_m^2\right)^{new}=\frac{1}{d}tr\left[\boldsymbol{S}_m-\boldsymbol{S}_m\boldsymbol{V}_m\boldsymbol{T}_m^{-1}\left(\boldsymbol{V}_m^{new}\right)^T\right] \tag{3-18}$$

其中

$$\boldsymbol{S}_m=\frac{1}{\pi_m^{new}N}\sum_{i=1}^{N}R_{im}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_m^{new}\right)\left(\boldsymbol{x}_i-\boldsymbol{\mu}_m^{new}\right)^T \tag{3-19}$$

$$\boldsymbol{T}_m=\sigma_m^2\boldsymbol{I}+\boldsymbol{V}_m^T\boldsymbol{V}_m \tag{3-20}$$

本文中采用 K-means 来初始化上述 EM 学习过程。最后,样本点 $\boldsymbol{x}_i$ 根据下述关系分组到第 j 个局部分析器:

$$p\left(\boldsymbol{x}_i\mid j\right)=\max_m p\left(\boldsymbol{x}_i\mid m\right) \tag{3-21}$$

同时其局部切空间由下式给出:

$$\Theta_i=span\left(\boldsymbol{V}_j\right) \tag{3-22}$$

利用 M 个局部线性分析器逼近潜在流形的重构误差为:

$$error\left(M\right)=\sum_{j=1}^{M}\sum_{l=1}^{N_j}\left(\boldsymbol{x}_l^j-\boldsymbol{\mu}_j\right)^T\left(\boldsymbol{I}-\boldsymbol{V}_j\boldsymbol{V}_j^T\right)\left(\boldsymbol{x}_l^j-\boldsymbol{\mu}_j\right) \tag{3-23}$$

其中 $\boldsymbol{x}_l^j$，$l=1,\ldots,N_j$ j 是分组到第 j 个局部分析器的个数据点 $N_j\left(\sum_{j=1}^{M}N_j=N\right)$。

3.2.3 SMMC 算法及其计算复杂度分析

当通过 3.2.2 节的方法估计出每个数据点的局部切空间后,即可根据 3.2.1 节介绍的方式计算得到相似性矩阵,随后通过谱方法即可得到聚类结果，SMMC 方法分组混合结构数据的基本过程表 3-1 所示。

表3-1 稀疏子空间聚类算法

算法 3：谱多流形聚类（SMMC）
输入：原始数据集 X ,聚类数 k ,流形维数 d ,局部化模型数 M ,近邻点数 K ,调节参数 o . 算法过程： 1. 利用 MPPCA 训练 M 个 d 维的局部线性模型来近似潜在的流形数据; 2. 根据式（22）确定每个点的局部切空间; 3. 利用式（2）计算两个局部切空间之间的结构相似性;。 4. 利用式（6）计算相似性矩阵 $\boldsymbol{W}\in\Re^{N\times N}$,并计算对角矩阵 $\boldsymbol{D}$,其中 $d_{ii}=\sum_j w_{ij}$; 5.计算广义特征矩阵 $\left(\boldsymbol{D}-\boldsymbol{W}\right)u=\lambda\boldsymbol{D}u$ 最小个特征值对应的特征向量 $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k$; 6.利用 K-means 将 $\boldsymbol{U}=\left\{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k\right\}\in\Re^{N\times k}$ 的行向量分组为 k 个聚类。 输出：原始数据对应的聚类结果。

SMMC 的计算复杂度主要由三部分组成:估计每个数据点的局部切空间、计算相似性矩阵 $\boldsymbol{W}$ 、利用谱方法进行聚类。N 个局部切空间 Θ_i，$i=1,\ldots,N$ 是通过 M 个混合的 MPPCA 的 EM 学习过程得到,其中模型参数通过 K-means 来初始化。这个过程的复杂度为 $O\left(NDM\left(t_1+dt_2\right)\right)$,其中 t_1 和 t_1 分别为 K-means 和 EM 过程收敛所需的迭代步数。在第二部分中,计算任意两个数据点局部切空间之间结构相

似性的复杂度为$O\left(N^2Dd^2\right)$,而搜索每个数据点的个近邻点的复杂度为$O\left(\left(D+K\right)N^2\right)$。第三部分对$W$利用谱方法将数据投影到$k$维嵌入空间并在该空间中利用 K-means 将数据分组为k个聚类,其中广义特征分析的复杂度为$O\left(\left(N+k\right)N^2\right)$而 K-means 在$k$维投影数据上的复杂度为$O\left(Nk^2t_3\right)$ (t_3为该过程中 K-means 收敛所需的迭代步数)。因此,SMMC 方法总的计算复杂度为:

$$O\left(N^3+N^2\left(Dd^2+K+k\right)+N\left(DM\left(t_1+dt_2\right)+k^2t_3\right)\right) \qquad (3\text{-}24)$$

由于K-means和EM过程收敛所需的迭代步数通常较低(少于50),并且$d<D$, $K\ll N$，$k\ll N$，$M\ll N$，因此 SMMC 复杂度主要由数据点数N和数据维数D决定。

3.2.4 参数对算法的影响

SMMC 方法中有三个可调节参数,即局部化模型数M,近邻点数K和调节参数o。下面我们考查这些参数的设置对方法聚类性能的影响,并进而给出参数设置的一些指导建议。

(1)SMMC的性能更多地依赖于局部化模型数M的值局部化模型数越大聚类的性能越好。原因是，随着局部化模型数增加,平均重构误差减小。这意味着对潜在流形局部线性块的近似越来越好,从而对每个数据点局部切空间的估计也越来越可靠,进而使得具有更好的性能,因为其性能依赖于局部切空间的正确估计。

(2)当近邻点数K既不太大也不太小时,SMMC的性能在一个很大的参数选取范围内都是稳健的,这和一些已有的观测事实(例如[2,14])是一致的。其原因在于,当值K太小时会出现很多不连通的子聚类,而当它太大时局部限制会逐渐丧失。

(3)当调节参数o足够大时,SMMC 的性能很好。其原因在于，o越大,来自不同流形的数据之间的可分离性越好,因为对$x<1$而言当o变大时x^o趋向于0。

(4)估计局部切空间的时间和局部化模型数M近似成线性关系,这和理论分析是一致的。然而,SMMC 的总运行时间似乎独立于局部化模型数。这个结果是合理的、可解释的,因为 SMMC 的计算复杂度主要由计算相似性矩阵和执行谱方法进行聚类分析组成,它们是独立于M的。

基于上述观测和分析,我们可以给出参数选取的一些指南。作为一般的推荐,我们建议设置$M=\lceil N/(10d)\rceil$，$K=2\lceil\log(N)\rceil^1$和$o=8$。当所有潜在流形都是线性时,可以采用一个较小的M值，例如$M=3k$。例如。此外,我们推荐在下述参数范围内寻找最优参数: $M\in\left[\lceil N/(10d)\rceil,\lceil N/(2d)\rceil\right]$，$K\in\left[\lceil\log(N)\rceil,3\lceil\log(N)\rceil\right]$，$o\in[4,12]$。然而,需要注意的是,对一般的数据集而言,这些参数的最优选取仍然依赖于数据的分布和噪声水平等等因素。

3.2.5 基于 PID 的参数调节方法

SMMC 作为一种基于流形学习的新型算法，都可以通过合理的调节参数来实现对一些子空间数据集合进行有效的精确分类，但是只有在一定的参数下，才能获得合理正确的分类，为了能够更快更好的获得参数，我们基于自动控制原理中的 PID 调节参数方法，提出了一种类似 PID 参数调节策略。经过我们测试是一种非常高效的针对 SMMC 的参数调节方法。

该方法的调节参数基本思想是，根据推荐参数，将其中两个固定，对其中一个参数根据 3.24 节中其对算法的影响效果进行大小调节，获得合理结果之后，将该参数固定，放开一个参数，如此反复，直到获得合理结果为止。具体算法流程如表 3-2 所示。

表 3-2 基于 PID 的 SMMC 参数调节算法

算法 4 基于 PID 的 SMMC 参数调节方法
输入: 原始数据集 X
算法过程:
1.根据经验规则，选择初始化的参数，一般有 $M \in \left[\left\lceil N/(10d)\right\rceil, \left\lceil N/(2d)\right\rceil\right]$，$K \in \left[\left\lceil \log(N)\right\rceil, 3\left\lceil \log(N)\right\rceil\right]$， $o \in [4,12]$
2: 一般选择 $o=8$,先选定 K，开始对 M 进行调整，每个 M 进行多次蒙特卡洛，具体重复次数根据数据集特点而定，本文给出参考次数为 $N \in [20,50]$.
3: if 在蒙特卡洛次数中出现合理的分类
输出分类参数，和分类的结果
4: else
重新设定 K，进行下一次 M 的选择，直到找到合理的分类结果。输出设定的参数
5: endif
6:while 没有找到合理的分类结果
调节 o，重复上面的过程。
输出: 原始数据的聚类结果。

3.3 问题求解

根据上面几节叙述的规则和一些测验实验，我们对题设问题进行求解。

3.3.1 问题 2(a)

针对问题 2(a)的数据特点，采用我们提出的调节方法进行调节，我们设置 $K=20$，$o=8$ 主要对 M 进行调节，设定合适的 M 进行测试我们选定的测试范围为 $[1\sim 25], M$ 一般不会多大，图 3-2 显示了部分的测试结果图，显然我们发现在这一组参数下的 M =3 时的数据非常的理想，达到了我们分类的目的，而其他的参数条件下，对于这类独立子空间的数据，相对较小的 M 可以达到分类的目的。

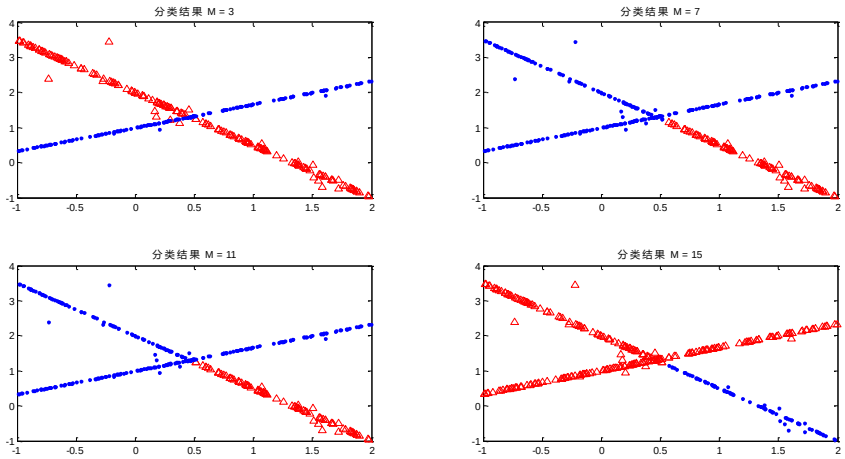


图 3-2 问题 2(a)的 M 值变化部分测试结果

在上述的设定 M 的条件下，我们在考虑 K 对数据的分类效果的影响，我们选定的测试区间为[10~26],部分结果如图 3-3 所示：

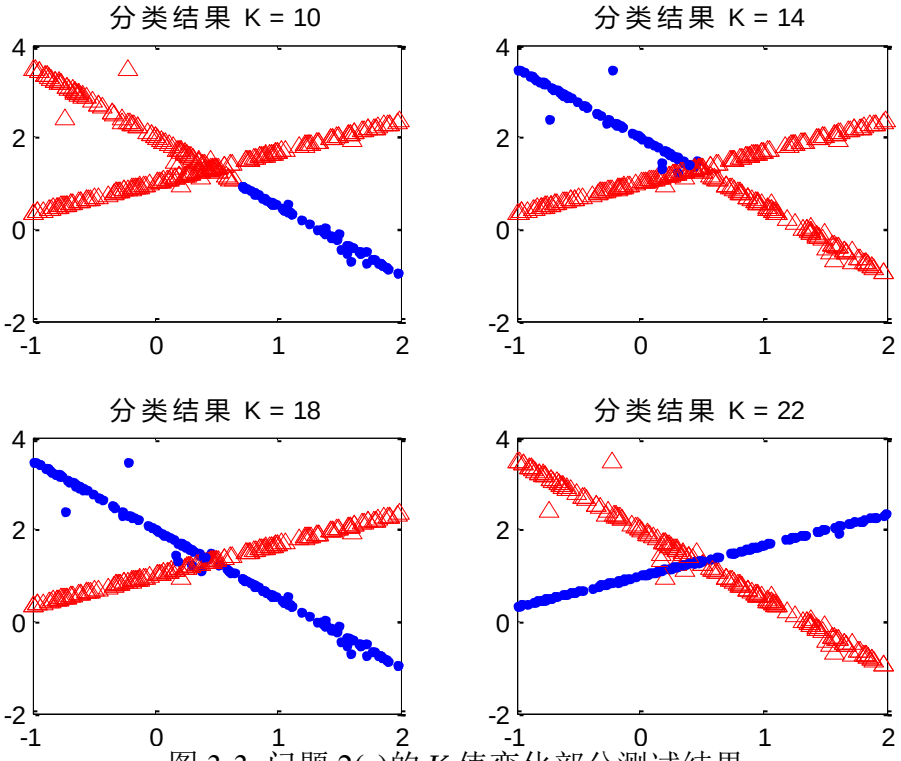


图 3-3 问题 2(a)的 K 值变化部分测试结果

从上面的效果图我们不能发现， K 的取值在 18 和 22 时都可以达到精确分类的效果，经过我们的多次蒙特卡洛实现，我们发现按照 SMMC 算法的相关文献提供的针对此类数据集的参数调节思路，可以得到 K 在一定的区间类可以达到分类的目的，本次得出在[18~30]这个区间内取值的 K 都是可以在其他参数不变化的情况合理的分出数据的。如图 3-4 为最终分类结果。

综合上述我们选择下面参数 $M=3, K=20, o=8$，可以精确分类此类数据集。

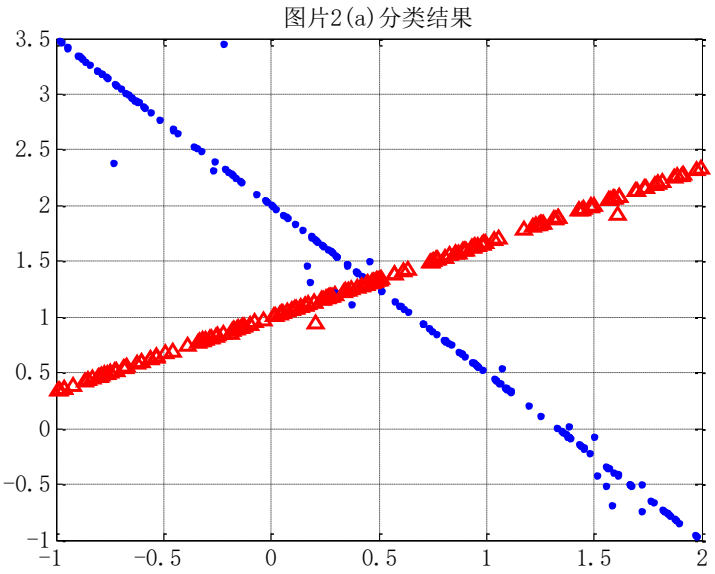


图 3-4 问题 2(a)的分类结果

3.3.2 问题 2(b)

问题 2(b)是一个经典的分类数据集，代表了子空间线性不独立的数据集合，为了设定合理的参数，达到我们分类的目的，我们依然使用 PID 的控制变量的调试方法，正如问题 2(a)一样，我们假定 $M=20, o=8$ 来进行参数 K 的调试，选择的区间[6~10]，从图 3-5 结果可以看出，除了 K =8 时将数据分为两类外，其他参数条件下，显然都不能很好的达到分类效果。

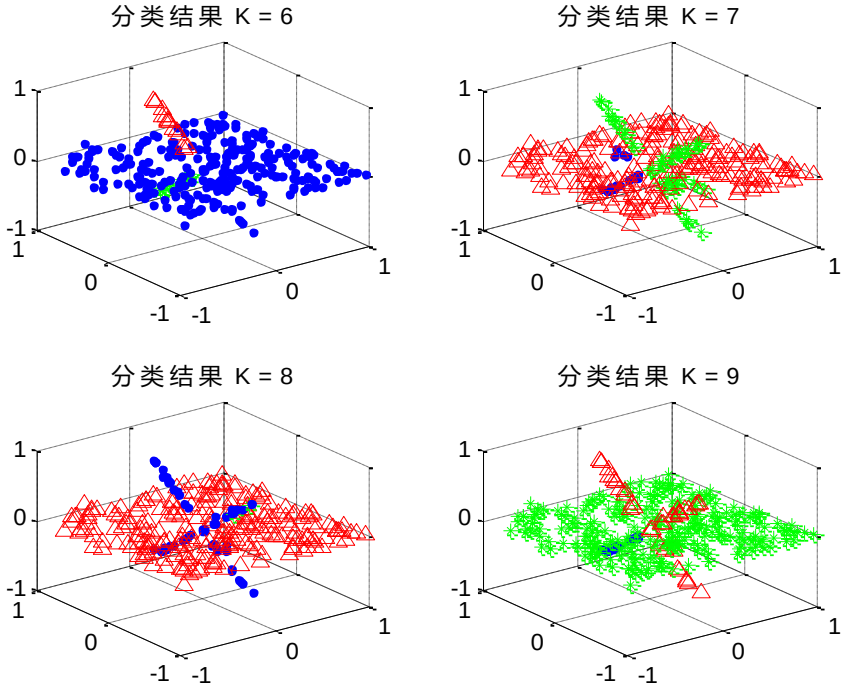


图 3-5 问题 2(b)的 K 值变化部分测试结果

根据 PID 参数调整的思路，以及 SMMC 算法本身参数所代表的特点，我们选择 K =8 不变开始对 M 参数进行调试，以期得到较好的分类效果,选定的测试区

间为[18~22]，从运行结果来看，参数运行至 M =21 时由于算法本身的特点，分类失败，其余分类结果如图 3-6 所示

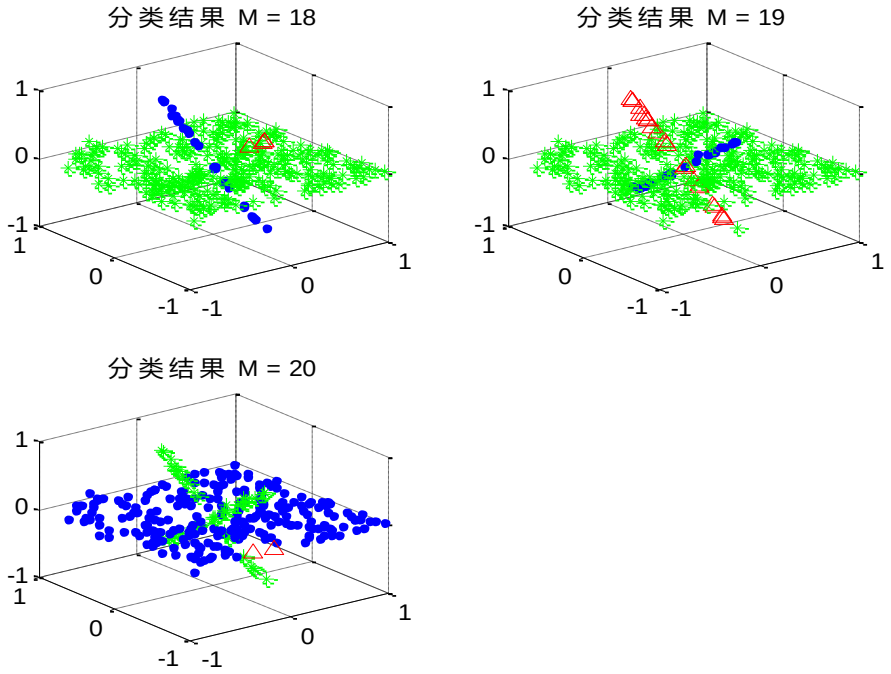


图 3-6 问题 2(b)的 M 值变化部分测试结果

综合上述我们选定这样一组参数 $M=19, K=8, o=8$，进行多次蒙特卡洛试验后，在绝大多数情况下，都很好的分类出了数据，选出一次结果显示如图 3-7：

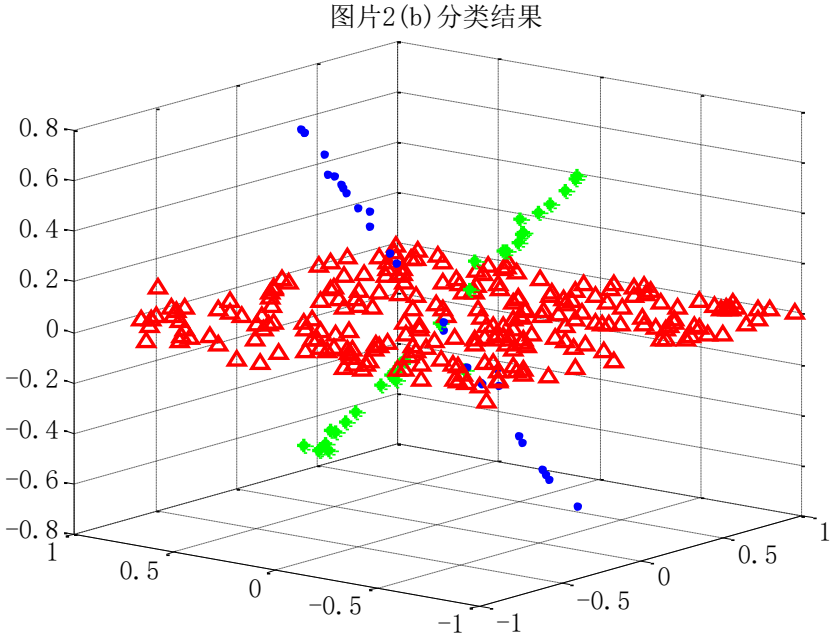


图 3-7 问题 2(b)的分类结果

针对本题中的数据集合，我们发现，SMMC 算法可以很好的实现分类的目的，但是对于此类问题，SMMC 算法的参数鲁棒性显得较差，分类结果的好坏很大程度上取决于我们设定的参数，参数的细微变动都会影响到分类的好坏，同时，

我们也应该看到要分类此种数据集合本身也非常考验算法的分类性能，所以SMMC 分类此种问题不失为一个非常实用的方法。

3.3.3 问题 2(c)

问题 2(c)的数据集是非常经典的独立的多流形聚类数据集，常被用来测试流形学习算法的分类性能，根据其数据集的特点，我们同样选择了 SMMC 算法来进行分类，根据参数调节的规则，我们知道 K 太小时会出现很多不连通的子聚类，而 K 太大的时候会失去局部限制的优势，因此，我们采用上述同样的测试方法测试出了此问题的参数。首先确定 K ,部分结果如图：

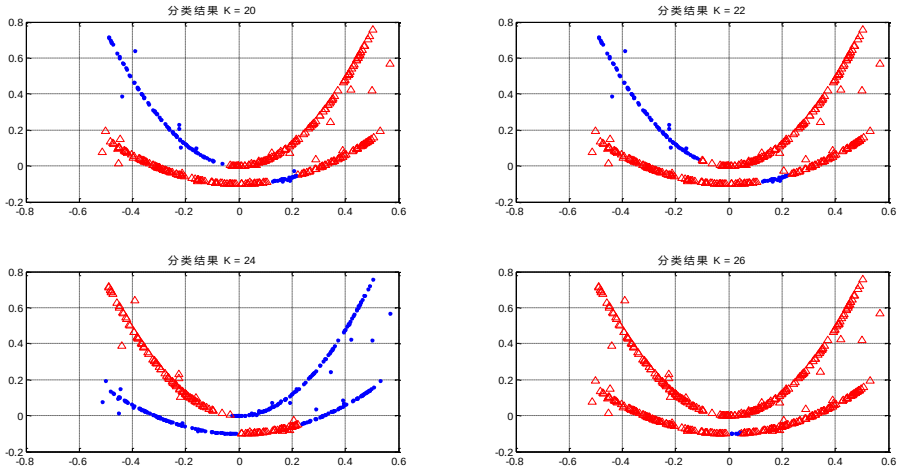


图 3-8 问题 2(c)的 K 值变化部分测试结果

在选定 K 的情况下，进一步测试 M 的取值。经过分析测试后，我们都得到在下面的参数情况下的分类效果可以达到最好，此时选择 $M=30, K=20, o=8$。最优情况下的分类图 3-9 如下

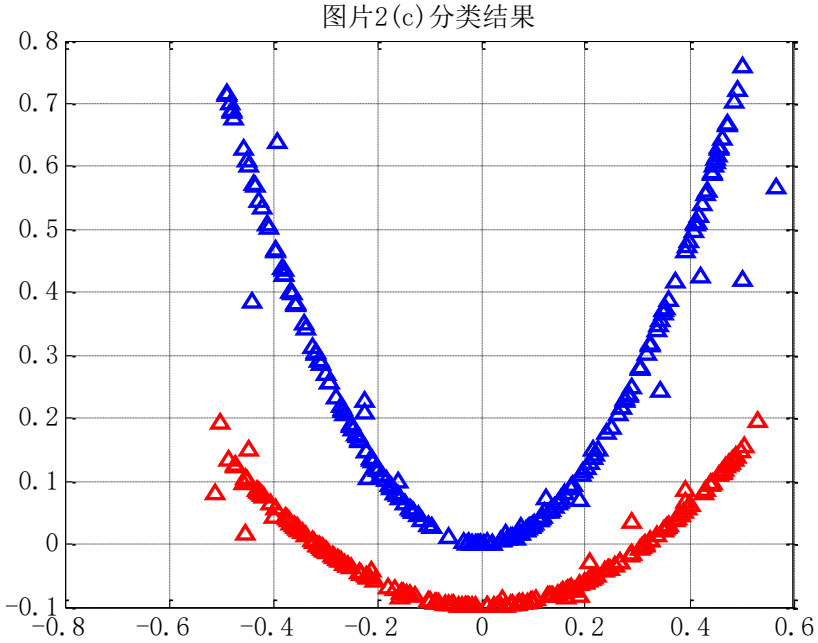


图 3-9 问题 2(c)的分类结果

3.3.4 问题 2(d)

这是一个不独立的多流形聚类的数据集合[18]，为本题中最为复杂的情况，我们同样采用我们提出的方法进行调节 SMMC 参数按照 SMMC 算法的调试依据在下述参数范围内寻找最优参数：$M\in\left[\left\lceil N/(10d)\right\rceil,\left\lceil N/(2d)\right\rceil\right]$，$K\in\left[\left\lceil\log(N)\right\rceil,3\left\lceil\log(N)\right\rceil\right]$，$o\in[4,12]$，一般情况下，我们选择$o=8$，来测试其他的两个参数,首先固定 M =50，测试[18~30]区间内的 K 的取值，其分布结果图如 3-10 所示。

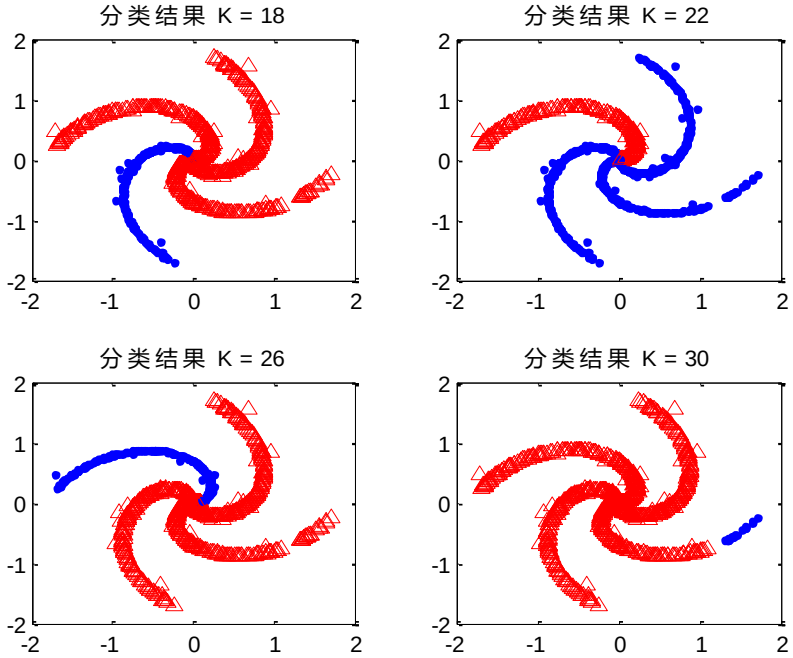


图 3-10 问题 2(d)的 K 值变化部分测试结果

此时我们选择 K =20 和 K =22 来进一步测试选取合适的 M ，测试区间为[30~60],其部分结果如图 3-11 所示

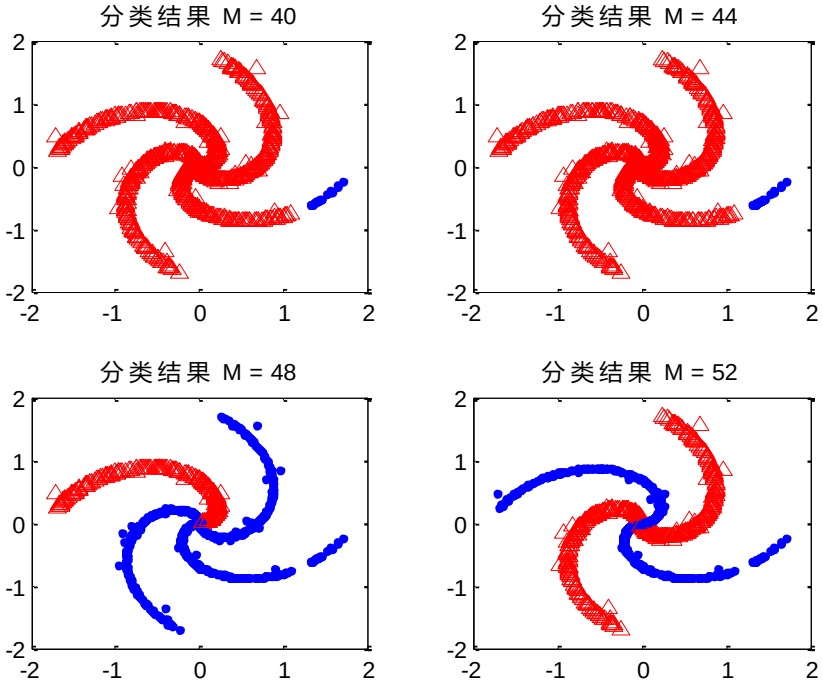


图 3-11 问题 2(d)的 M 值变化部分测试结果

经过多次的测试之后，我们得到了在 $M=52, K=20, o=8$ 时可以有效地将数据进行合理的分类，如图 3-12 所示。

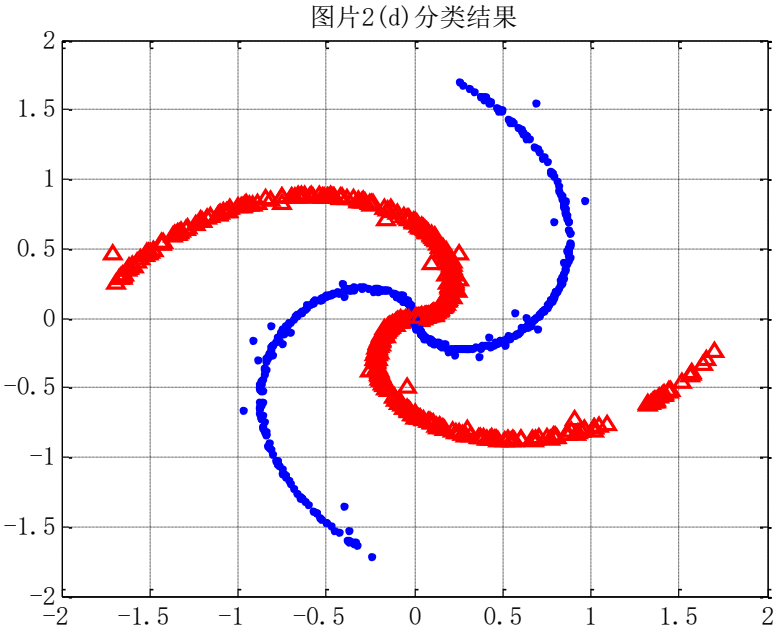


图 3-12 问题 2(d)的分类结果

我们可以看到针对本题中的四个典型问题，SMMC 作为一种基于流形学习的新型算法，都可以通过合理的调节参数来实现对一些子空间数据集合进行有效的精确分类，通过采用提出的参数调节方法，可以很快的获得分类结果，弥补了 SMMC 方法依赖于参数的缺点。

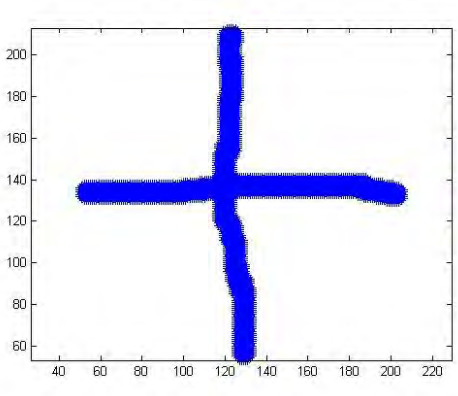
四．问题三的建模与求解

4.1 问题重述

请解决以下三个实际应用中的子空间聚类问题，数据见附件三

(a)受实际条件的制约，在工业测量中往往需要非接触测量的方式，视觉重建是一类重要的非接触测量方法。特征提取是视觉重建的一个关键环节，如图 4-1(a)所示，其中十字便是特征提取环节中处理得到的，十字上的点的位置信息已经提取出来，为了确定十字的中心位置，一个可行的方法是先将十字中的点按照“横”和“竖”分两类。请使用适当的方法将图 4-1(a)中十字上的点分成两类。

(b)运动分割是将视频中有着不同运动的物体分开，是动态场景的理解和重构中是不可缺少的一步。基于特征点轨迹的方法是重要的一类运动分割方法，该方法首先利用标准的追踪方法提取视频中不同运动物体的特征点轨迹，之后把场景中不同运动对应的不同特征点轨迹分割出来。已经有文献指出同一运动的特征点轨迹在同一个线性流形上。图 4-1(b)显示了视频中的一帧，有三个不同运动的特征点轨迹被提取出来保存在了 3b.mat 文件中，请使用适当方法将这些特征点轨迹分成三类。



(a)

(b)

图 4-1 问题四原始图

(c)3c.mat 中的数据为两个人在不同光照下的人脸图像共 20 幅（X 变量的每一列为拉成向量的一幅人脸图像），请将这 20 幅图像分成两类。

问题分析：

作为三个实际问题，我们拟采用三种不同的算法，这样可以更加有针对性更好的解决不同的问题。

(a)为一个十字图形，采用特征提取处理得到，数据为2×2835,为 2835 组二维数据，为低维大数据，要求分为两类，采用 SSC 算法运行时间较长，采用 mum-Cluster 算法，基本思想为直接从整个数据中找出流形交叠的部分并拆开不同的流形结构,从而构造出更忠实于流形结构的近邻图实现混合流形聚类。

(b)作为一个典型的运动分割问题，我们假设三组不同的点来自三个线性流形上，该题目的数据为62×297，其为 297 组 31 帧 2 维坐标数据，我们也可认为其为 297 组 62 维数据，因此可以当成高维数据来进行处理，对于这样一个问题，我们采用一种改进的 SSC 算法，加权 SSC 算法来处理，加权稀疏子空间聚类具有良好的聚类性能，对噪声有更好的鲁棒性，对于自然图像有很好的分割效果。

(c)为两个人在不同光照下的人脸图像，数据为2016×20，即 20 组 2016 维数据，每一列为一张人脸图，对人脸图采用流行学习的方法进行分类，在 PCA 的基础上我们采用二维广义主成分分析，有别于传统的人脸识别算法需要将二维人脸图像矩阵压缩成一维向量, 该方法直接采用二维图像矩阵来构建方差矩阵, 通过在水平和垂直 2 个方向上顺序执行 2 次广义主成分分析（IMPCA）运算[19], 消除了人脸图像行和列的相关性,大大压缩了特征的维数. 选用二维最小近邻分类法进行分类。与主成分分析(PCA)和 IMPCA 相比, 该方法具有更高的识别率和更快的识别速度。

4.2 问题 3(a)

如上节所说，我采用混合流形聚类中的 mum-Cluster 方法来解决源于特征提取的十字数据的聚类问题。

多流形聚类方法(MUlti-Manifold Clustering,简记为 mum-Cluster)[20]。它直接从整个数据中找出流形交叠的部分并拆开不同的流形结构,从而构造出更忠实于流形结构的近邻图来实现混合流形聚类。采用这种方法我们可以很容易获得结果。

4.2.1 mum-Cluster 方法

为了利用谱聚类方法来分组混合结构数据,我们需要构造更“忠实于”流形结

构的无向近邻图,即尽量让来自不同流形聚类的数据不被近邻图连通在一起。由于不“忠实”的无向近邻图通常来自于相互交叠的流形结构。因此,mum-Cluster采用一种“分而治之”(divide and conquer)的思想来处理混合流形聚类问题,它首先找出并处理混合结构的容易部分(easy part),然后再重点解决混合结构的困难部分(hard part)。具体地说,mum-Cluster 首先从整个数据中分离出不同的连通或可分离子集(disjoint subsets),从而将由单一流形构成的纯粹子集(pure subset)和由相交流形构成的交叠子集(intersected subset)区分开来,然后进一步将相交的子集分割为相交区域(intersection area)和非相交区域(non-intersection area),对容易出错的相交部分设法将其拆开为不同的结构并去除不正确的近邻图连接关系得到更“忠实”的无向近邻图,最后用谱聚类方法得到最后的聚类结果。mum-Cluster 方法的基本流程如下图其具体过程如图 4-2.

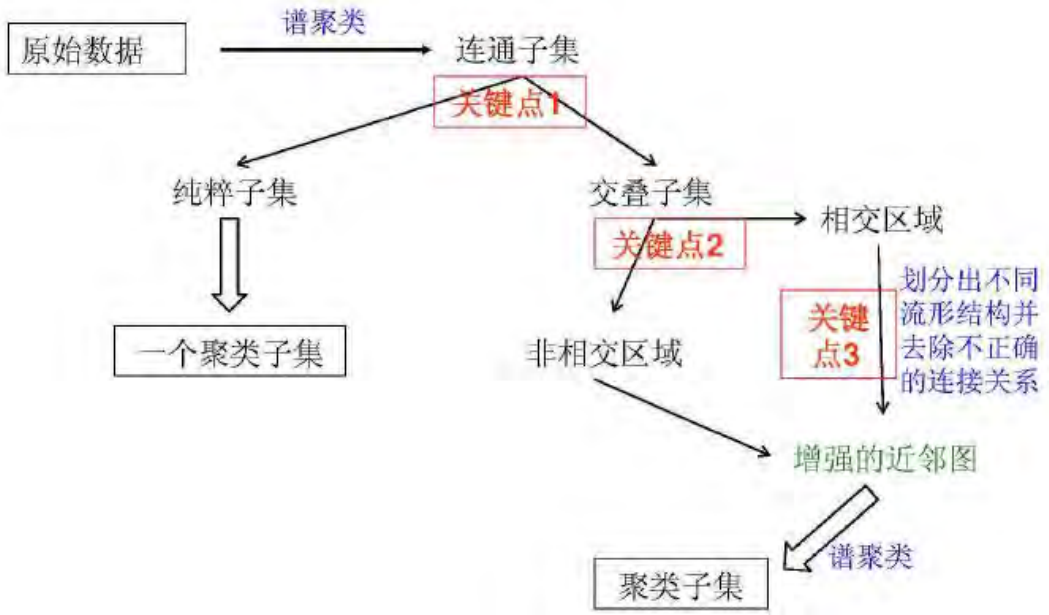


图 4-2 mum-Cluster 方法的基本流程

(1)粗聚类:混合模型通常可以被划分为不同的连通子集,其中一些是由单一流形构成的纯粹子集(pure subset),另一些是由相交流形构成的交叠子集(intersected subset)。为分别处理这两类不同结构的子集,我们采用谱聚类将整个数据集粗略地划分为不同的连通子集(称为粗聚类,coarse clusters)。注意,本节中的谱聚类采用基于 K-近邻图和简单核的非对称型规范化谱聚类。需要注意的是,UNSC-KS 中需要提供低维嵌入空间的维数或连通子集的个数 r。

(2)确定连通子集的结构:在得到了不同的连通子集 $\Theta_c, c=1,\ldots r$ 后,一个关键性的问题是如何确定它们的结构,即该聚类子集是纯粹的还是交叠的。我们可以利用数据点的本征维数来解决这个问题,它基于这样一个观测:如果聚类子集中的数据点来自一个单一的流形,它们的本征维数应该相同,否则本征维数不同。

(3)确定交叠子集的相交区域和非相交区域:如果一个连通子集是纯粹的,那么我们实际上已经得到了一个流形聚类。然而,更困难的问题在于交叠子集,我们需要进一步处理它以得到该子集中的不同流形聚类。这里的另一个关键点是如何找出数据的相交区域和非相交区域。通常,相交区域的数据点由于有其它流形结构上的数据点存在,其估计的维数会大于流形的真实维数。因此,我们将具有最高估计维数 $d_{\max}$ 的数据点作为相交区域中的点。在实践中,由于相交处结构的复杂性,我

们将具有最高估计维数的ε–近邻点也作为相交区域的点,即: X 是相交区域中的点, 如果 $\| X - X^{ip} \|^2 < \varepsilon$，其中 X^{ip} 是任一个具有 $d_{\max}$ 维数的数据点。最后,每个交叠子集被划分为相交区域和非相交区域。

(4)相交 / 非相交区域聚类:相交 / 非相交区域中的数据点可能是由很多更小的子集组成的,即数据被划分成了很多不同的相交聚类(intersection clusters)和非相交聚类(non-intersection clusters),同样需要找出这些不同的子聚类并进行分别处理。由于这些子聚类通常是良分离的,因此我们同样利用 UNSC-KS 对它们进行分组。

(5)细聚类:相交子聚类意味着其中的数据点是由不同流形上的数据点交叠而成的,因此一个关键点就是如何把它们区分开来。尽管整个流形数据是非线性的,然而每个相交子聚类只是一个局部区域,因此它可以看作是由非线性流形的线性部分交叠而成的。另一方面,我们已经看到线性流形聚类方法能很好地拆开交叠流形的不同部分。因此,我们可以利用 K-flats 方法来将每个相交子聚类中的不同流形结构区分开来(找到的不同结构称为细聚类,fine clusters)。

(6)最终聚类:由于传统谱聚类基于欧氏距离来构造近邻图,每个交叠子集中所构造出的近邻图因此将不同的流形连接在了一起。因此根据上一小节的分析,为利用谱聚类将不同的流形分割出来,需要将这些不正确的连接关系去除,同时保持同一流形内部的连接关系。由于不正确的连接关系主要来自于流形相交区域的不同细聚类,因此我们将这些细聚类之间的连接关系去除同时将每个细聚类自身的点都连接起来以保持流形结构,最后我们就得到了一个增强的更“忠实”于流形结构的近邻图。随后可以利用谱聚类进一步来得到最后的聚类子集,称为 final clusters。

整个多流形聚类(MUlti-Manifold Cluster,简记为 mum-Cluster)方法的基本步骤如表 4-1 所示。

表 4-1mum-Cluster 算法

算法 5 mum-Cluster
输入: 原始数据集 X ,近邻点数 K ,确定相交区域的阈值参数 ε ,最大误差阈值 $\zeta_{\max}$
算法过程:
1: 在原始数据上构造近邻图和相似性矩阵,并利用谱聚类和 eigengap 策略来分组连通子集;
2: 对每个连通子集,计算其中数据点的本征维数;
3: **if** 该连通子集中数据点的本征维数都相同
将该连通子集作为一个流形聚类输出;
4: **else**
通过 4.2.1 节中过程 (3)-(6) 构造新的近邻图并利用谱聚类来分组出不同的流形聚类;
5: **endif**
输出: 原始数据的聚类结果。

4.2.2 算法复杂度分析

mum-Cluster 方法的计算复杂度主要由三部分组成:本征维数估计、连通成分确定和细聚类发现。 N 个 D 维数据点的本征维数通过在每个点的 K 个近邻点上执行局部 PCA 来估计,其复杂度为 $N \times O(KD\min(K, D))$。谱聚类被用来确定 r 个

连通成分,该过程总的复杂度为$O((D+K+r)N^2+Nr^2t)$,其中$O((D+K)N^2)$是构造相似性图的复杂度,而$O(rN^2)$和$O(Nr^2t)$分别是计算前r个广义特征向量和在r 维空间迭代t次 K-means 得到聚类结果的复杂度。由于$r \ll N$,$L \ll N$ 并且K-means 能快速地收敛,因此连通成分确定的复杂度主要为 $O(N^2 \max(D,N))$。利用 K-flats 来分组细聚类的计算复杂度不易直接估计,因为我们并不知道待分组数据的个数,同时需要一个自下而上的估计策略来确定其最优的细聚类数及其维数。然而,类似 [89] 中的分析,该过程的最坏复杂度为$O(m^2)\cdot O(DN\min(D,N))$,其中$m$为非相交聚类的个数。总之,mum-Cluster 的计算复杂度被限制在$O(N^2\max(D,N))$的量级上,即主要由原始样本点数N和数据维数D确定。

4.2.3 参数影响

mum-Cluster 方法中有三个参数,即K, ε和$\zeta_{\max}$。,我们通过固定其它参数而变化所关心的参数来考察不同参数对 mum-Cluster 方法聚类性能的影响。结果发现 mum-Cluster 在很大的参数选取范围内，都能得到很好的聚类结果。具体地说,只要参数K选取的既不太大也不太小, mum-Cluster 的性能对它的设置是不敏感的。其原因在于K代表近邻数据点的个数,当它太小时会不足以提供足够的数据结构信息同时会导致很多不连通的子聚类;而当它太大时,局部特性会随之丧失。mum-Cluster 的性能会随着ε的变大而降低,其原因在于ε控制相交区域点的扩展区域,当它太大时不能保证该区域是近似局部线性的。mum-Cluster 对$\zeta_{\max}$的设置不是很敏感。

总的来说,从图中可以看出,参数K的选取在三个参数中对 mum-Cluster 方法的聚类性能影响最大,这表明局部本征维数的正确估计是 mum-Cluster 的一个关键过程。

值得一提的是，在问题二求解过程中采用的基于 PID 的 SMMC 参数调节算法同样可以应用到 mum-Cluster 参数调节上，实验证明也具有很好的效果。

4.2.4 问题 3(a)求解

本题中数据样本分布特点和问题 2(a)非常的类似，属于线性独立的子空间数据集，但是其不同点在于本例中数据的样本量很大，所以我们选择了一种基于流形学习的 mum 聚类方法，我们在进行图像的处理后，通过上节提出参数调节的方法，对 mum-Cluster 进行参数调节，获得结果。我们需要的调节的参数为K, $\zeta_{\max}$,ε。三个参数来进行测试，首先我们先固定$\zeta_{\max}$=9，ε=1。然后开始搜索寻找来进行寻找合适的最优的值。

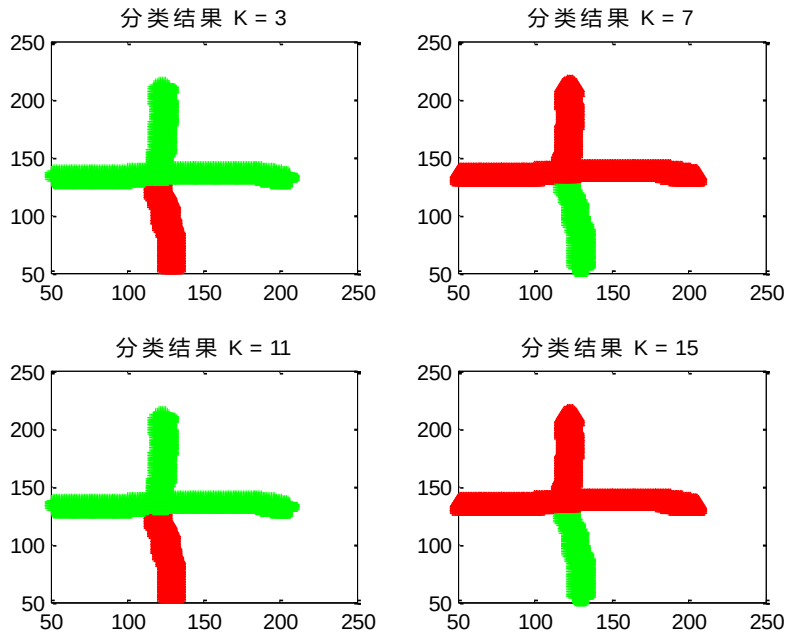


图 4-3 问题 3(a) K 变化时部分分类结果

图 4-3 显示了部分的仿真实验结果，通过我们 PID 参数调试方法，我们最终得出了合适的参数 $K=15,\zeta_{\max}=10,\varepsilon=1$

采用上述参数，获得聚类结果如图 4-4，可以看出采用 mum-Cluster 可以很容易将其分为两类，分类效果极佳。

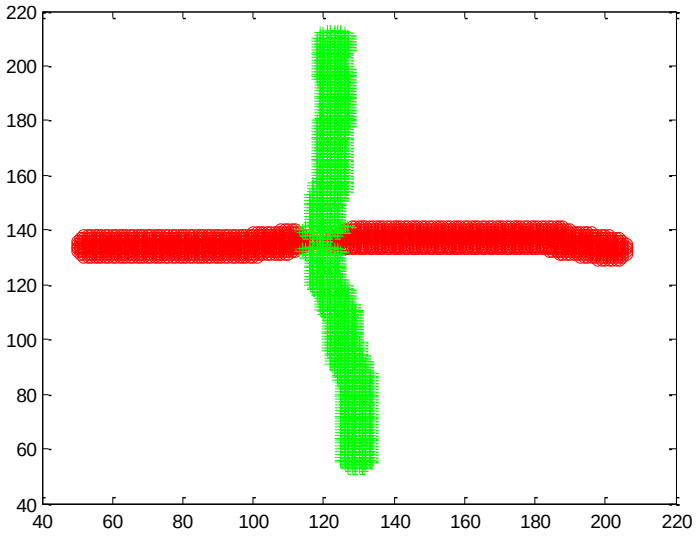


图 4-4 问题 3(a)分类结果

从上面的分类图像我们可以看出，采用了 mum-Cluster 聚类方法可以合理的分类出最后的结果，同样本算法也是一个对参数很敏感的算法，对于这样的大数据量的数据，我们采取图像处理的方法进行抽样后进行小块分类，在进行 mum-Cluster 分类算法进行分类就解决了样本特征不够，无法精确分类的缺陷，而处理后就可以使用基于流形学习的 mum-Cluster 算法进行聚类。

4.3 问题 3(b)

作为一个运动分割问题，其每一组点我们都认为来自于同一个线性流行，这样采用基本的 SSC 算法就可以解决这个问题，但 SSC 算法存在一定缺陷，为了能够获得更好更快的结果，我们采用了加权稀疏子空间聚类方法，加权 SSC 具有良好的聚类性能，对噪声有更好的鲁棒性，对于自然图像有很好的分割效果。并与 SSC 算法结果进行了对比。

4.3.1 加权稀疏子空间聚类

在 2.2.1 节 SCC 算法基础上，我们简要介绍加权稀疏子空间聚类算法[21]。

子空间聚类方法的基本假设是所有数据来自若干线性或仿射子空间的并。对图像的特征数据来讲，这一假设不一定准确成立，因此，在 2.2.1SCC 算法基础上，采用一个加权稀疏子空间表示模型，优点有两方面：一方面，本文采用高斯相似度作为权重，这是一种全局相似度，从而加权稀疏是对数据的一种全局约束，弥补了稀疏是一种局部约束的不足；另一方面，权重的引入有利于使稀疏子空间聚类和图像分割结果对前述假设更加鲁棒。加权稀疏子空间聚类具有良好的聚类性能，对噪声有更好的鲁棒性，对于彩色自然图像有很好的分割效果。因此对于 3(b)问题来源于实际车辆的运动会有更好的效果。

加权稀疏子空间表示模型中的权重有利于使得数据尽可能被最相似的数据线性表示。利用高斯相似度函数

$$\boldsymbol{w}_{ij}=\exp\left(-\frac{\left\|\boldsymbol{d}_i-\boldsymbol{d}_j\right\|_2^2}{\sigma^2}\right) \tag{4-1}$$

式中，$\boldsymbol{d}_i$和$\boldsymbol{d}_j$是 2 个数据样本，对l_1范数加权，建立如下加权稀疏表示模型

$$\min_{\boldsymbol{u}_i}\sum_{j\neq i}\frac{1}{\boldsymbol{w}_{ji}}\left|\boldsymbol{u}_{ji}\right|+\lambda\left\|\boldsymbol{D}\boldsymbol{u}_i-\boldsymbol{d}_i\right\|_2,\ \boldsymbol{u}_{ii}=0 \tag{4-2}$$
$$i=1,\ldots,N$$

模型中相似度越大则权重越小，相似度越小则权重越大，因此加权稀疏约束有利于使得数据尽可能被最相似的数据线性表示，与不相似的数据无关。

模型求解时常转换为如下模型

$$\min_{\boldsymbol{u}_i}\sum_{j\neq i}\frac{1}{\boldsymbol{w}_{ji}}\left|\boldsymbol{u}_{ji}\right|+\lambda\left\|\boldsymbol{D}\boldsymbol{u}_i-\boldsymbol{d}_i\right\|_2,\ s.t.\ \boldsymbol{u}_{ii}=0 \tag{4-3}$$

式中，$i=1,\ldots,N$，$\lambda>0$为参数。再利用 CVX—MAT—LAB 工具包逐步迭代求出u_i的最优近似解。利用$W=\dfrac{\left|\boldsymbol{U}+\boldsymbol{U}^T\right|}{2}$构造相似度矩阵，利用成熟的谱聚类算法，如规范化割(Ncut)就可以得到最终聚类结果。

基于上述分析，本文给出基于加权稀疏子空间聚类的数据聚类算法表 4-2 所示。

表 4-2 加权稀疏子空间聚类的数据聚类算法

算法 6 加权稀疏子空间聚类的数据聚类算法

输入：数据集 X ，分类数目 n ,超像素个数 N
算法过程：
1 利用边界概率和 Ncut 算法得到 N 个超像素；
2 对于每个超像素提取直方图，得到特征矩阵 $\boldsymbol{D}=\left[\boldsymbol{d}_1,\boldsymbol{d}_2,\ldots,\boldsymbol{d}_N\right]$；
3 利用加权稀疏子空间聚类模型式（4-3）得到表示系数 $\boldsymbol{U}$ ；
4 由 $\boldsymbol{U}$ 构造图的相似度矩阵 $W=\dfrac{\left|\boldsymbol{U}+\boldsymbol{U}^T\right|}{2}$ ；
5 利用 Ncut 算法得到超像素的聚类结果，即图像分割结果。
输出: 原始数据的聚类结果。

4.3.2 规范划割 Ncut 算法

基于图论的图像分割是自底向上的分割方法之一，Minimum Cut、Normalized Cut、Min-Max Cut 等都属于这种类型。其中，Normalized Cut(简称 Ncut)[22]较其它方法更有优势，因为它提供了一种规范化的分割准则，不会产生分割时偏向小区域的情况。如今，Normalized Cut 的图像分割方法已经在计算机视觉的各个领域被广泛使用。

在基于图论的图像分割方法中，一幅图像被视作带权的无向图 $G=\{V,E,W\}$，其中 y 代表节点的集合，在图像中则表示为像素集，E 代表了连接两两节点的边集，而 $W(i,j)$ 代表了两个节点之间的权重值，在图像中权重可根据像素之间颜色、亮度等信息的距离来计算。将图像分割为两部分 $A,B: A\cup B=V, A\cap B=\varnothing$。而两个子集的相似程度用 cut 表示：

$$cut(A,B)=\sum_{i\in A,\, j\in B} W(i,j) \tag{4-4}$$

Shi 和 Malik 提出了一种规范化的准则来衡量一个分割，称为 Normalized Cut，简称为 N-Cut，并可以求解出一个 Ncut 值作为衡量标准：

$$Ncut(A,B)=\frac{cut(A,B)}{assoc(A,V)}+\frac{cut(B,A)}{assoc(B,V)} \tag{4-5}$$

其中 $assoc(A,V)=\sum_{i\in A,k\in V} W_{ik}$ 代表 A 集的节点集与整个节点集之间的权重和。而最优的分割方案就是使上式的值达到最小。

之后，Stella Yu 和 Shi 利用相同的思想提出了一种多分割的 K-way N-Cut 方法，根据给出的 K 值将图像分割成最优的 K 个区域。在 K-way N-Cut 方法中，像素集合 y 需要被分割成 K 个不相交的子集：$\{V_1,V_2,\cdots,V_K\}$。用 $\boldsymbol{W}$ 表示图像集合 V 的权重矩阵，$\boldsymbol{D}$ 表示一个对角矩阵，满足 $\boldsymbol{D}_{ii}=\sum_j \boldsymbol{W}_{ij}$。这个 K 分割的结果可以用 $N\times K$ 的矩阵 $\boldsymbol{X}$ 来表示，而且 $\boldsymbol{X}=\left[X_1,\cdots,X_K\right]$，其中 N 是待分割图像的像素个数，X_l 表示了一个由 0，1 组成的 $N\times 1$ 的向量，而且每一个像素点只能属于唯一的一个子集。所以满足：

$$X(i,l)=\langle i\in V_l\rangle, \forall i\in V, l\in\{1,\cdots,K\} \tag{4-6}$$

上式中$\langle\bullet\rangle$中的值若为真，则返回 1；若为假，则返回 0。

现将V 分割成A,B两部分， $A,B\subset V$ ，则定义连接A,B集合的权重和为：

$$links(A,B)=\sum_{l\in A,k\in B}\boldsymbol{W}(i,j) \tag{4-7}$$

及其规范化后的结果：

$$linkratio(A,B)=\frac{links(A,B)}{links(A,V)} \tag{4-8}$$

为了达到最佳分割所要求的子集内部的紧密性和子集之间的松散性，分别定义了*knassoc* 和*kncut* 这两个衡量标准：

$$knassoc=\frac{1}{K}\sum_{l=1}^{K}linkratio(V_l,V_l) \tag{4-9}$$

$$kncut=\frac{1}{K}\sum_{l=1}^{K}linkratio(V_l,V/V_l) \tag{4-10}$$

由于$knassoc+kncut=1$，最小化*knassoc* 和最大化*kncut* 完全等价，所以选择了最大化*knassoc* 来达到最优分割。

最终，分割的最优解可以根据求解以下公式得到：

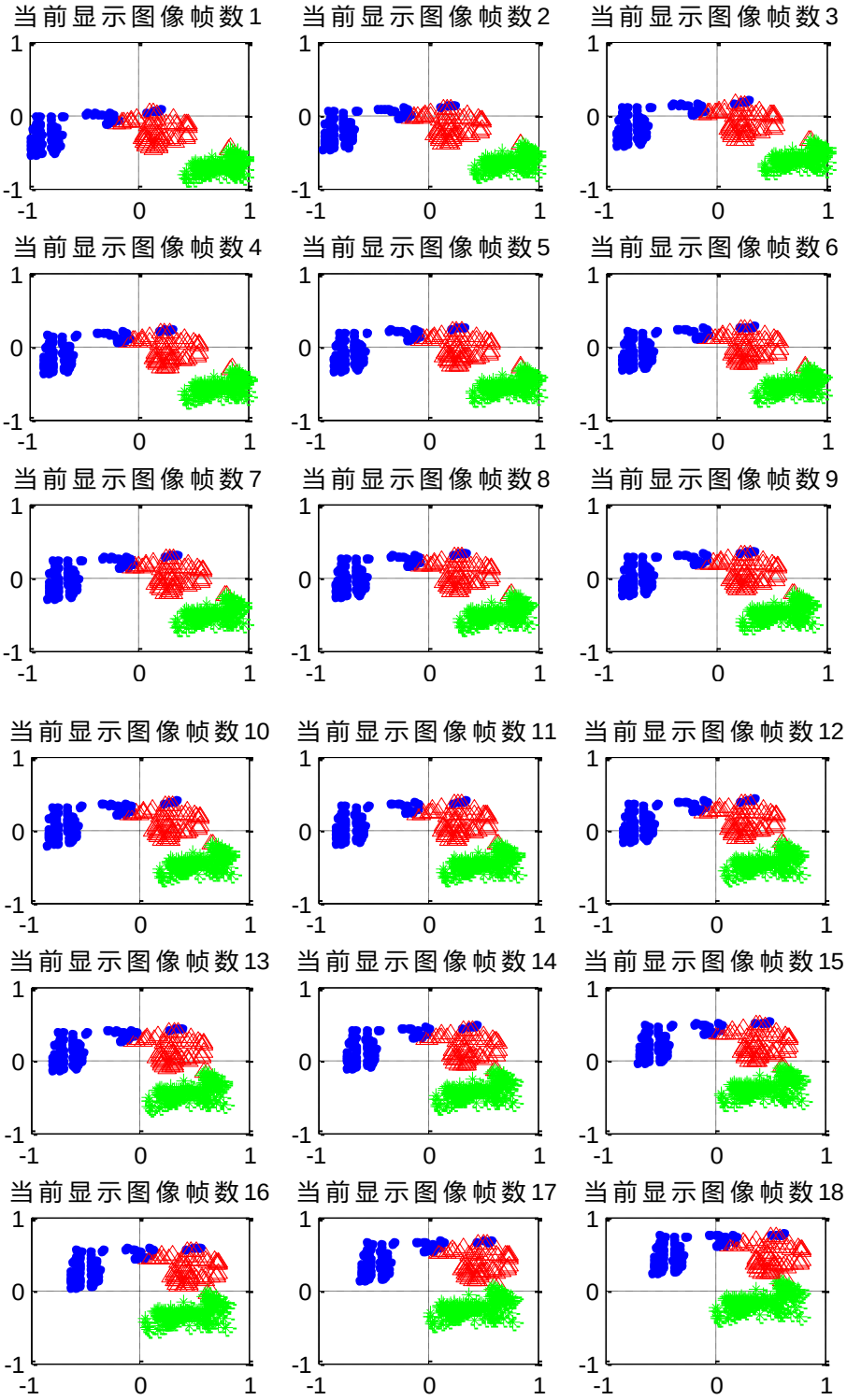
$$\boldsymbol{X}=\arg_X\min\frac{1}{K}\sum_{l=1}^{K}\frac{\boldsymbol{X}_l^T\boldsymbol{W}\boldsymbol{X}_l}{\boldsymbol{X}_l^T\boldsymbol{D}\boldsymbol{X}_l} \tag{4-11}$$

在应用 N-Cut 准则进行图像分割时，分割结果与像素之间的权重值W_{ij}的定义方法密切相关。

4.3.3 问题 3(b)求解：

运动分割是指在一个场景下根据不同刚性物体的运动, 将视频序列分成多个时空区域,即把视频里做刚性运动的物体上的特征点进行聚类, 使得每一类对应于一个独立运动的物体, 从而得到物体运动的轨迹.显然我们可以看出 3(b)题目是一个运动分割的类型数据，我们需要一个算法对其进行运动分割，并进行运动识别，针对这样的数据，稀疏子空间聚类算法 SSC 来处理解决这类问题。

首先我们采用常规的 SSC 算法来处理这个问题，得到分类的问题如下其分类的结果如图 4-5 所示：


当前显示图像帧数1
当前显示图像帧数2
当前显示图像帧数3
当前显示图像帧数4
当前显示图像帧数5
当前显示图像帧数6
当前显示图像帧数7
当前显示图像帧数8
当前显示图像帧数9
当前显示图像帧数10
当前显示图像帧数11
当前显示图像帧数12
当前显示图像帧数13
当前显示图像帧数14
当前显示图像帧数15
当前显示图像帧数16
当前显示图像帧数17
当前显示图像帧数18
1
0
-1
-1
0
1

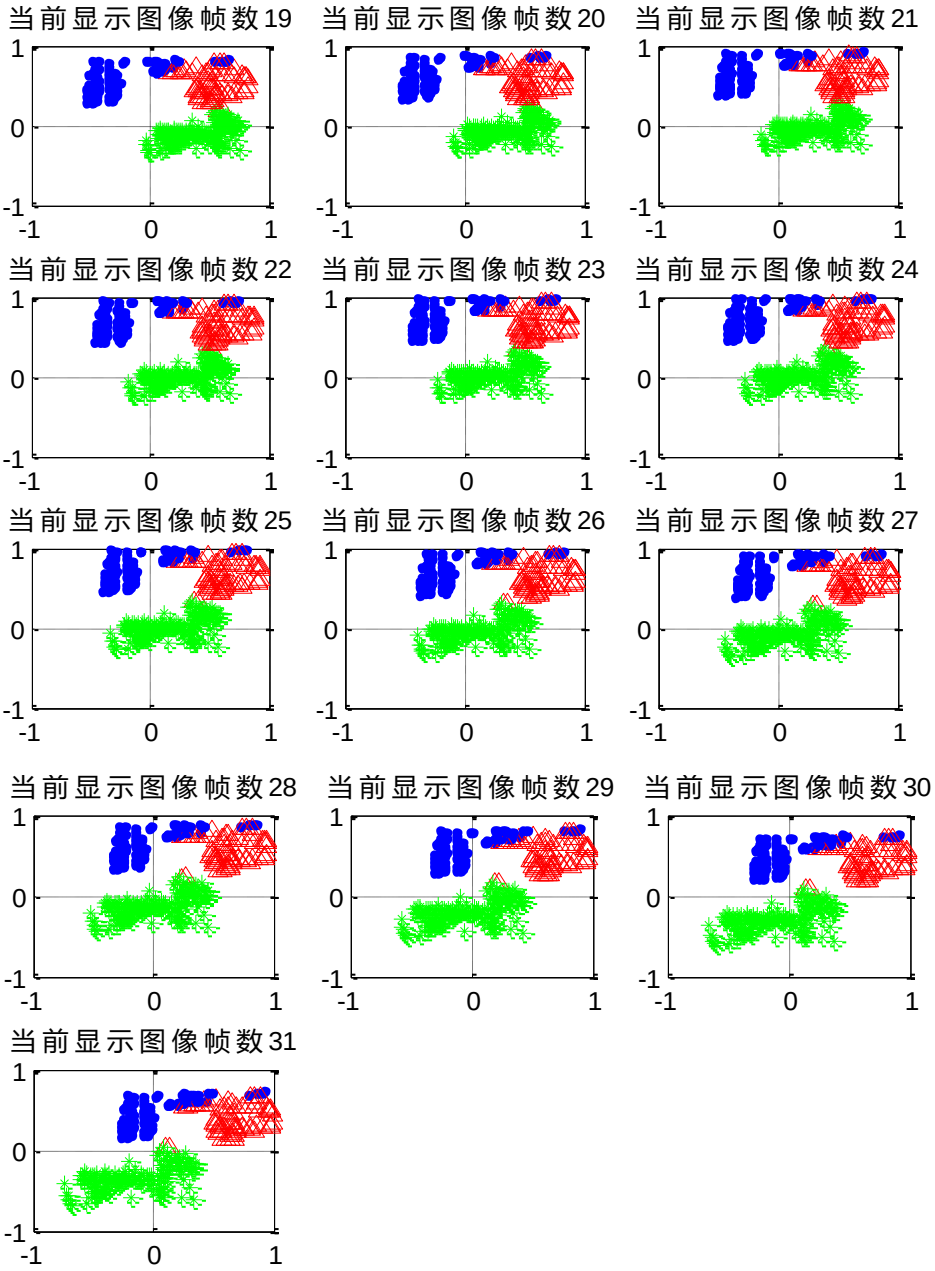


图 4-5 采用 SSC 算法的跟踪分类结果

从上面的结果可以看出，采用 SSC 算法只能进行粗略的跟踪，分类情况并不准确。从中选取其中一帧的图片其分类结果如图 4-6 所示。

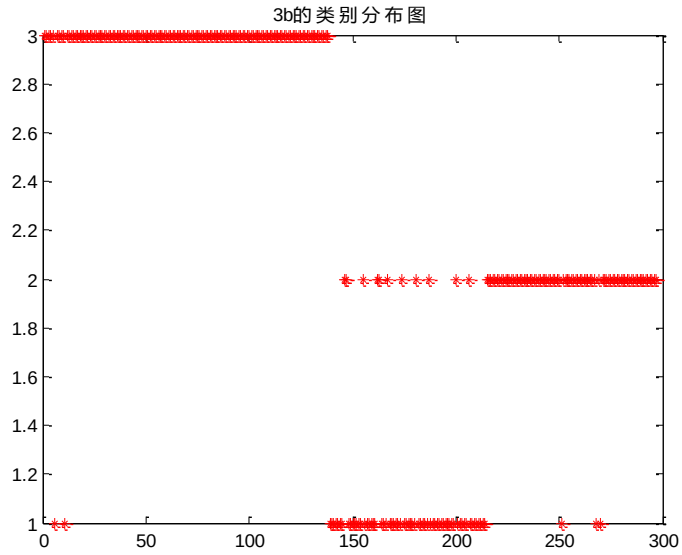
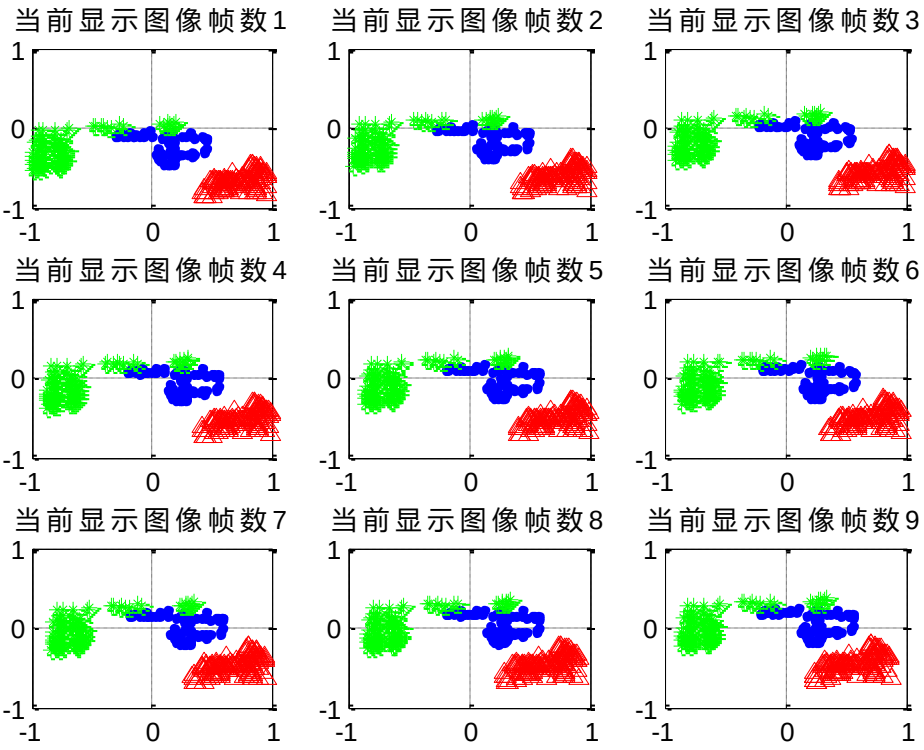
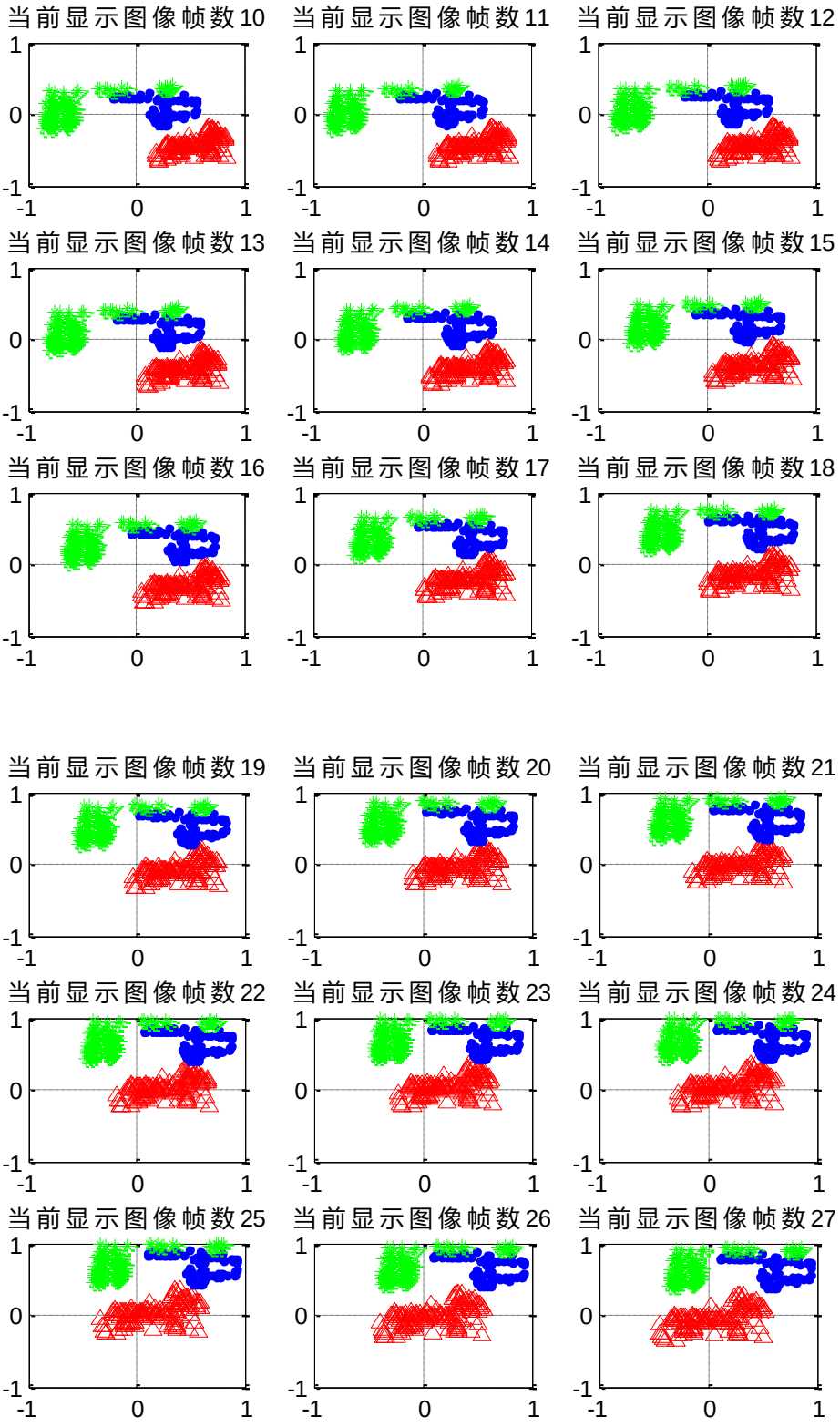


图 4-6 采用 SSC 算法的第 12 帧类别分布

从类别的结果来看，我们看出基本的 SSC 算法针对这一类数据只能进行粗略的跟踪，根据这样的分类情况，我们引入了加权 SSC 算法[23]来进行运动分割，加权 SSC 采用高斯相似度作为权重，这是一种全局相似度，从而加权稀疏是对数据的一种全局约束，弥补了稀疏是一种局部约束的不足；另一方面，权重的引入有利于使稀疏子空间聚类和图像分割结果对前述假设更加鲁棒。采用加权 SSC 算法获得 31 帧图像如图 4-7 所示。




当前显示图像帧数10
当前显示图像帧数11
当前显示图像帧数12
当前显示图像帧数13
当前显示图像帧数14
当前显示图像帧数15
当前显示图像帧数16
当前显示图像帧数17
当前显示图像帧数18
当前显示图像帧数19
当前显示图像帧数20
当前显示图像帧数21
当前显示图像帧数22
当前显示图像帧数23
当前显示图像帧数24
当前显示图像帧数25
当前显示图像帧数26
当前显示图像帧数27

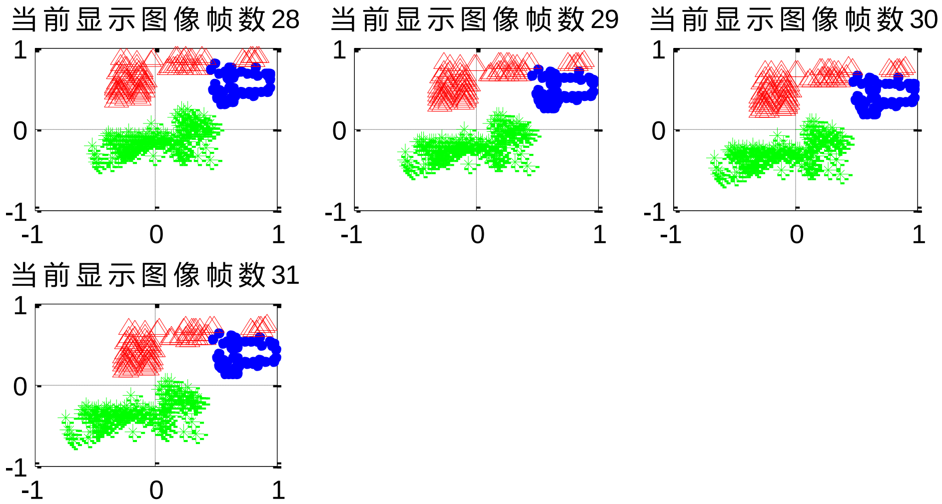


图 4-7 加权 SSC 的跟踪分类结果

从跟踪的结果来看，加权 SSC 算法很好的达到了运动分割的特点，每一帧都很好的跟踪上了目标，特别是有一个独立的小块，区分这样的运动部分是非常有难度的，但是基于稀疏表示之后，这样的一种谱聚类算法实现了运动分割。同样，选取第 12 图像做出类别分布图，如图 4-8 所示。

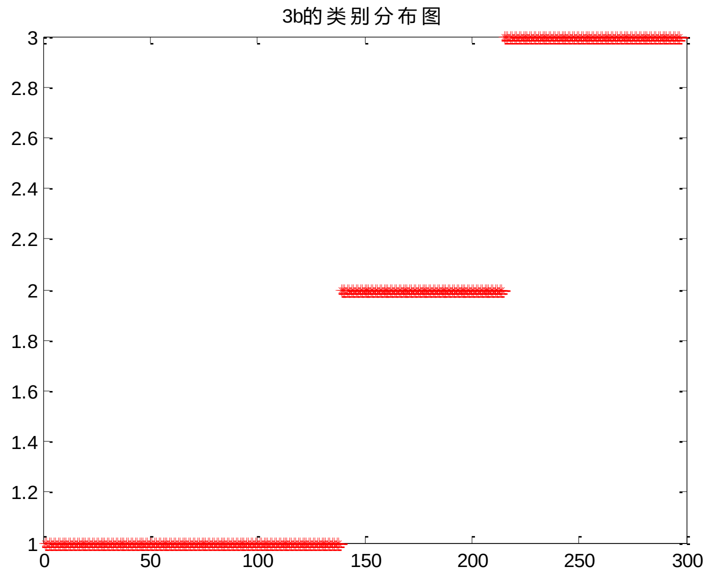


图 4-8 采用加权 SSC 算法的第 12 帧类别分布。

从图 4-8 对比图 4-6，我们可以发现采用加权 SSC 算法的结果更加准确，效率更高。根据题目要求给出问题 3(b)的标签表，如表 4-3 所示。

表 4-3 问题 3(b)分类标签

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2

2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3			

结合表中分类标签，和题目中的原始图像信息，可以得知
1—138 属于同一流形，属于一个类，为背景信息；
139—214 属于同一流形，属于一个类，为公共汽车；
215—297 属于同一流形，属于一个类，为小汽车。

综上，采用加权 SSC 算法又好又快的解决了 3(b)的运动分割问题，同样该算法具有一定普适性，可以用来解决其他运动分割的问题。

4.4 问题 3(c)

问题 3(c)为一个典型的不同光照下人脸识别的问题，对于这种问题有很多算法可以很好的处理，但有些算法的速度较差，作为 2 维灰度图像我们采用 2 维广义主成分分析来解决这个问题。同时与其他算法进行了对比。

4.4.1 广义主成分分析

设 $\boldsymbol{U}$ 为 n 维列向量，$\boldsymbol{A}$ 为 $m\times n$ 维人脸图像，将 $\boldsymbol{A}$ 通过线性变换投影到 $\boldsymbol{U}$ 上：

$$\boldsymbol{B}=\boldsymbol{AU} \tag{4-12}$$

式中：m 维向量 $\boldsymbol{B}$ 为图像 $\boldsymbol{A}$ 的投影特征向量。可以通过投影特征向量的散布情况来决定最佳的投影轴 $\boldsymbol{U}$，采用以下准则：

$$J(\boldsymbol{U})=tr\boldsymbol{S}_{\boldsymbol{u}} \tag{4-13}$$

式中：$\boldsymbol{S}_{\boldsymbol{u}}$ 为投影特征向量 $\boldsymbol{B}$ 的类间散布矩阵，tr 表示取类间散布矩阵的迹。最大化准则(2)就是寻找一个投影轴 $\boldsymbol{U}$，使投影后特征向量的类间散布量最大。

$$\begin{aligned}\boldsymbol{S}_{\boldsymbol{u}}&=E\left\lfloor\left(\boldsymbol{B}-E(\boldsymbol{B})\right)\left(\boldsymbol{B}-E(\boldsymbol{B})\right)^T\right\rfloor=\\&E\left\lfloor\left(\boldsymbol{AU}-E(\boldsymbol{AU})\right)\left(\boldsymbol{AU}-E(\boldsymbol{AU})\right)^T\right\rfloor=\\&E\left\{\left[\left(\boldsymbol{A}-E(\boldsymbol{A})\right)\boldsymbol{U}\right]\left[\left(\boldsymbol{A}-E(\boldsymbol{A})\right)\boldsymbol{U}\right]^T\right\}\end{aligned} \tag{4-14}$$

$$tr\boldsymbol{S}_{\boldsymbol{u}}=\boldsymbol{U}^TE\left\lfloor\left(\boldsymbol{A}-E(\boldsymbol{A})\right)^T\left(\boldsymbol{A}-E(\boldsymbol{A})\right)\right\rfloor\boldsymbol{U}=\boldsymbol{U}^T\boldsymbol{G}_t\boldsymbol{U} \tag{4-15}$$

式中：

$$\boldsymbol{G}_t=E\left\lfloor\left(\boldsymbol{A}-E(\boldsymbol{A})\right)^T\left(\boldsymbol{A}-E(\boldsymbol{A})\right)\right\rfloor \tag{4-16}$$

E 表示数学期望，$n\times n$ 维非负定矩阵 $\boldsymbol{G}_t$ 为图像类间散布矩阵。

设共有 M 个训练样本，$\boldsymbol{A}_j(j=1,2\cdots,M)$ 为第 j 个 $m\times n$ 维图像矩阵。所有训

练样本的均值图像为 $\overline{\boldsymbol{A}}=\frac{1}{M}\sum_{j=1}^{M}\boldsymbol{A}_j$

则

$$\boldsymbol{G}_t=\frac{1}{M}\sum_{j=1}^{M}\left(\boldsymbol{A}_j-\overline{\boldsymbol{A}}\right)^T\left(\boldsymbol{A}_j-\overline{\boldsymbol{A}}\right) \tag{4-17}$$

由式(4-13)、(4-15)、(4-17)得

$$J(\boldsymbol{U})=\boldsymbol{U}^T\boldsymbol{G}_t\boldsymbol{U} \tag{4-18}$$

式(4-18)称为广义类间散布准则。最大化该准则的归一化向量 $\boldsymbol{U}_{opt}$ 称为最佳投影轴，其物理意义是，图像在 $\boldsymbol{U}_{opt}$ 轴上投影所得的特征向量的类间散布值最大。事实上， 该最佳投影轴就是图像类间散布矩阵 $\boldsymbol{G}_t$ 的最大本征值所对应的本征向量。

一般来说，在样本类别比较多的情况下，仅有一个最佳投影轴是不够的，需要寻找一组投影轴 $\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_p$ 满足如下条件：

$$\left.\begin{aligned}&\{\boldsymbol{u}_1,\cdots,\boldsymbol{u}_p\}=\arg\max\left(J(\boldsymbol{U})\right),\\&\boldsymbol{u}_i^T\boldsymbol{u}_j=0;i\neq j,\ i,\ j=1,2,\cdots,p,\end{aligned}\right\} \tag{4-19}$$

实际上正交投影轴 $\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_p$ 是 $\boldsymbol{G}_t$ 的前 p 个最大本征值所对应的本征向量。

对于给定的人脸图像 $\boldsymbol{A}$，可以得到一组投影特征分量：

$$\boldsymbol{b}_k=\boldsymbol{A}\boldsymbol{u}_k;k=1,2\cdots,p \tag{4-20}$$

这样得到的 $m\times p$ 维矩阵 $\boldsymbol{B}=\left[\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_p\right]$ 就是图像 $\boldsymbol{A}$ 的特征矩阵。

4.4.2 二维广义主成分分析

4.4.2.1 2DIMPCA 概述

IMPCA 算法按照水平方向将人脸信息压缩到一组列向量上，消除了人脸图像列的相关性，但是忽略了行的相关性。因此 IMPCA 提取的特征维数远高于普通 PCA，需要更多的存储空间，而且在后面分类阶段将需要更多的计算量。为了克服这个缺点，本文提出了 2DIMPCA[24]，基本思想是在水平和垂直方向上顺序执行 2 次 IMPCA，如图 4-9 所示。

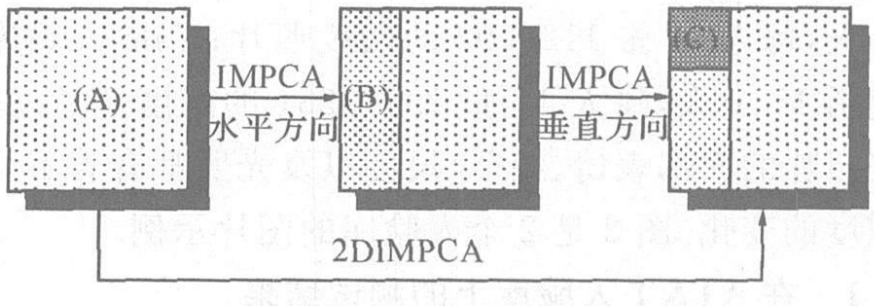


图 4-9　2DIMPCA 示意图

容易证明，对图像矩阵在垂直方向上进行 IMPCA，可以用对图像矩阵的转置在水平方向上进行 IMPCA 来替代。根据前一章介绍的方法，获得人脸图像的特征矩阵 $\boldsymbol{B}$，然后对 $\boldsymbol{B}^T$ 进行 IMPCA，寻找最佳投影轴 $\boldsymbol{V}$，提取特征 $\boldsymbol{C}^T=\boldsymbol{B}^T\boldsymbol{V}$。

与前面类似，定义类间散布矩阵

$$\boldsymbol{H}_t = E\left[\left(\boldsymbol{B}^T - E\left(\boldsymbol{B}^T\right)\right)^T\left(\boldsymbol{B}^T - E\left(\boldsymbol{B}^T\right)\right)\right] \tag{4-21}$$

和均值矩阵

$$\overline{\boldsymbol{B}^T} = \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{B}_j^T \tag{4-22}$$

则

$$\boldsymbol{H}_t = \frac{1}{M}\sum_{j=1}^{M}\left(\boldsymbol{B}_j^T - \overline{\boldsymbol{B}^T}\right)^T\left(\boldsymbol{B}_j^T - \boldsymbol{B}^T\right) \tag{4-23}$$

寻找投影轴$\boldsymbol{V} = \left[\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_q\right]$，使得$J(\boldsymbol{V}) = \boldsymbol{V}^T\boldsymbol{H}_t\boldsymbol{V}$最大。实际上，$\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_q$为$\boldsymbol{H}_t$的前$q$个最大本征值所对应的本征向量。因此，

$$\boldsymbol{C}^T = \boldsymbol{B}^T\boldsymbol{V} \Rightarrow \boldsymbol{C} = \boldsymbol{V}^T\boldsymbol{B} = \boldsymbol{V}^T\boldsymbol{A}\boldsymbol{U} \tag{4-24}$$

式中：$\boldsymbol{C}$为$p \times q$维特征矩阵。因为一般取p、q远小于m、n，所以$\boldsymbol{C}$的维数远小于$\boldsymbol{B}$的$m \times p$维和$\boldsymbol{A}$的$m \times n$维。

4.4.2.2 分类方法

采用常用的二维最小近邻分类法。设 2 个特征矩阵为

$\boldsymbol{C}_i = \left[\boldsymbol{c}_1^{(i)}, \boldsymbol{c}_2^{(i)}, \cdots, \boldsymbol{c}_q^{(i)}\right], \boldsymbol{C}_j = \left[\boldsymbol{c}_1^{(j)}, \boldsymbol{c}_2^{(j)}, \cdots, \boldsymbol{c}_q^{(j)}\right]$

定义它们之间的距离为

$$d\left(\boldsymbol{C}_i, \boldsymbol{C}_j\right) = \sum_{k=1}^{q}\left\|\boldsymbol{c}_k^{(i)} - \boldsymbol{c}_k^{(j)}\right\|_2 \tag{4-25}$$

式中：$\left\|\boldsymbol{c}_k^{(i)} - \boldsymbol{c}_k^{(j)}\right\|_2$表示$\boldsymbol{c}_k^{(i)}$、$\boldsymbol{c}_k^{(j)}$之间的欧氏距离。

设训练样本的特征矩阵为$\boldsymbol{C}_1, \boldsymbol{C}_2, \cdots, \boldsymbol{C}_M$，每一个样本都属于一个类别$\omega_k$，测试样本的特征矩阵为$\boldsymbol{C}_T$，如果满足

$$d\left(\boldsymbol{C}_T, \boldsymbol{C}_1\right) = \min_j d\left(\boldsymbol{C}_T, \boldsymbol{C}_j\right); \boldsymbol{C}_1 \in \omega_k \tag{4-26}$$

则测试样本$\boldsymbol{C}_T \in \omega_k$。

4.4.2.3 计算复杂度分析

比较 PCA、KPCA 以及 2DIMPCA 的计算复杂度。PCA 和 KPCA 的计算复杂度为$O(M^3)$与样本数 M 有关；而 2DIMPCA 的计算复杂度为$O(L), L = \max\{m, n\}$，仅与人脸图像的大小相关。一般而言，对于大型人脸数据库，m、n远远小于M，所以 2DIMPCA 的计算复杂度低，效率更高。

4.4.3 问题 3(c)求解：

人脸识别. 人脸识别是基于人的脸部特征信息进行身份识别的一种生物识别技术, 是计算机视觉与模式识别领域的一个重要研究问题. 已经证明了在不同的光照或表情变换条件下的人脸图像可以用一个低维子空间来近似, 取自多个人的一组人脸图像可以看作是 9 维线性子空间的并, 从而人脸识别问题等价于子空间聚类问题，我们使用经典的人脸识别算法，PCA 算法进行一个基本的实验，其分类的结果如图 4-10 所示。

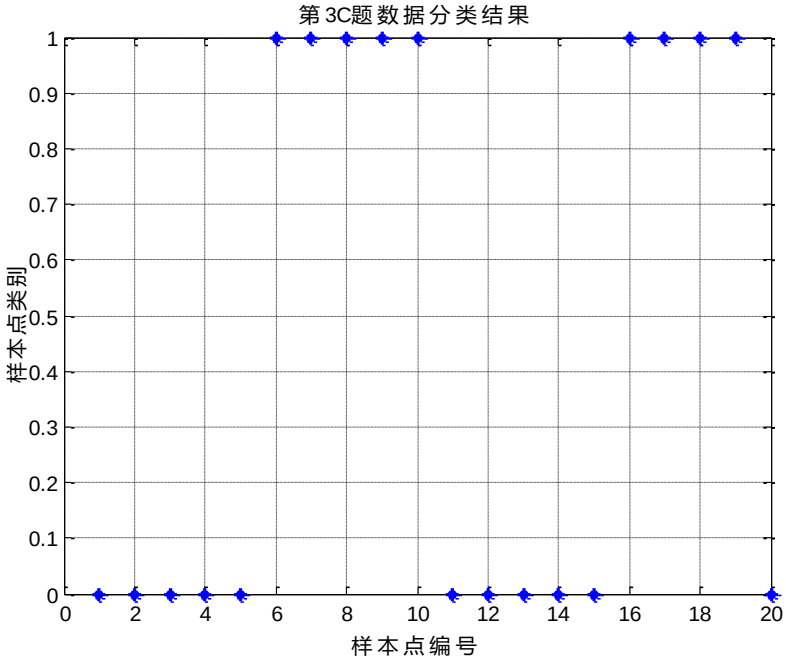


图 4-10 问题 3(c) PCA 分类结果

基本的 PCA 可以得到一个差强人意的分类结果，但是对于本题中光照强度不同时，总不能保证完美的分类数据，其原因在于此时 PCA 提取的特征从某种角度而言并不能很好的提取出不同光照条件下人脸的面部差异，因此我们引入了如上所述的一种改进的 PCA 方法 2DIMPCA，其分类结果如图 4-11 所示。

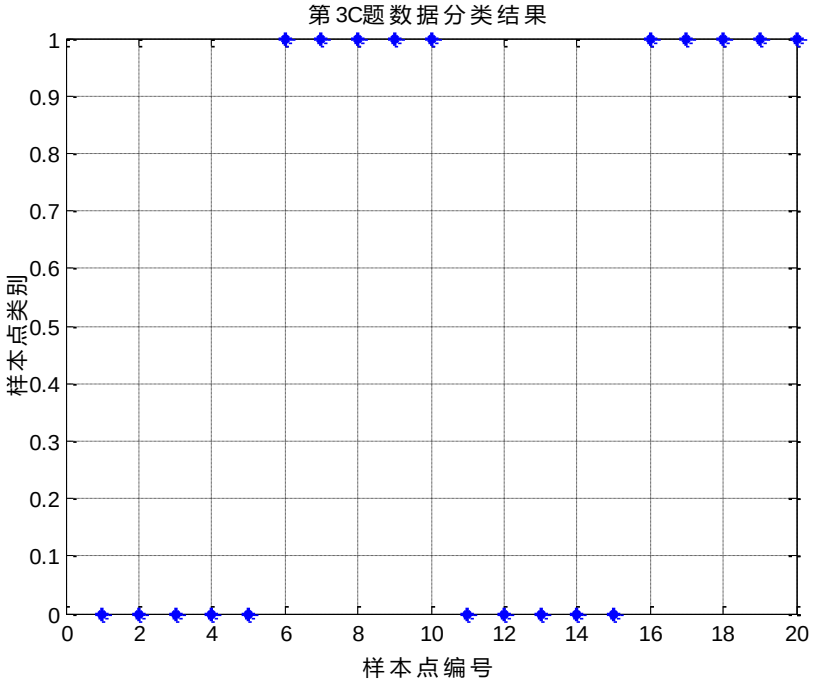


图 4-11 问题 3(c) 2DIMPCA 分类结果

将不同光照下人脸图像绘制如图 4-12 所示

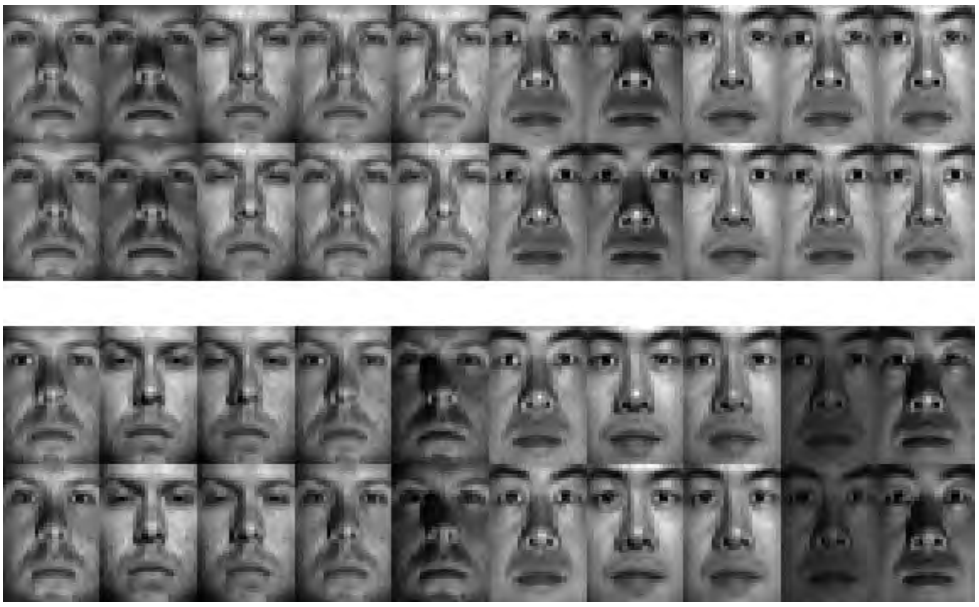


图 4-12 不同光照下人脸图像

结合分类结果和人脸图像，可以明显的看出图 1-5 和图 11-15 为同一个人，为中国人；图 6-10 和图 16-20 为同一个人，为外国人。根据题目要求分类标签如表 4-4 所示。

表 4-4 问题 3(c)分类标签表

0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1

在基于流形学习上，我们同时也采用了 SSC 算法进行同样的分析测试，我们发现在进行图像预处理之后（使用 RPCA 算法进行处理），在使用 SSC 算法进行分类，同样可以达到准确分类的效果,其分类结果如下图 4-13：

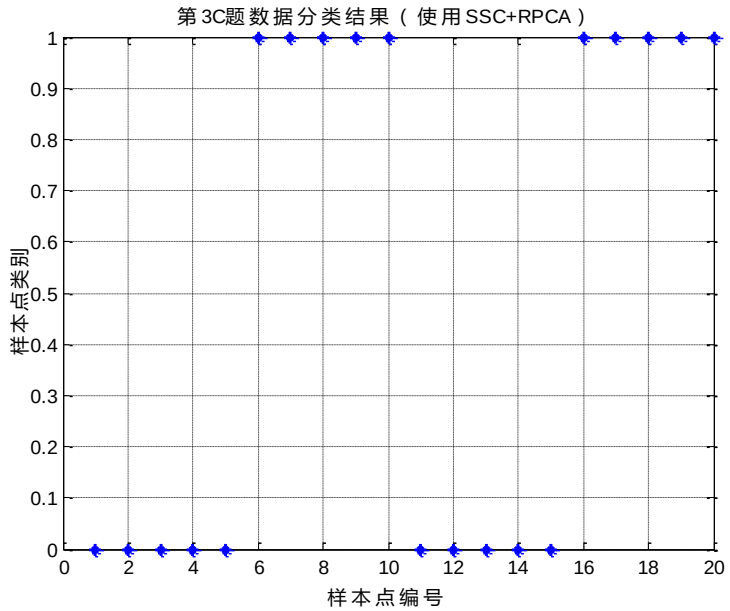


图 4-12 问题 3(c) SSC+GPCA 分类结果

显然 2DIMPCA 算法克服了在不同光照下特征提取不足的缺点，很好的分开了这类问题。同时，我们进行在时间上的对比，2DIMPCA 算法不仅在识别的精确度上，而且在运行时间上都显示去极高的效率，其分析结果如表 4-5 所示

表 4-5 不同识别方法对比

识别方法	$R_{\max}/\%$	t_E/s	t_C/s	t_τ/s
PCA	85.0	48.85	0.19	49.04
IMPCA	91.0	0.81	4.78	5.59
SSC	92.0	30.25	0.24	30.49
2DIMPCA	93.5	1.34	2.71	4.05

表中 $R_{\max}$ 为最大识别率，t_E 为特征提取时间，t_C 为分类时间，t_τ 为总时间。

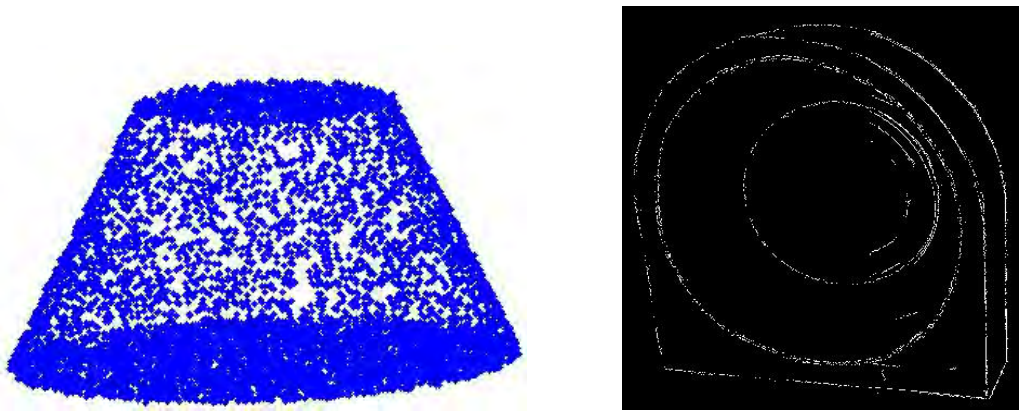
综合上面的表格和求解结果，我们可以得出结论，PCA 效果较差且时间长，IMPCA 识别效果一般，但时间较快，SSC+GPCA 识别效果很好但时间较长，最后 2DIMPCA 效果很好且时间较短。

对于光照不同的人脸识别，我们必须先使用相应的方法进行图像的预处理，如 RPCA 和 2DIMPCA，然后再进行特征提取和分类，对于分类部分使用谱分解方法可以很好的达到分类的效果。

五．问题四的建模与求解

5.1 问题重述

该题为多流形空间聚类问题，如图 5-1 所示。图 5-1(a)为一个圆台的点云，请将点按照其所在的面分开(即圆台按照圆台的顶、底、侧面分成三类)。图 5-1(b)是机器工件外部边缘轮廓的图像，请将轮廓线中不同的直线和圆弧分类，类数自定。



(a) (b)

图 5-1 问题四原始图形

问题分析：4(a)为一个圆台，由圆台顶、底和侧面三部分组成，3×8318 数据，为 8318 个三维数据，三个面分别为两个平面一个曲面，显然此平台的点为非线性，数据来源于三个不独立的流形，由于不都过原点，因此不属于子空间，属于仿射空间，三个曲面交叉点很多，属于多交叉的混合多流形问题。

典型的 SMMC 方法是从相似性矩阵的角度出发，充分利用流形采样点所内

含的自然局部几何结构信息来辅助构造更合适的相似性矩阵，从而发现正确的流形聚类。但是此处的流形集规模大，不同流形的交叉点数量众多，很容易出现过拟合，达不到分类要求。

采用改进的 SMMC 方法可以很好的解决这个问题。重新选择邻接矩阵是改进 SMMC 算法的核心问题，针对此圆台的结构特性，以局部切向子空间中心点之间的欧氏距离作为相似性的衡量标准，然后构建新的相似性矩阵，从而避免传统 SMMC 算法的不足之处。通过这种改进方法，得到正确的分类结果，同时降低了复杂度。

4(b)是由两个圆弧、两个半圆弧以及多条曲线组成的平面结构，2×2465 数据，为 2465 个二维数据，数据点为非线性，来源于多个不独立的流形，该数据不属于子空间，属于仿射空间，因此该问题属于多交叉的混合多流形问题。

半圆弧的切向角与直线相接近，但是属于不同的流形，容易导致不同流形的局部切向子空间相类似，使用传统的流形聚类算法容易产生错分的情况，不能准确聚类。本文提出 LSC 算法，避免边界处的干扰情况，可以得到正确的分类结果。

综上，为 4(a)圆台问题设计了改进的 SMMC 算法，为 4(b)机器边缘问题设计了 LSC 算法。

5.2 问题 4(a)

5.2.1 改进的 SMMC 算法

问题 4(a)数据集是一个大样本的数据集，我们根据问题二的求解经验，选择采用了 SMMC 算法进行了启发式聚类，但是基本的 SMMC 算法定义的自然局部集合结构信息来辅助构造的相似性矩阵并不能满足圆台两侧的聚类要求，所以，我们通过改变 SMMC 算法的局部集合构造信息，从点的信息转变成小的子块的信息，从而使相似性矩阵能更好的反映数据的特点，是侧面的数据能够精确的关联，经过改进后的 SMMC 算法，在我们 PID 参数调试策略的指导下，可以较为精确的完成数据集的分类。改进的 SMMC 算法流程如表 5-1 所示

表 5-1 改进的稀疏子空间聚类算法

算法 7：改进的谱多流形聚类（改进的 SMMC）
输入：原始数据集 X ,聚类数 k ,流形维数 d ,局部化模型数 M ,近邻点数 K ,调节参数 o . 算法过程： 1. 利用 MPPCA 训练 M 个 d 维的局部线性模型来近似潜在的流形数据; 2. 根据式（3-22）确定每个点的局部切空间; 3. 利用式（3-2）计算两个局部切空间之间的结构相似性;。 4. 利用式（3-6）计算相似性矩阵 $\boldsymbol{W}\in\Re^{N\times N}$,并计算对角矩阵 $\boldsymbol{D}$,其中 $d_{ii}=\sum_j w_{ij}$; 5.计算广义特征矩阵 $\left(\boldsymbol{D}-\boldsymbol{W}\right)u=\lambda\boldsymbol{D}u$ 最小个特征值对应的特征向量 $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k$; 6.利用 K-means 将 $\boldsymbol{U}=\{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k\}\in\Re^{N\times k}$ 的行向量分组为 k 个聚类。 输出：原始数据对应的聚类结果。

5.2.2 问题 4(a)求解

根据上述分析，导入题目的原始数据集，设置聚类数 k 为 3,流形维数 d 为 3,局部化模型数 M 为 9,近邻点数 K 为 5,调节参数 o 为 8。经过对参数的细微调整，并实现改进的 SMMC 算法，在 matlab 中运行，结果如图 5-2 所示：

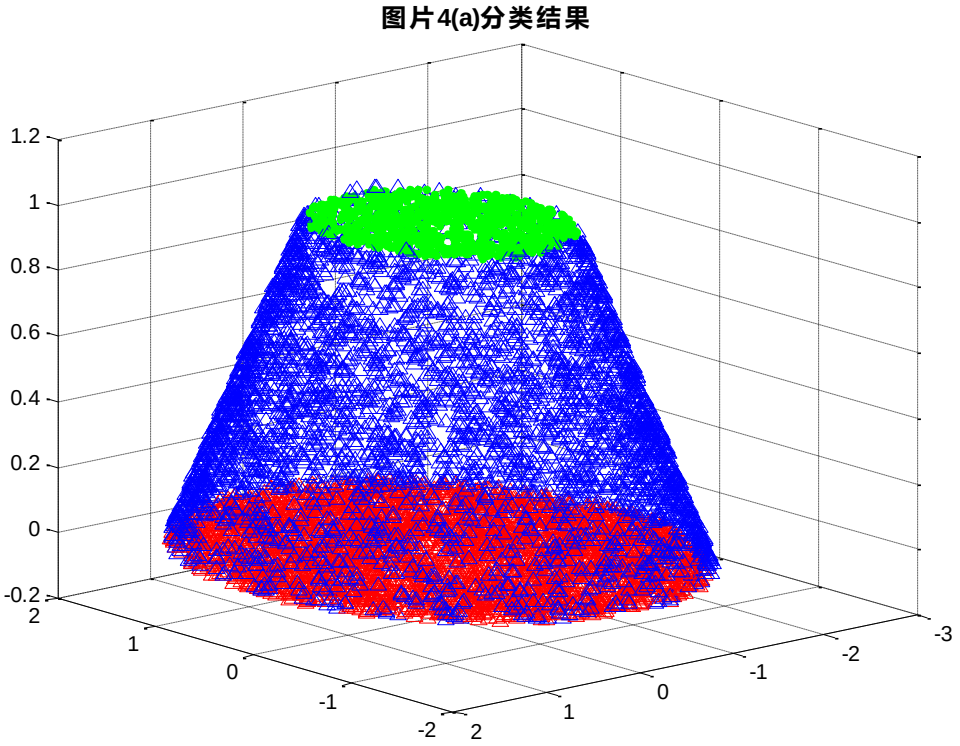


图 5-2 问题 4(a)聚类结果

如图 5-2 所示，算法清晰的分出了顶、底、侧面这三类，将不同的聚类用不同的颜色表示。实验证明改进的 SMMC 算法可以很好的解决问题。

5.3 问题 4(b)

根据数据的特点采用 LSC 算法来进行聚类，其算法思想是：既考虑局部近邻信息又充分考虑流形结构数据所内含的额外的结构信息来指导近邻点的选取，以尽量地从同一个潜在流形上选取近邻点而不是整个欧氏空间。

5.3.1 局部与结构一致性方法（LSC）

局部与结构一致性方法（Local and Structural Consistency，简记为 LSC）[25]来实现多交叉混合多流形问题。LSC 的基本思想是充分利用内含在流形结构数据中的几何信息来辅助确定近邻点，并进而发现正确的聚类。

5.3.2 对称型规范化谱聚类的潜力

当近邻点主要从同个潜在流形上选取时，对称型规范化谱聚类具有用于混合

流形聚类的潜力。然而，传统的谱聚类方法都是根据欧氏距离度量来选取近邻点，因此不适合于分组混合流形结构。为此，我们提出了局部与结构一致性准则(local and structural consistency criterion)来选取近邻点，它的基本思想是：既考虑局部近邻信息又充分考虑流形结构数据所内含的额外的结构信息来指导近邻点的选取，以尽量地从同一个潜在流形上选取近邻点而不是整个欧氏空间。然后，我们基于该准则设计了一个简单而有效的方法，即局部与结构一致性方法(LSC)，来分组或检测多个混合流形结构。

我们主要从近邻点的选取角度来考察用于混合流形聚类的潜力。首先，聚类的性能依赖于两个相互独立又相互关联的关键因素在于所采用的聚类算法和待聚类数据的分布特性或结构。当聚类算法固定后，聚类的性能主要由数据的特性所决定。传统聚类中直接在原始数据上采用某种聚类算法(如 K-means)来分组数据不同，谱聚类的分组过程是在原始数据的低维嵌入表示上进行的。然而低维嵌入数据是通过对近邻图上相似性矩阵或相似性矩阵的 Laplacian 矩阵做谱分析得到的，而近邻图又是由近邻点的选取决定的。因此上述事实表明谱聚类的性能在本质上依赖于如何选取近邻点。

可以得知，如果每个数据点的 k 个近邻点都来自于同一个潜在流形并且 k 足够大以保证能将同一个潜在流形上的所有数据点连通在一起，则对称型规范化谱聚类方法能完全正确地分组出不同的流形。

5.3.3 Arias-Castro 定理

对于一般的情况，即数据点的 k 个近邻点不是全都来自同一个潜在流形，我们有下面的定理，即 Arias-Castro 定理：

对 $m=1,\cdots,k$，令第 m 个聚类的指标集为 ζ_m 并记相似性矩阵 $\boldsymbol{W}$ 对应 ζ_m 的子矩阵为 $\boldsymbol{W}_m$。同时定义 $\hat{D}_i^m=\sum_{j\in\zeta_m} w_{ij}$ 并且 $\tilde{D}_i^m=\sum_{j\notin\zeta_m} w_{ij}$（分别称为 $i\in\zeta_m$ 的聚类内连接度和聚类间连接）。则在下述条件下，存在 k 个相互正交的向量 $\boldsymbol{r}_1,\boldsymbol{r}_2,\cdots,\boldsymbol{r}_k$ 使得对称型规范化谱聚类得到的低维嵌入表示满足：

$$\sum_{m=1}^{k}\sum_{i\in\zeta_m}\left\|\boldsymbol{y}_i^m-\boldsymbol{r}_m\right\|_2^2\le 8CN\delta^{-2}\left[k^2\varepsilon_1+k\varepsilon_2^2\right] \tag{5-1}$$

其中 $\boldsymbol{y}_i^m$ 表示第 m 个聚类中第 i 个样本的低维嵌入表示。

（1）存在 $\delta>0$ 使得对所有 $m=1,\cdots,k$，$\boldsymbol{W}_m$ 的第二大特征值大于 $1-\delta$；

（2）存在某个固定的 $\varepsilon_1>0$，使得对所有的 $m,n\in\{1,\cdots,k\}$ 且 $m\ne n$，有

$$\sum_{i\in\zeta_m}\sum_{j\in\zeta_n} w_{ij}^2\big/\hat{D}_i^m\hat{D}_j^n\le\varepsilon_1 \tag{5-2}$$

（3）存在某个固定的 $\varepsilon_2>0$ 使得对所有 $m=1,\cdots,k$ 和 $i\in\zeta_m$，有

$$\tilde{D}_i^m\big/\hat{D}_i^m\le\varepsilon_2\left(\sum_{s,t\in\zeta_m} w_{st}^2\big/\hat{D}_s^m\hat{D}_t^m\right)^{-1/2} \tag{5-3}$$

（4）存在某个常数 $C>0$ 使得对所有 $m=1,\cdots,k$ 和 $i,j\in\zeta_m$，有 $\hat{D}_i^m\ll C\hat{D}_j^m$。

从近邻点选取的角度来说，Arias-Castro 定理表明：当一个数据点的近邻点更多地来自其自身所在的潜在流形或聚类时，低维嵌入数据将会形成一些紧凑的“簇”聚集在 k 个良分离的数据点附近。显然谱聚类最后的 K-means 过程执行在这样结构的低维数据集上将给出几乎完全对应原始流形结构的分组结果。

为了分组混合流形结构数据，我们应该尽可能地从每个数据点自身所在的潜

在流形上选取近邻点。此外，传统的基于欧氏距离度量选取近邻点的谱聚类方法不能有效用于分组相互交叠或混合结构数据的原因在于，传统的方法基于欧氏距离度量选取近邻点，因此对不同流形相交区域附近的数据点，来自不同潜在流形的数据点很容易被选取为相互的近邻点从而将两个不同的潜在流形紧密地连接在一起，违背了 Arias-Castro 定理中的条件，从而会在不同的流形之间传播错误的信息。

5.3.4 LSC 方法及其计算复杂度分析

为了利用基于谱的聚类算法来分组混合结构数据，我们应该尽量地从同一个潜在流形上而不是整个欧氏空间中选取近邻点。我们的基本思想是：既然前提假设是数据位于多个光滑的潜在流形上，那么我们可以充分利用内含在流形结构数据中的几何信息来辅助确定近邻点，并进而发现正确的聚类。在不同潜在流形的相交区域，来自于同一个潜在流形的数据点有相似的局部切空间（即它们是结构一致的），而来自不同潜在流形的数据点的局部切空间是完全不同的。近邻点不仅要结构一致，还必须局部一致，即它们的空间位置应该相互靠近。

局部与结构一致性准则（LSC）可以用来选取近邻点，该准则基于下述距离度量来选取每个数据点的 K 个近邻点：

$$Dis(\boldsymbol{x}_i,\boldsymbol{x}_j)=\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|_2^2+\lambda*\left\|\Theta_i-\Theta_j\right\|_F^2 \tag{5-4}$$

其中 $Dis(\boldsymbol{x}_i,\boldsymbol{x}_j)$ 是新的距离度量，$\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|_2$ 是数据点 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 之间的欧氏距离，而 $\left\|\Theta_i-\Theta_j\right\|_F$ 是 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 的局部切空间（分别记为 Θ_i 和 Θ_j）之间的投影 **F** 范数距离，定义为

$$\left\|\Theta_i-\Theta_j\right\|_F=\sqrt{2\sum_{l=1}^{d}\sin^2\theta_l} \tag{5-5}$$

其中 $0\le\theta_1\le,\cdots,\le\theta_d\le\pi/2$ 是两个切空间 Θ_i 和 Θ_j 之间的主角度，λ 是一个平衡局部一致性和结构一致性的参数。

从式(5-4)中看出，通过局部与结构一致性准则来选取近邻点的关键是有效地估计每个数据点的局部切空间，我们可以通过训练 M 个混合概率主成分分析器来估计局部切空间。值得注意的是，这里在 EM 过程训练出模型参数 $\boldsymbol{\mu}_m$、$\boldsymbol{V}_m$、σ_m^2 后，我们通过最大化数据点的后验概率来估计局部切空间。具体地说，在最大化数据点 $\boldsymbol{x}_i$ 到第 j 个局部模型的后验概率的条件下将它划分到第 j 个局部分析器：

$$p\left(j\,|\,\boldsymbol{x}_i\right)=\max_m p\left(m\ \ \boldsymbol{x}_i\right) \tag{5-6}$$

其中后验概率 $p\left(m\,|\,\boldsymbol{x}_i\right)$ 为：

$$p\left(m\,|\,\boldsymbol{x}_i\right)=p\left(\boldsymbol{x}_i\,|\,m\right)\pi_m\Big/\sum_{m=1}^{M}p\left(\boldsymbol{x}_i\,|\,m\right)\pi_m \tag{5-7}$$

最后，样本点 $\boldsymbol{x}_i$ 的局部切空间由下式给出：

$$\Theta_i=span\left(\boldsymbol{V}_j\right) \tag{5-8}$$

当每个数据点的 K 个近邻点通过(1)中定义的距离度量确定后，可以采用和传统谱聚类相同的过程来得到最终的聚类结果，因此，我们将所提出的混合流形

聚类方法称为局部与结构一致性（LSC）。

LSC 算法的步骤如表 5-2 所示。

表 5-2 局部与结构一致性算法

算法8：局部与结构一致性算法（LSC）
输入：原始数据集 χ，聚类数 k，流形维数 d，局部化模型数 M，近邻点数 K，平衡参数 λ 算法过程: 1：估计每个数据点的局部切空间； 2：根据新的距离度量(1)确定每个数据点的 K 个近邻点，将近邻点连接起来构成近邻图 G 的边； 3：确定相似性矩阵 $\boldsymbol{W}$，当 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 之间有边连接时 $w_{ij}=1$，否则 $w_{ij}=0$； 4：提取规范化拉普拉斯矩阵 $\boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}$ 的最小 k 个特征值所对应的特征向量 $\boldsymbol{U}=[\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k]$，规范化 $\boldsymbol{U}$ 的行范数为 1 得到矩阵 $\boldsymbol{V}\in\mathbb{R}^{N\times k}$； 5：利用 K-means 将 $\boldsymbol{V}=[\boldsymbol{v}_1,\cdots,\boldsymbol{v}_k]\in\mathbb{R}^{N\times k}$ 的行向量分组为 k 个聚类。 输出：原始数据的 k 个聚类

由于聚类方法的计算复杂度直接影响到它的可适用性，下面我们对 LSC 方法的计算复杂度进行简要的分析。

LSC 方法的计算复杂度主要由三部分组成：估计每个数据点的局部切空间、根据局部与结构一致性准则计算新的距离度量(1)、利用谱方法进行聚类。,估计 N 个局部切空间 $\Theta_i, i=1,\cdots,N$ 的复杂度为 $O\left(NDM(t_1+dt_2)\right)$，其中 t_1 和 t_2 分别为 K-means 和 EM 过程收敛所需的迭代步数。在第二部分中，计算任意两个数据点之间的欧氏距离和投影 **F** 范数距离的复杂度分别为 $O(DN^2)$ 和 $O(M^2Dd^2)$，而在新的距离度量下搜索每个数据点的 K 个近邻点的复杂度为 $O(KN^2)$。第三部分利用谱方法进行聚类的复杂度 $O((N+K)N^2+Nk^2t_3)$，其中 t_3 为 K-means 在 k 维投影数据上收敛所需的迭代步数。因此，LSC 方法总的计算复杂度为：

$O(N^3+N^2D+N(DM(t_1+dt_2)+k^2t_3)+M^2Dd^2)$

这和传统谱聚类以及 mumCluster 方法、SMMC 方法的计算复杂度是可比较的。

5.3.5 问题 4(b)求解

LSC 方法中有三个可调节参数，即局部化模型数 M、近邻点数 K 和平衡参数 λ。下面我们考查这些参数的设置对 LSC 方法聚类性能的影响，并进而给出参数设置的一些指导建议。为平衡欧氏距离和投影 $\boldsymbol{F}$ 范数距离之间的量级，我们设置 $\lambda=\hat{\lambda}*mLnn(\chi)$，其中 $mLnn(\chi)$ 是 χ 中所有数据点到其第 K 个近邻点的平方欧氏距离的均值，$\hat{\lambda}$ 是一个标量。

LSC 在很大的参数选取范围内都得到了比传统谱聚类方法更好的性能。具体地说，有下列有价值的观测：

(1) 局部化模型数 M 越大，LSC 聚类的性能越好，随着局部化模型数增加，平均重构误差减小。这意味着对潜在流形局部线性块的近似越来越好，从而对每个数据点局部切空间的估计也越来越可靠，进而使得依赖于局

部切空间正确估计的 LSC 方法的性能变好。

(2) 当近邻点数 K 既不太大也不太小时，LSC 的性能在一个很大的参数选取范围内都是稳健的。其原因在于，当 K 值太小时会出现很多不连通的子聚类，而当它太大时局部限制会逐渐丧失。

(3) 当调节参数 $\hat{\lambda}$ 相对大而不是特别大的时候，LSC 的性能很好。其原因在于，$\hat{\lambda}$ 越大结构一致性的比重越大而局部一致性的比重越来越小。这一事实表明,我们应该尽量地平衡局部一致性和结构一致性。

基于上述观测和分析，我们可以给出参数选取的一些指南。作为一般的推荐，我们建议设置 $M=\lceil N/(7d)\rceil$，$K=2\lceil\log(N)\rceil$，$\hat{\lambda}=1.2$。同样地，当所有潜在流形都是线性时，可以采用一个较小的 M，例如 $M=3k$。

根据上述分析，导入题目的原始数据集，设置聚类数 k 为 4，流形维数 d 为 2，局部化模型数 $M=3k$ 为 12，近邻点数 K =6，平衡参数 λ 为 1.2，经过对参数的细微调整，并实现 LSC 算法，在 matlab 中运行，结果如图 5-3 所示.

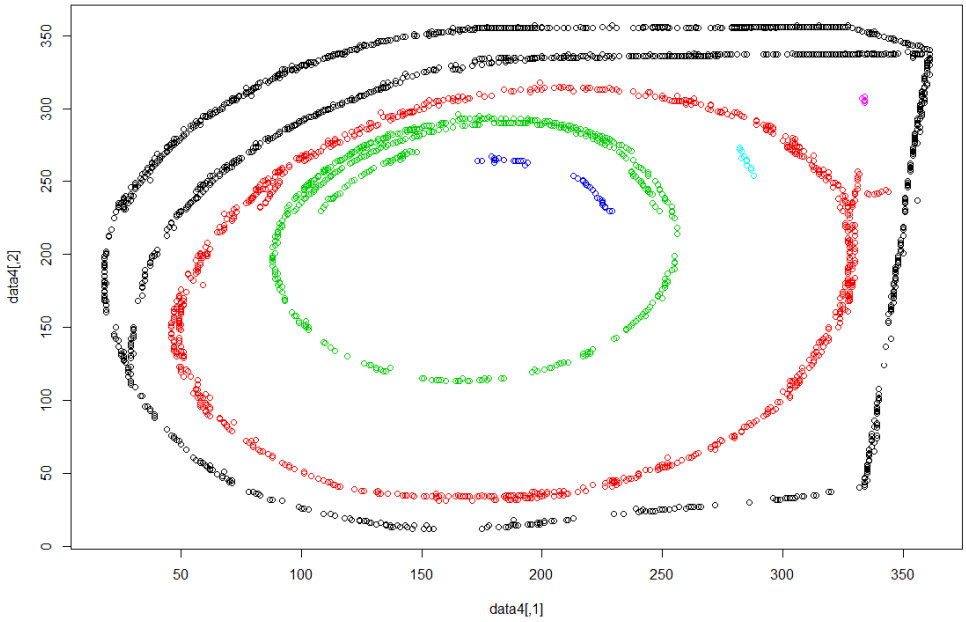


图 5-3 LSC 算法聚类结果

实验结果清晰的分出了不同的直线和圆弧，将不同的聚类用不同的颜色表示。但有一些干扰的小段曲线识别错误，并没有完全准确的聚类。

这样的流形结构在理论上可以达到我们的聚类的目的，但是从我们聚类的结果来看，即使这样理论上可行的算法在实际的运用过程也只能相对精确的达到聚类的结果。另一方面，我们也设想通过构造新的局部集合构造信息，来得到更加符合数据特征的相似度矩阵。其背后的思路和 LSC 算法是一致的，限于时间也没能做出绝对精确的结果。

六．模型的优缺点

本文根据不同的数据集合的特点，选用了最适应的基于流形学习的聚类算法，

本文所采用的主要是 SSC，SMMC，MUM，LSC 等四种算法。其基本的思想都是子空间聚类，也就是子空间分割，假设数据分布于若干个低维子空间的并，是将数据按某种方式分类到其所属的子空间的过程。通过子空间聚类，可以将来自同一子空间中的数据归为一类，由同类数据又可以提取对应子空间的相关性质。SSC 算法是一种基于稀疏表示的子空间聚类方法，在算法约束条件下，可以很好的处理复杂数据集的聚类问题，在高维度数据，运动分割，人脸识别几个方面都有很好的表现，如果配合一些特征提取的算法，可以达到非常好的聚类效果。但是，SSC 算法对于数据的污染的处理能力较弱，在人脸识别的题中，如果不进行 RPCA 的预处理，其效果并不是很理想，同时，SSC 算法的计算量也相对较大。

MUM 方法通过“分而治之”的策略直接从整个数据中找出流形交叠的部分并拆开不同的流形结构构造出更忠实于流形结构的近邻图实现混合流形聚类;SMMC 方法从相似性矩阵的角度出发,充分利用流形采样点所内含的局部几何结构信息来辅助构造更合适的相似性矩阵实现混合流形聚类;LSC 方法从近邻点选取的角度出发,在局部与结构一致性准则的指导下尽量从同一个潜在流形结构中选取近邻点实现混合流形聚类。SMMC 算法是一种非常高效的基于流形的聚类算法，通过构造合理的流形结构可以很好的解决很多的复杂数据的聚类，但是，SMMC 对于参数的敏感性太强，参数的轻微变动都可能得到完全不理想的结果,所以对于 SMMC 算法的使用需要对参数调节算法有很高的要求,另一方面，SMMC 算法中流形结构的定义也是一个非常有挑战性的问题，其不但影响着算法的性能，还影响着算法的计算复杂度，往往制约着 SMMC 算法的使用范围。MUM 算法和 SMMC 算法类似，也有着对参数的很强的依赖性，但是在某些特殊的场景，MUM 有着其独特的优势。LSC 算法在特殊的数据分布中有着其难以替代的优势,但是其同样存在着依赖参数的问题。

七. 总结与展望

7.1 总结

进入信息时代, 人们生活的各个方面都离不开数据的处理, 对大规模数据的分析与处理在科学研究领域占据着越来越重要的地位. 数据的维数之高结构之复杂给数据的分析与处理带来了一定的困难。基于流形学习的聚类算法，是当前计算机模式识别领域的新思路。本文通过引入基于流形学习的聚类方法来解决复杂数据集的聚类问题，主要采用了 SSC,SMMC,MUM,LSC 等算法以及相应的改进算法解决了一系列不同数据特征，不同实际场景的聚类问题。聚类的场景包括了线性子空间和非线性子空间，包括了独立和不独立的空间数据集合，并结合求解过程中涉及的多参数调节问题，提出了一种类似于 PID 调节的参数调节算法，并在本文的参数调节中得以发挥作用。

在第一题中，我们采用了 PCA 算法进行了一个基本求解，并从结果中得到启发，采用了基于流形学习的 SSC 算法来就行数据的聚类，通过调节 SSC 的参数，得到合理的相似度矩阵，通过 K-means 聚类算法聚类得到了高精度的聚类结果，分类结果为前 40 个点和最后 60 个点为一类，中间 100 个点分为一类。

在第二个问题中，需要分类四种不同类型的数据集合，我们引入了 SMMC 算法来进行聚类，其核心思想是：从相似性矩阵的角度出发，充分利用流形采样点所内含的自然局部集合结构信息来辅助构造更合适的相似性矩阵并而发现正

确的流形聚类。相对而言，本题中的四个数据集合可以利用基本局部几何结构构造辅助信息来聚类得到正确的结果，但是，参数的合理调节确实一个难点，它代表这不同的相似度矩阵构造置信度，甚至是直接影响到分类结果的好坏，我们根据 SMMC 算法的经验公式，在合理范围内设定初始的算法参数，并根据其经验的取值范围，创新性的提出和归纳出了一种基于 PID 调节思想的参数调节策略，使我们每一次的参数组合都向着更优的聚类结果逼近。因此，能够快速合理的逼近所需要的参数组合，得到所需要的聚类结果。同时，考虑到聚类算法的效能，我们在文中针对 SMMC 算法做了算法的时间复杂度的分析。

在第三个问题中，首先是一个大样本量的线性交叉的数据集合，针对其数据的特点我们引入了多流形聚类方法（MUM）。它直接从整个数据中找出流形交叠的部分并拆开不同的流形结构,从而构造出更忠实于流形结构的近邻图来实现混合流形聚类，在进行基本的图像处理后，通过我们提出的 PID 参数调节策略，得出了理想的分类结果，在第二个运动分割问题中，将视频序列分成多个时空区域,即把视频里做刚性运动的物体上的特征点进行聚类, 使得每一类对应于一个独立运动的物体，根据这样的思路，我们采用了 SSC 算法来进行了启发式的聚类，根据聚类的结果不能很好的分类局部的小的数据子块，我们引入了加权 SSC 算法来进行运动分割，这种以高斯相似度作为权重的算法很好的解决了局部子块分类不准确的问题。在全部 31 帧数据中都能准确的聚类好数据，实现了运动分割的预期效果，分类结果为 1 到 138 个数据属于同一流形，属于一个类，为背景信息；39 到 214 个数据点属于同一流形，属于一个类，为公共汽车；215 到 297 个数据点属于同一流形，属于一个类，为小汽车。针对 3C 中的人脸识别问题，我们首先采用了经典的 PCA 算法来进行启发式聚类，但是对于光照不同的特征提出不是很准确，导致了聚类结果的精度不高。因此，我们采用了 2DIMPCA 算法来进行了人脸识别，达到了完全精确的聚类效果。同时，我们也采用了 RPCA 先进行图像的预处理后，在采用 SSC 算法进行人脸识别，也可以达到同样的效果，分类结果为 1 到 5,11 到 15 属于一类，为外国人。6 到 10,16 到 20 属于一类，为中国人。两种算法的互相借鉴和组合也给不同条件下的人脸识别算法提供了新的思路。

在第四个问题中，第一个数据集是一个大样本的数据集，我们根据第二题的求解经验，选择采用了 SMMC 算法进行了启发式聚类，但是基本的 SMMC 算法定义的自然局部集合结构信息来辅助构造的相似性矩阵并不能满足圆台两侧的聚类要求，所以，我们通过改变 SMMC 算法的局部集合构造信息，从点的信息转变成小的子块的信息，从而使相似性矩阵能更好的反映数据的特点，是侧面的数据能够精确的关联，经过改进后的 SMMC 算法，在我们 PID 参数调试策略的指导下，可以较为精确的完成数据集的分类。在第二问中，我们根据数据的特点采用了 LSC 算法来进行聚类，其算法思想是：既考虑局部近邻信息又充分考虑流形结构数据所内含的额外的结构信息来指导近邻点的选取，以尽量地从同一个潜在流形上选取近邻点而不是整个欧氏空间。这样的流形结构在理论上可以达到我们的聚类的目的，但是从我们聚类的结果来看，即使这样理论上可行的算法在实际的运用过程也只能相对精确的达到聚类的结果。另一方面，我们也设想通过构造新的局部集合构造信息，来得到更加符合数据特征的相似度矩阵。其背后的思路和 LSC 算法是一致的，限于时间也没能做出绝对精确的结果。

本文有以下的创新点，首先，我们引入了 SSC 算法去解决高维数据，运动分割，采用了 SSC+RPCA 算法解决了有噪声（光照不同）情况下的人脸识别问题。

然后，我们引入了 SMMC 算法去解决聚类问题，引入了 MUM 算法去解决高密度有汇聚数据的聚类问题。更为重要的是，我们根据实际的经验提出了一套基于 PID 思想的算法参数调节策略，很好的解决了算法的调节问题。针对 SMMC 所依赖的流形结构，我们采用了定义新的流形解决了自然流形无法聚类的问题，最后我们针对特殊的复杂的数据采用了 LSC 算法去定义一个近邻点的流形，可以有效的解决特殊的分类问题。

综上，基于流形学习的算法作为聚类算法的新思路，让我们从一个全新的视角去看待复杂聚类问题，当然，基于流形学习的聚类算法其本身还很多可以挖掘的地方，诸如如何提高算法的参数鲁棒性，如何根据数据集的特点给出一套适应性强的流形结构来解决不同的聚类问题。问题需要探索和实践，希望在以后的工作能够进一步挖掘出基于流形聚类算法的新思路。

7.2 展望

流形学习作为分析和处理高维、非线性数据的一种有效工具,为实现“不分时间和地域,可以有效地利用数据和信息”提供了解决思路,已经成为数学、计算机科学以及工程应用等方面的交叉研究课题,同时也是机器学习、人工智能领域的一个研究热点。

现有多流形建模和方法大多还局限于某些特定的情形,应该在更一般的框架下深入研究多流形建模与应用,并解决其内在的若干问题。例如,现有流形建模方法大多假设潜在的流形具有相同的本征维数,并且本征维数和聚类数目是事先给定的,而这并不符合实际情况,因此为了使得多流形模型适用于更一般的真实任务,应该研究混合维数下的混合流形建模问题,并且让算法自适应地获取潜在流形的本征维数和数目。另一方面,现有方法往往具有较高的计算复杂度(样本数 N 的三次方的量级)并且对噪声或离群点敏感,这就使得它们对大容量(large-scale)的真实采样往往效果不佳,严重制约了多流形建模的实用化,因此　可以对大规模、噪声复杂、相对稀疏采样下的数据进一步进行探讨和分析。

八．参考文献

[1] Seung H S, Lee D D. Cognition - The Manifold Ways of Perception[J]. Science, 2000, 290(5500):2268--2269.

[2] Tenenbaum J B, de Silva V, Langford J C. A Global Geometric Framework for Nonlinear Dimensionality Reduction[J]. Science, 2000, 290(5500):2319--2323.

[3] Roweis S T, Saul L K. Nonlinear Dimensionality Reduction by Locally Linear Embedding[J]. Science, 2000, 290(5500):2323--2326.

[4] 陈维桓. 微分流形初步 [M]. 第二版. 北京: 高等教育出版社, 2001.

[5] Maaten L v d, Postma E, Herik H v d. Dimensionality Reduction:A Comparative Review[R]. Holland: Tilbrug University, Technical Report, TiCC-TR 2009-005, 2009.

[6] Roweis S, Saul L K, Hinton G E. Global Coordination of Local Linear Models[C]. Advances in Neural Information Processing Systems 14. Cambridge, MA: MIT Press, 2002:889--896.

[7] Tipping M E, Bishop C M. Probabilistic Principal Component Analysis[J]. Journal of the Royal Statistical Society Series B-Statistical Methodology, 1999, 61:611--622.

[8] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning[M]. New

York: Springer Verlag, 2001.
[9] Murty M, Jain A, Flynn P. Data Clustering: A Review[J]. ACM Comput. Surv., 1999, 31(3):264--323.
[10] Xu R, Wunsch D I. Survey of Clustering Algorithms[J]. IEEE Transactions on Neural Networks, 2005, 16(3):645--678.
[11]E. Elhamifar, R. Vidal. Sparse subspace clustering. IEEE Conference on Computer Vision and Pattern Recognition, 2009.
[12] Wang Y, Jiang Y, Wu Y, et al. Multi-Manifold Clustering[C]. Proceedings of the Eleventh Pacific Rim International Conference on Artificial Intelligence. New York: Springer Verlag, 2010:280--291.
[13] Tipping M E, Bishop C M. Mixtures of Probabilistic Principal Component Analyzers[J]. Neural Computation, 1999, 11(2):443--482.
[14] Saul L K, Roweis S T. Think Globally, Fit Locally: Unsupervised Learning of Low DimensionalManifolds[J]. JournalofMachineLearningResearch, 2004, 4(2):119- -155.
[15] Zhang Z Y, Zha H Y. Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment[J]. SIAM J. Scientific Computing, 2005, 26(1):313--338.
[16] Golub G, Loan C V. Matrix Computations[M]. 3rd. Baltimore, Maryland: John Hopkins University Press, 1996.
[17] Yan J Y, Pollefeys M. A General Framework for Motion Segmentation: Independent,Aarticulated,Rigid,Non-rigid,DegenerateandNon-degenerate[C].European Conf. on Computer Vision. 2006:94--106.
[18] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(2):218–233, 2003.
[19] R. Vidal. Subspace clustering. IEEE Signal Processing Magazine, 28(2):52–68, 2011.
[20] J. Shi and J. Malik, “Normalized cuts and image segmentation,” IEEE Transactions Pattern Analysis Machine Intelligence, 22(8):888–905, 2000.
[21] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(1):171–184, 2013.
[22] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(11):2765–2781, 2013.
[23] Y. Wang, Y. Jiang, Y. Wu, and Z. Zhou. Spectral clustering on multiple manifolds. IEEE Transactions on Neural Networks, 22(7):1149–1161, 2011.
[24] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan, Multi-task low rank affinity pursuit for image segmentation, ICCV, 2011.
[25] C. Lang, G. Liu, J. Yu, and S. Yan, Saliency detection by multitask sparsity pursuit, IEEE Transactions on Image Processing, 21(3): 1327–1338, 2012.

九．代码目录

test_1.m——问题一代码
test_2a.m——问题二(a)代码
test_2b.m——问题二(b)代码
test_2c.m——问题二(c)代码
test_2d.m——问题二(d)代码

test_3a.m——问题三(a)代码
test_3b.m——问题三(b)代码
test_3c.m——问题三(c)代码
test_4a.m——问题四(a)代码
test_4b.m——问题四(b)代码
文件 SMMC——SMMC 算法代码
文件 SSC-ADMMv1.1——SSC 算法代码