

(由组委会填写)



1. 问题一中采用谱聚类的独立子空间分割模型, 将独立子空间中的高维数据分成两类; 进一步采用基于共享近邻的自适应谱聚类模型, 克服了传统谱聚类模型依赖于人工设定参数的限制, 分类结果见附录 1。
2. 为了解决谱聚类模型无法解决有交叉区域的分割问题, 在处理问题二中四个低维空间的聚类问题时采用了基于谱多流形算法的聚类模型, 该模型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, 进而发现正确的流形聚类; 分别采用 F-measure 值、RI 值、NMI 值作为评价测度, 仿真结果表明, 谱多流形聚类算法明显优于谱聚类算法; 分析了算法主要参数对分类性能的影响, 并分析了该算法的复杂度。

3. 对于子空间聚类在实际中的应用, 首先通过分离宽十字交叉点来验证谱多流形聚类对于交叉区域的鲁棒性; 为建立不同帧上同一特征点的运动联系, 建立了基于最小矢量差的谱多流形聚类模型, 分类结果见附录 2; 将人脸图像向量数据转化为矩阵, 加强同一人脸图像上像素点之间的关联性, 采用谱多流形聚类算法将所有人脸图像分成两类, 结果表明, 该算法将人脸图像按照地缘特征严格地分成两类, 即 10 个欧美人脸图像为一组, 10 个亚洲人脸图像为另外一组, 分类结果见附录 3, 同时针对一般图片的高像素问题, 给出了降低算法复杂度的数据处理方法并通过分类结果验证其有效性。
4. 问题四为混合多流形聚类在实际中的应用问题, 采用谱多流形聚类算法能够利用点云特别是交叉点云的局部空间的几何结构信息, 克服了单纯的基于距离的远近分类点集, 将圆台点云有效地分割为顶、底以及侧面三类; 对于工件轮廓线的分类问题, 提出了一种改进的分部谱聚类算法, 通过粗聚类划分连通集、细聚类划分流形, 能够明显提升轮廓线分类的准确率, 最终轮廓线被分成 10 类, 包括 4 类直线和 6 类圆弧线。

关键词: 高维, 局部空间, 子空间, 谱多流形聚类, 分部谱聚类

目录

1	问题重述.....	1
2	基本假设.....	2
3	主要符号说明.....	2
4	问题分析.....	2
5	谱聚类的相关知识.....	5
5.1	谱聚类理论基础.....	5
5.1.1	图的基本概念.....	5
5.1.2	相似矩阵和图的 Laplacian 矩阵.....	6
5.2	谱聚类评价测度.....	8
6	问题一：独立子空间高维数据聚类问题.....	11
6.1	模型一：基于谱聚类的独立子空间分割模型.....	11
6.1.1	模型建立.....	11
6.1.2	模型求解.....	12
6.2	模型二：基于共享近邻的自适应谱聚类模型.....	15
6.2.1	模型准备.....	15
6.2.2	模型建立与求解.....	17
7	问题二：低维空间中的子空间与多流形聚类问题.....	18
7.1	模型三：基于谱多流形算法的聚类模型.....	18
7.1.1	模型建立.....	18
7.1.2	模型求解.....	18
7.1.3	问题二分类效果及分析.....	20
8	问题三：子空间聚类在实际中的应用.....	25
8.1	宽交叉十字形数据点群的分类问题.....	25
8.2	多帧图像运动特征点轨迹分类问题.....	26
8.2.1	模型四：基于最小矢量差的谱多流形聚类模型.....	28
8.2.2	模型求解与讨论分析：.....	29
8.3	人脸图像分类问题.....	31

8.3.1	数据预处理.....	31
8.3.2	问题求解.....	32
8.3.3	求解过程的简化：数据再处理.....	33
9	问题四：实际应用中的多流形聚类问题的探讨	36
9.1	三维点云的分类问题.....	37
9.1.1	谱多流形聚类方法分类圆台点云.....	37
9.1.2	实验结果与分析.....	38
9.2	工件轮廓线的分类问题.....	38
10	总 结.....	43
	参考文献.....	44
	附 录.....	45

1 问题重述

在如今这个信息爆炸的时代,海量的数据不断产生,迫切需要对这些大数据进行有效的分析,以至数据的分析和处理方法成为了诸多问题成功解决的关键,涌现出了大量的数据分析方法。几何结构分析是进行数据处理的重要基础,已经被广泛应用在人脸识别、手写体数字识别、图像分类、等模式识别和数据分类问题,以及图象分割、运动分割等计算机视觉问题中。更一般地,对于高维数据的相关性分析、聚类分析等基本问题,结构分析也格外重要。

数据结构分析的难点在于以下几个方面:一是从线性到非线性的扩展,流形学习的出现,很好地解决了具有非线性结构的样本集的特征提取问题。然而流形学习方法通常计算复杂度较大,对噪声和算法参数都比较敏感,并且存在所谓的样本溢出问题;二是流形或子空间从一个到多个的扩展,子空间聚类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,然而有些实际问题的数据并不符合混合子空间结构的假设,假设数据的结构为混合多流形更具有一般性,由于混合流形不全是子空间的情况,数据往往具有更复杂的结构,分析这种数据具有更大的挑战性。

本文主要解决以下问题:

1. 将 1.mat 中采样于两个独立子空间的高维数据分成两类。
2. 处理四个低维空间中的子空间聚类问题和多流形聚类问题:
 - (a) 将两条交点不在原点且互相垂直的两条直线分为两类;
 - (b) 将不满足独立子空间关系的一个平面和两条直线分为三类;
 - (c) 将两条不相交的二次曲线分为两类;
 - (d) 将两条相交的螺旋线分为两类。
3. 解决三个实际应用中的子空间聚类问题:
 - (a) 将宽交叉十字上的数据点分成两类;
 - (b) 视频的一帧中,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,将这些特征点轨迹分成三类;
 - (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),将这 20 幅图像分成两类。

4. 解决两个实际应用中的多流形聚类问题：

- (a) 将圆台中的点云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)；
- (b) 将机器工件外部边缘轮廓线中不同的直线和圆弧分类，类数自定。

2 基本假设

-
- 1、假设题目所给数据都是有效数据，即不存在噪声。
-

3 主要符号说明

V	图顶点的集合
W	边权值矩阵
L	拉普拉斯矩阵
D	度矩阵
S	相似矩阵
Θ_i	局部切空间
k	分类数
K	近邻点数

4 问题分析

在过去的几十年里，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。近年来，聚类正在蓬勃发展，聚类分析在多个研

究领域都有了很大贡献，包括数据挖掘、机器学习、统计学、空间数据库技术、生物信息学以及市场营销等。

问题一给出了一组高维数据，代表高维空间中的点分布情况。当子空间独立时，聚类问题相对容易，但是当数据维数较高的时候，其受到随机因素的影响较大，同时数据处在高维空间中，分布较为离散。因而可采用先降维后聚类的方式，即先对高维数据进行降维预处理，再用谱聚类算法对数据进行聚类。而谱聚类算法中，相似矩阵的构建是一个关键，在相似矩阵的构建过程中如果能有效利用数据局部近邻信息，更有利于提高谱聚类算法性能。

问题二要求处理四个低维空间中的子空间聚类问题和多流形聚类问题。简单观察之后，我们可以将问题归结为有相交区域和无相交区域的数据处理问题；考虑到问题中出现多流形的情形，可以采用基于多流形分类的算法进行处理。对于无相交区域的数据处理，往往依赖于欧氏距离的聚类算法就可以实现有效的聚类；而一旦出现相交区域时，传统的聚类算法往往在交叉点位置判断出现误差，造成最终的结果偏差较大，需要更进一步挖掘数据的信息。

问题三抽取出了现实生活的三个例子进行讨论：宽十字交叉数据的分割、运动轨迹检测以及人脸图像分类问题。分开来看的话，宽十字交叉数据的分割实际上是垂直交叉直线的扩展，不同的是交叉区域的增大造成了分离难度的增大，需要使用对交叉区域鲁棒性较强的算法，才能实现有效分割；运动轨迹检测问题给出了不同帧的特征点分布数据，事实上，如果单纯地按帧分割特征点，虽然能够在一定程度上实现正确分割，但是一旦出现交叉区域，分割正确性无法保证，同时也无法把握物体的运动规律，因此需要以特征点的额外信息（如帧变化）进行分割；人脸分割问题可以看成是二维矩阵的聚类问题，因此并没有理论上的难度，但考虑到算法的复杂度，可以采取数据降维提高运行速度。

问题四中涉及到的两个实际应用为混合多流形聚类问题，自然而然地需要采用多流形聚类算法进行求解。不同的是，相对于圆台的简单结构，工件的轮廓线由于相互靠近或者连接而显得更加复杂，这种不同类之间的非明显性差异造成了分割的难度，可以首先将这种干扰进行筛除，然后再按照之前的步骤完成类别分割。

5 谱聚类的相关知识

谱聚类算法是解决子空间分割和混合多流形问题的经典方法之一^[1]，它可以归为图论的分割算法，实现简单并且其往往优于传统的聚类算法如 K-means 算法^[2]。由于图划分问题的本质，求取图划分准则的最优解是一个 NP-hard 问题^[3]，一个很好的求解方法是考虑问题的连续放松形式，便可将原问题转换成求解相似矩阵或 Laplacian 矩阵的谱分解^{[4][5]}。而通过对相似矩阵进行特征分解，获得某个聚类准则在放松的连续域中的全局最优解，使得谱聚类算法不用知道数据在样本空间上的全局分布，只需考虑数据点个数，且与数据点的维数无关，不仅可以避免特征向量的维数太高所造成的奇异性问题，而且使谱聚类划分准则相关理论得到解释，这也是谱聚类算法成功的主要原因。

聚类结果是否准确直接影响聚类分析的后续步骤，且聚类准确性评价测度的选择对谱聚类算法是非常重要的。本章主要介绍谱聚类算法基本理论和评价测度的相关知识。

5.1 谱聚类理论基础

5.1.1 图的基本概念

由若干个不同顶点与连接其中某些顶点的边所组成的图形就称为图，图 5-1 表示的是一个简单的图。在图论中，一幅图 G 可以写成 $G=(V,E)$ ，其中 V 代表的是顶点的集合，假设一幅图有 n 个顶点，则顶点集可写为 $V(G)=(v_1,v_2,...,v_i,...v_j,...v_n)$ 。 E 代表的是边的集合，图 5-1 中 e_{ij} 即为图的边，边表示两个顶点间的联系，假设图中有 m 条边，那么边集 E 可表示为 $E(G)=(e_1,e_2,...e_m)$ 。图分为无向图和有向图，有向图的边都有方向，无向图中的边均无方向。

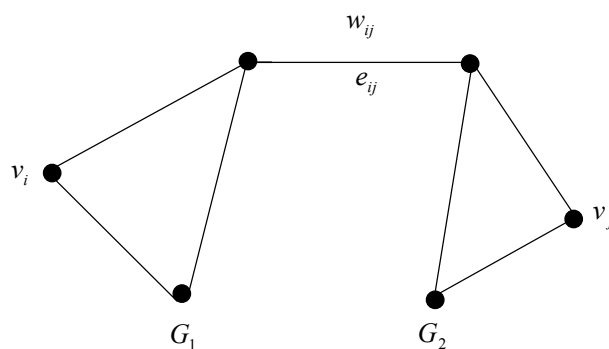


图 5-1 一个简单的图

如果给图的边加上权值，图就成为加权图。图 5-1 中 w_{ij} 是图中连接两顶点 v_i ， v_j 的权值，它反映了顶点 v_i 与 v_j 之间的相似程度，所有存在连接的顶点之间连接权值构成图的权值矩阵 $W = \{w_{ij}\}$ ，图 G 就可以表示为 $G = (V, E, W)$ ，若 $w_{ij} = 0$ ，表示顶点 v_i 和 v_j 之间不存在连接，若图 G 为无向图，那么 $w_{ij} = w_{ji}$ 。接下来介绍图的几个重要基本概念。

(1) 顶点的度

顶点 $v_i \in V$ ，则顶点 v_i 的度定义为 $d_i = \sum_j w(i, j)$ ，即为顶点 v_i 与其相连接顶点之间边的权值总和。

(2) 子图

如果对于图 $G = (V, E)$ 与 $G_1 = (V_1, E_1)$ ， G_1 的顶点集是 G 的顶点集的子集， G_1 的边是 G 的边集的子集，则 G_1 是 G 的子图，记为 $G_1 \in G$ 。

(3) 图的阶和容量

图的阶和容量是用来衡量集合的大小，若存在子图 $A \in G$ ，则图 A 的阶和容量分别为：

$$|A| = \text{图} A \text{ 内顶点的个数} \quad (5-1)$$

$$vol(A) = \sum_{i \in A} d_i \quad (5-2)$$

(4) 图的割

设图 G 有两个子图 A 和 B ，且 $A \cup B = V$ ， $A \cap B = \emptyset$ ，连接 A 和 B 的边的集合就称为图的割集，那么割集中权值总和称为割^[6]：

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) \quad (5-3)$$

其中 $w(u, v)$ 为顶点 u 和 v 之间的边权值，对图最优二划分就是最小化函数 $cut(A, B)$ 。

5.1.2 相似矩阵和图的 Laplacian 矩阵

谱聚类将图划分准则优化问题转换成求解相似矩阵或 Laplacian 矩阵特征问

题，相似矩阵是用来描述数据点之间的相似关系，常用 S 表示。经典谱聚类中相似矩阵一般定义为高斯核函数：

$$S_{ij} = \exp(-\frac{d(x_i, x_j)}{2\sigma^2}) \quad (5-4)$$

其中 x_i 表示每个数据样本点， $d(x_i, x_j)$ 一般取 $\|x_i - x_j\|^2$ ， σ 为人工指定的参数。将相似矩阵 S 的每行元素相加，即得到该顶点的度，以所有度值为对角元素构成的对角矩阵即为度矩阵，常用 D 表示度矩阵。

谱聚类算法的主要工具是图 Laplacian 矩阵，其主要通过 Laplacian 矩阵将样本点映射到较低维空间，因为数据在低维数据空间具有更好的聚类特性。构造 Laplacian 矩阵方式主要有两种：

(1) 未规范化 Laplacian 矩阵：

$$L = D - W \quad (5-5)$$

性质 1: 未规范化 Laplacian 矩阵具有如下性质：

- a. 对于向量 $f \in R^n$ ，满足 $f'Lf = \frac{1}{2} \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2$ ；
- b. L 是对称切半正定的；
- c. L 的最小特征值为 0，对应的特征向量为所有元素都为 1 的向量；
- d. L 有 n 个非负的、实的特征值，即 $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$

定理 1: 令 G 为一个无向权值图，则未规范化 Laplacian 矩阵 L 的特征值是零的个数即为其图中独立的连通分量的个数，即若其有 k 个零特征值，则其有 k 个连通分量 A_1, A_2, \dots, A_k ，特征值为零的特征空间由连通分量的指示向量 $l_{A_1}, l_{A_2}, \dots, l_{A_n}$ 张成。

通过以上阐述，可以得到谱聚类算法的具体实现流程：

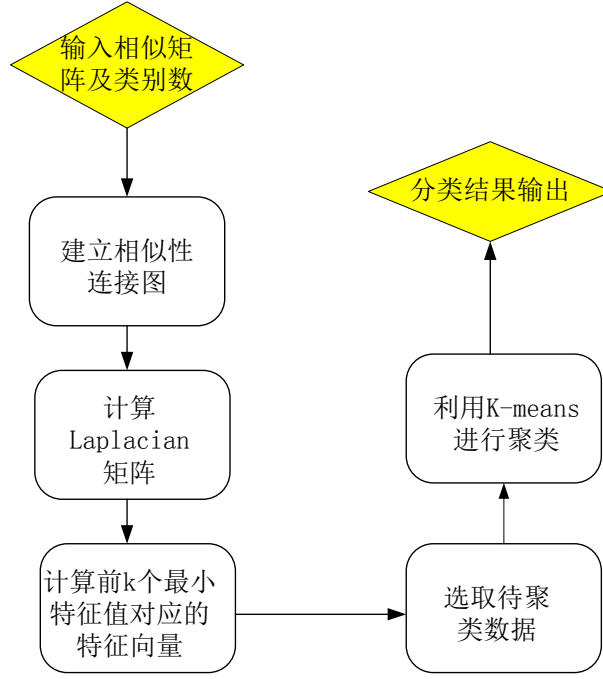


图 5-2 未规范化谱聚类算法实现流程图

(2) 规范化 Laplacian 矩阵有两种形式:

$$L_{sym} = D^{-1/2} L D^{1/2} \quad (5-6)$$

$$L_{rm} = D^{-1} L = I - D^{-1} W \quad (5-7)$$

其中 L_{sym} 指对称矩阵, L_{rm} 指与随机游走(Random Walk)相关的矩阵。关于规范化 Laplacian 矩阵的性质定理以及规范化谱聚类算法流程不再赘述。

5.2 谱聚类评价测度

评价测度是衡量聚类算法聚类效果的重要指标, 目前存在很多评价测度, 其中集合匹配评估测度, 成对匹配测度和基于熵的测度是基本的三类评价测度^[7]。假设 C 表示将要被评估的聚类结果, A 表示初始理想的聚类结果, n 表示样本数据集的个数, 聚类数目为 k , 下面将介绍这三类评价测度。

(1) 集合匹配评估测度

集合匹配评估测度最主要的特点是假设初始理想的聚类的结果的类别和聚类的结果的类别之间存在一对一映射关系, 主要依据了信息检索技术中的精度(precision)和召回率(recall)的概念, 其中比较常用测度之一是 F-measure:

$$F-measures = \sum_i \frac{|A_i|}{n} \max_j \{F(A_i, C_j)\} \quad (5-8)$$

$$F(A_i, C_j) = \frac{2 \times Recall(A_i, C_j) \times Precision(A_i, C_j)}{Recall(A_i, C_j) + Precision(A_i, C_j)} \quad (5-9)$$

$$Recall(A, C) = Precision(C, A) \quad (5-10)$$

$$Precision(C_i, A_j) = \frac{|C_i \cap A_j|}{|C_i|} \quad (5-11)$$

其中 $C_i \cap A_j$ 表示公共交集也就是 C 和 A 共同的样本集，F-measure 试图获取被研究的类别与初始理想类别相匹配的好坏程度，例如匹配任意两个集合直到匹配到两个最相似的集合，所以经常被看作是量化 C 和 A 之间的平均匹配水平。F-measure 的取值范围是 $[0 \sim 1]$ ，最优的划分是期望其值尽可能的大。

(2) 成对匹配评估测度

此类方法的特点是对成对的项进行数理统计，首先引入 SS ， SD ， DS 和 DD 四个统计量，它们的定义分别如下：

SS —两个点属于 C 中的同一类， A 中同一类；

SD —两个点属于 C 中的同一类， A 中不同类；

DS —两个点属于 C 中的不同类， A 中同一类；

DD —两个点属于 C 中的不同类， A 中不同类；

设 a ， b ， c ， d 分别表示 SS ， SD ， DS 和 DD 的数目，那么 a 和 d 用来计算划分的一致性， b 和 c 用来计算偏差，且 a 和 d 一般被称为好的选择，而 b 和 c 被称为坏的选择。 $a+b+c+d=M$ 就为数据集中所有对的最大数，即 $M = n \times (n-1) / 2$ ，下面列举几种比较常见的评价测度：

Jaccard 系数：

$$J = \frac{a}{a+b+c} \quad (5-12)$$

Folkes 和 Mallows：

$$FM = \sqrt{\frac{a}{a+b} \frac{a}{a+c}} \quad (5-13)$$

RandIndex(RI)系数：

$$RI = \frac{a+d}{M} = \frac{a+d}{n(n-1)/2} \quad (5-14)$$

其中 RandIndex (RI) 系数是一种经典且常用的评价测度之一,这种测度主要基于三种基本的考虑:

- 1) 聚类过程是严格意义上的严谨,也就是说每个数据点聚类中不存在二义性;
- 2) 所有数据集中的点以成对的形式存在,就是说两个数据点要么被聚在同一类,要么被聚在不同类;
- 3) 所有数据点在聚类中都是同等重要性的。

所以 RI 强调的成对的数据对象被看作为一类或者不同类的概率大小,通常用以反映集群同质性和完整性,且这两种性质是集合的本质特征。RI 取值范围为[0~1], RI 值越大,表示聚类效果越好。且当 RI=1 时,表明聚类结果与真实类标号完全吻合。

(3) 基于熵的评估测度

聚类熵的思想来自于信息理论,其反映的是 k 个聚类结果成员与真实类的成员之间的分布情况,是一种比较群集性质的方法。标准化互信息(Normal Mutual Information, NMI)是量化两个不同分布的交互统计信息的一种常见方法,即可以用来反映 C 和 A 的互信息:

$$NMI = \frac{\sum_{i=1}^k \sum_{j=1}^k \frac{X_{ij}}{n} \log \frac{n|X_{ij}|}{|C_i||A_j|}}{\sqrt{(\sum_{i=1}^k \frac{|C_i|}{n} \log \frac{|C_i|}{n})(\sum_{j=1}^k \frac{|A_j|}{n} \log \frac{|A_j|}{n})}} \quad (5-15)$$

NMI 取值范围为[0~1], 值越大表明聚类效果越好。

数据集聚类结果的好坏取决于不同的评价准则,因此对于谱聚类算法,评价测度的选取是非常重要的,从不同方法的适用角度考虑,本文选取 RI, NMI, F-measure 三种方法作为评价测度。

6 问题一：独立子空间高维数据聚类问题

在面向数据挖掘的聚类研究中，所要处理的数据经常会有几十甚至几百个属性。将这些数据对象表示成高维属性空间中的点或者向量，就可以把现实应用中的对象集用高维空间中点的集合来表示。多数传统聚类算法都是针对低维空间进行处理的。由于高维数据的稀疏性、空空间现象以及维度效应的影响，高维空间中不可能使数据在全维空间密集，而传统的聚类算法都是针对全维空间进行的，并且多数算法都运用了全维空间中距离的概念。上述问题给高维数据的聚类带来诸如运算量大、算法复杂度高等问题，需要进一步挖掘数据本身的性质，基于上述图的概念和性质，建立以下模型。

6.1 模型一：基于谱聚类的独立子空间分割模型

6.1.1 模型建立

聚类的目标是对给定点集进行分类，使得各个类间的相似度达到最小，而每个类内部点集的相似度达到最大。对谱聚类而言，给定点集 $\{x_1, x_2, \dots, x_n\}$ ，通过图模型建立起一个相似图，各个点之间通过相似函数 s_{ij} 来定义两点间的相似性。谱聚类的目标则是使得各个不同类之间点与点的相似函数之和最小，而每个类内部的点与点的相似函数之和最大。给定一个图模型 G ，可以得到图 G 的加权矩阵 W ，因而可以建立基于谱聚类的子空间分割模型：

$$\begin{cases} \min Cut(B_1, \dots, B_k) = \min \frac{1}{2} \sum_{i=1}^k W(B_i, \overline{B_i}) \\ s.t. s_{ij} = s(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / (2\sigma^2)), i=1, \dots, n; j=1, \dots, n \end{cases} \quad (6-1)$$

其中， k 表示分割的个数，表示图的一种划分，同时也表示这种划分下的 k 个点集。因子 $1/2$ 表示图中点与点链接的每条边只计数一次。当 $k=2$ 时，此最优切割问题可以得到很好的解释。但是实际问题中该切割问题的最优解往往与实际情况有些差距，因为有时候最优切割问题的解会包含一些元素个数较少的点集，这种切割结果与一般实际情况不相符。因而需要将点集的大小或者点集中含有的点的个数考虑在内，故建立基于**比值最优分割**（*RatioCut*）谱聚类的子空间分割模型：

$$\begin{cases} \min RatioCut(B_1, \dots, B_k) = \min \frac{1}{2} \sum_{i=1}^k \frac{W(B_i, \overline{B_i})}{|B_i|} = \min \sum_{i=1}^k \frac{cut(B_i, \overline{B_i})}{|B_i|} \\ s.t. s_{ij} = s(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / (2\sigma^2)), i=1, \dots, n; j=1, \dots, n \end{cases} \quad (6-2)$$

该模型充分考虑了点集的大小，若点集较小，即 $|B_i|$ 较小，则 *RatioCut* 的值会较大，此时对应的点集并非为最优切割问题的解。于是通过引入 *RatioCut* 可以避免以上描述问题的出现。

6.1.2 模型求解

(1) $k = 2$ 时的比值最优分割问题的近似解

因为 $k = 2$ 时 *RatioCut* 问题的近似解较好理解，所以先对 $k = 2$ 的 *RatioCut* 问题进行求解。首先建立如下的目标函数：

$$\min_{B \subset V} \text{RatioCut}(B, \bar{B}) \quad (6-3)$$

为了易于求解 (6-3) 式，对 (6-3) 式进行改写。首先给出指示向量 $t \in R^n$ 的定义。给定点集 $B \subset V$ ，定义集合 B 的指示向量 $t = (t_1, t_2, \dots, t_n) \in R^n$ ，其中 $t_i, i = 1, \dots, n$ 满足下列条件：

$$t_i = \begin{cases} \sqrt{|B|/B} & \text{if } v_i \in B \\ -\sqrt{|B|/B} & \text{if } v_i \in \bar{B} \end{cases} \quad (6-4)$$

此时向量 $t = (t_1, t_2, \dots, t_n) \in R^n$ 可以较好地将集合 V 内的点集区分开，若第 i 个点属于集合 B ，则相应的向量 t 的第 i 个分量 $t_i > 0$ 为正值；若第 i 个点不属于集合 B ，则相应的向量 t 的第 i 个分量 $t_i < 0$ 。

根据式 (6-3) 中的定义，将式 (6-4) 改写成矩阵和向量相乘的形式，同时将 *RatioCut* 与非标准化的 Laplacian 矩阵建立联系，具体推导如下：

$$\begin{aligned}
t'Lt &= \frac{1}{2} \sum_{i,j=1}^n w_{ij} (t_i - t_j)^2 \\
&= \frac{1}{2} \sum_{i \in B, j \in \bar{B}} w_{ij} \left(\sqrt{\frac{|\bar{B}|}{|B|}} + \sqrt{\frac{|B|}{|\bar{B}|}} \right)^2 + \frac{1}{2} \sum_{i \in B, j \in \bar{B}} w_{ij} \left(-\sqrt{\frac{|\bar{B}|}{|B|}} - \sqrt{\frac{|B|}{|\bar{B}|}} \right)^2 \\
&= cut(B, \bar{B}) \left(\frac{|\bar{B}|}{|B|} + \frac{|B|}{|\bar{B}|} + 2 \right) \\
&= cut(B, \bar{B}) \left(\frac{|\bar{B}| + |B|}{|\bar{B}|} + \frac{|\bar{B}| + |B|}{|B|} \right) \\
&= |V| \cdot RatioCut(B, \bar{B})
\end{aligned} \tag{6-5}$$

于是得到：

$$t'Lt = |V| \cdot RatioCut(B, \bar{B}) \tag{6-6}$$

其中 $|V| = |B| + |\bar{B}|$ ， $|B|$ 表示集合 B 中点集的个数。则由 (6-6) 式可以看出非标准化的 Laplacian 矩阵与 $RatioCut$ 之间的关系，即 (6-6) 式的最优化问题可以转换成下述问题：

$$\min_{B \subset V} t'Lt \tag{6-7}$$

由式 (6-4) 可对上述优化问题进行约束，最终得到带约束的优化问题模型：

$$\begin{aligned}
&\min_{B \subset V} t'Lt, \\
&s.t. \quad t \perp \tilde{1}, \\
&\quad \|t\| = \sqrt{n}
\end{aligned} \tag{6-8}$$

其中， $\tilde{1} = [1, \dots, 1]$ 为全 1 向量。(6-8) 式得到的是一个离散点优化问题，即其中的指示向量 t 的分量是两个特定的值，求解这样一个离散优化问题是一个 NP-hard 问题，于是需要求解 (6-8) 式的近似解。不限制指示向量 t 的特殊取值，让其可在全实数空间内取值，即：

$$\begin{aligned}
&\min_{t \in R^n} t'Lt, \\
&s.t. \quad t \perp \tilde{1}, \|t\| = \sqrt{n}
\end{aligned} \tag{6-9}$$

采用 Rayleigh-Ritz 定理对式 (6-9) 进行求解：

定理 2: (Rayleigh-Ritz 定理) 令 $A \in C^{m \times n}$ 为 Hermitian 矩阵, 对下述实函数:

$$\begin{cases} M: C^n / \{0\} \rightarrow R \\ M(t) = \frac{t'At}{t't}, \|t\| \neq 0 \end{cases}$$

可以得到:

$$\begin{cases} \lambda_{\max} = \max M(t) = \max_{\|t\| \neq 0} \frac{t'At}{t't} \\ \lambda_{\min} = \min M(t) = \min_{\|t\| \neq 0} \frac{t'At}{t't} \end{cases}$$

由 Rayleigh-Ritz 定理可以看出式 (6-9) 的解是非标准化 Laplacian 矩阵 L 的次最小特征值对应的特征向量。因为非标准化 Laplacian 矩阵 L 的最小特征值为 0, 其对应的特征向量为全 1 向量, 显然不是最终希望得到的结果, 所以选择次最小特征值对应的特征向量作为对式 (6-8) 的近似解较为合适。

在得到对于问题式 (6-9) 的近似解之后, 因为得到的近似解不同于 (6-4) 式的表示形式, 即指示向量 t 中的元素并非只有两个离散数字连续出现, 于是需要对得到的指示向量 t 进行转换, 最终易于从指示向量中得到所需要的类的结果。

由式 (6-9) 的解可以看出只需要对非标准化的 Laplacian 矩阵求解其特征向量, 再对求解后的特征向量使用 K-means 聚类就可以完成对数据点的谱聚类。

(2) $k > 2$ 时的比值最优分割问题的近似解

类似于 $k = 2$ 时的情况, 给定一个点集的划分得到划分集合 B_1, \dots, B_k , 定义对于 k 个点集 B_1, \dots, B_k 的指示向量 $h_j = (h_{1,j}, \dots, h_{n,j})$ 如下:

$$h_{i,j} = \begin{cases} 1/\sqrt{|B_j|}, & v_i \in B_j \\ 0, & v_i \notin B_j \end{cases} \quad (i=1, \dots, n; j=1, \dots, k) \quad (6-10)$$

构造矩阵 $H_{n \times k} = (h_1, \dots, h_k)$, 可知矩阵 $H_{n \times k}$ 满足 $H'H = I$, 且有:

$$h_i' L h_i = \frac{cut(B_i, \overline{B_i})}{|B_i|} \quad (6-11)$$

由矩阵 $H_{n \times k}$ 的构造, 得到:

$$h_i' L h_i = (H' L H)_{ii} \quad (6-12)$$

结合式（6-11）、（6-12）可得：

$$\begin{aligned} RatioCut(B_1, \dots, B_k) &= \sum_{i=1}^k h_i' L h_i \\ &= \sum_{i=1}^k (H' L H)_{ii} = Tr(H' L H) \end{aligned} \quad (6-13)$$

此时优化问题可转换成如下形式：

$$\min_{B_1, \dots, B_k \subset V} Tr(H' L H) \quad (6-14)$$

其中矩阵 H 满足 $HH^T = I$ 。

可以得到式（6-14）的解即是 Laplacian 矩阵 L 对应的前 k 个最小特征值的特征向量。类似于 $k=2$ 的情形，将得到的 k 个特征向量拼成一个矩阵 $X_{n \times m}$ 即得到了原始数据点转换成的 k 维新的数据点。新的数据点的优点是具有较好的区分度，可以通过直接使用 K-means 聚类完成对原始数据点的分类。算法流程图可以表示为：

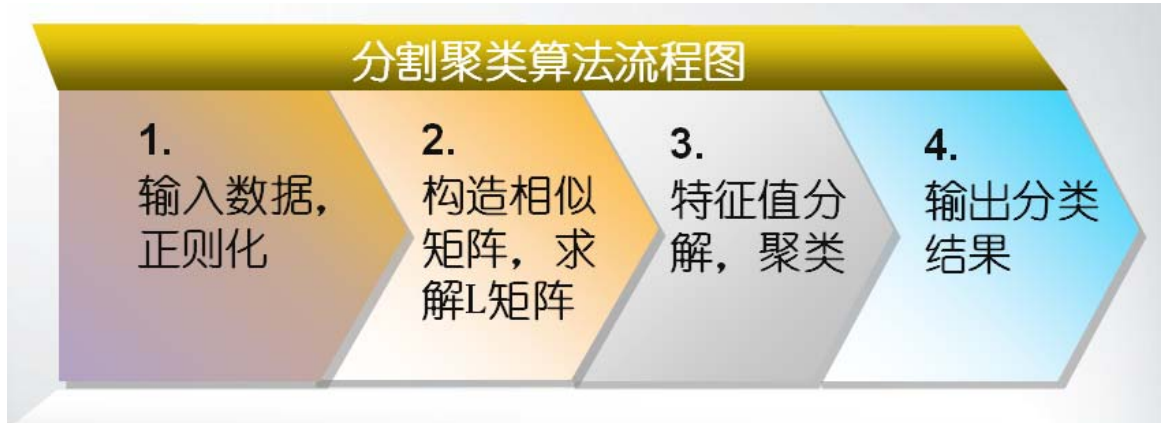


图 6-1 最优分割聚类算法流程图

6.2 模型二：基于共享近邻的自适应谱聚类模型

1.mat 给出了一组高维数据，要求将其分为两类。当数据维数较高的时候，其受到随机因素的影响较大，同时数据处在高维空间中，分布较为离散。我们采用先降维再聚类的方式，首先使用随机投影技术将数据投影到较低的维数，再对模型一中的相似矩阵进行改进，建立基于共享近邻的自适应谱聚类模型。首先对随机投影和共享近邻点的概念进行介绍。

6.2.1 模型准备

(1) 随机投影

定义 1: 给定一原始数据矩阵 $X_{n \times m}$ ，则对该原始数据矩阵左乘一个随机矩阵即 $A_{m \times p}$ ($m > p$)，即 $M_{n \times p} = X_{n \times m} A_{m \times p}$ ，则得到的新的数据矩阵 $M_{n \times p}$ 即成为由原始数据矩阵经过随机投影之后得到的新的数据矩阵。

定理 3 (Johnson-Lindenstrauss): 对 $\forall 0 < \varepsilon < 1$ 和任意整数 n ，令 k 满足： $k \geq 4(\varepsilon^2 / 2 - \varepsilon^3 / 3)^{-1} \ln n, k > 0$ ，则对任意的 n 个点的集合 V ，存在一个投影映射： $f: R^d \rightarrow R^k$ ，使得对于所有的 $u, v \in V$ 有下式成立：

$$(1 - \varepsilon) \|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \varepsilon) \|u - v\|^2$$

随机投影之所以能够起到较好的作用是因为 Johnson-Lindenstrauss 定理在理论上给了该投影方法性质上的保障，该定理的证明及其他形式的推广说明了以较大的概率存在随机投影变换使得保距性质成立。即投影后的点集两点之间的距离跟原始点集中两点之间的距离可以通过 $(1 + \varepsilon)$ 和 $(1 - \varepsilon)$ 来进行控制，通过限定 ε 的值可以确定降维需要的维数范围，则得到的随机投影矩阵即是满足定理 3 条件的投影矩阵，再通过大量的计算机模拟，即可得到降维效果较好的随机投影矩阵。定理 3 即使得先使用随机投影再使用聚类方法对高维点集进行聚类成为了可能。

(2) 共享近邻点

相似矩阵的构建是谱聚类算法的关键，实验发现在相似矩阵的构建过程中如果能有效利用数据局部近邻信息，更有利于提高谱聚类算法性能。给定 n 个 d 维数据集 $X = \{x_1, x_2, \dots, x_n\} \in R^d$ ，那么获取任意两点 x_i 和 x_j 之间的共享近邻顶点的步骤如下：

- a. 确定每个点 $x_i (i = 1, \dots, n)$ 的前 k 个近邻，其中近邻的确定依据欧氏距离判定，根据顺序并标上标签记号 $1 \sim k$ ，并且每一次计算把点本身看作自己的第 0 个近邻；
- b. 建立一个长度为 n 的整数标签表格，并且每一行第一个元素为数据的标签，每行最初设置为相应行第一个元素的所有近邻（如图 6-2 所示）；
- c. 所有数据对的 kd 个共享近邻数都依据以下方法得到：根据从小到大的顺序匹配两个标签行前 kd 个近邻点的标签，并记录其中相同标签，即是所要求的

共享近邻的顶点。

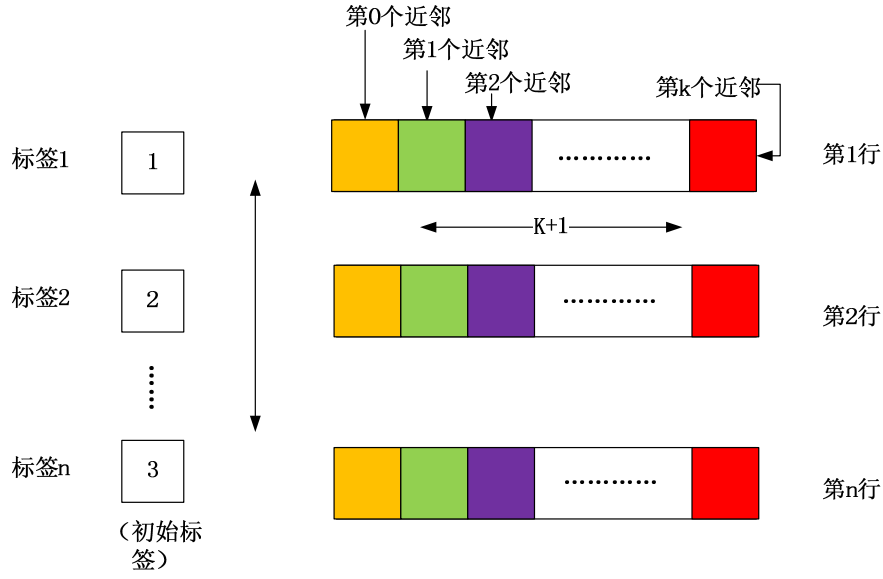


图 6-2 近邻和标签示意图

6.2.2 模型建立与求解

结合模型一和共享近邻的思想，建立基于共享近邻的自适应谱聚类模型：

$$\left\{ \begin{array}{l} \text{RatioCut}(B_1, \dots, B_k) := \frac{1}{2} \sum_{i=1}^k \frac{W(B_i, \overline{B_i})}{|B_i|} = \sum_{i=1}^k \frac{\text{cut}(B_i, \overline{B_i})}{|B_i|} \\ \text{s.t. } s_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma_i \sigma_j (SNN(x_i, x_j) + 1)}\right) & i \neq j \\ 0 & i = j \\ SNN(x_i, x_j) = |N(x_i) \cap N(x_j)|, i = 1, \dots, n; j = 1, \dots, n \end{cases} \end{array} \right. \quad (6-15)$$

其中， $N(x_i)$ 、 $N(x_j)$ 分别为与点 x_i 、 x_j 最近的前 kd 个点构成的集合。

与大部分谱聚类模型中构建的图都是无向图不同，基于共享近邻的相似测度中首先构造的是个有向图，基于共享近邻的相似测度不仅表征了数据点对之间的结构信息又利用了局部密度信息从而构建了顶点之间的相似度。

模型二与模型一的求解过程类似，不同的地方在于两处：一是对数据进行了降维处理，二是所构建的相似矩阵不同。附录 1 给出了对 1.mat 中的数据进行分类的结果，其中，编号 1~40 和编号 141~200 的数据被分为第 2 类，编号 41~140 的数据被分为第 1 类。

7 问题二：低维空间中的子空间与多流形聚类问题

问题二包含了子空间和多流形聚类问题，数据中既包括线性结构又包括非线性结构、既有良分离（不交叠）的结构又有相互交叠的结构。模型一中的谱聚类算法可用来较好地解决非线性聚类问题，但是当流形之间存在交叠时，该模型不能很好地将来自于不同流形上的数据区分开来，我们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矩阵，进而发现正确的流形聚类。

7.1 模型三：基于谱多流形算法的聚类模型

7.1.1 模型建立

建立谱多流形聚类模型如下：

$$\left\{ \begin{array}{l} \min \frac{1}{2} \sum_{i=1}^k W(B_i, \overline{B_i}) \\ \text{s.t. } \Theta_i = \text{span}(\mathbf{V}_i) \\ p_{ij} = p(\Theta_i, \Theta_j) = \left(\prod_{l=1}^d \cos(\theta_l) \right)^o \\ s_{ij} = p_{ij} q_{ij} = \begin{cases} \left(\prod_{l=1}^d \cos(\theta_l) \right)^o & \text{if } \mathbf{x}_i \in \text{Knn}(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \text{Knn}(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} \end{array} \right. \quad (7-1)$$

其中， Θ_i 为数据点 $\mathbf{x}_i (i=1, \dots, N)$ 处的“局部切空间”， p_{ij} 为两个数据点 \mathbf{x}_i 、 \mathbf{x}_j 局部切空间之间的“结构相似性”， s_{ij} 即为所构造的相似矩阵的相似性权值。

7.1.2 模型求解

(1) 局部切空间

我们训练 M 个混合概率主成分分析器来估计局部切空间，其中每个分析器由模型参数 $\theta_m = \{\boldsymbol{\mu}_m, \mathbf{V}_m, \sigma_m^2\} (m=1, \dots, M)$ 刻画，其中 $\boldsymbol{\mu}_m \in \mathbb{R}^D$ ， $\mathbf{V}_m \in \mathbb{R}^{D \times d}$ ， σ_m^2 是一个标量。 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。模型参数 $\boldsymbol{\mu}_m$ ， \mathbf{V}_m ， σ_m^2 可以通过利用 EM 算法最大化观测数据 \mathbf{x}_i 的对数似然来得到。

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

$$R_{im} = \frac{\pi_m p(\mathbf{x}_i | m)}{\sum_{m=1}^M \pi_m p(\mathbf{x}_i | m)} \quad (7-2)$$

$$\pi_m^{new} = \frac{1}{N} \sum_{i=1}^N R_{im} \quad (7-3)$$

$$\mu_m^{new} = \frac{\sum_{i=1}^N R_{im} \mathbf{x}_i}{\sum_{i=1}^N R_{im}} \quad (7-4)$$

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

$$\mathbf{V}_m^{new} = \mathbf{S}_m \mathbf{V}_m (\sigma_m^2 \mathbf{I} + \mathbf{T}_m^{-1} \mathbf{V}_m^T \mathbf{S}_m \mathbf{V}_m)^{-1} \quad (7-5)$$

$$(\sigma_m^2)^{new} = \frac{1}{d} \text{tr}[\mathbf{S}_m - \mathbf{S}_m \mathbf{V}_m \mathbf{T}_m^{-1} (\mathbf{V}_m^{new})^T] \quad (7-6)$$

最后，样本点 \mathbf{x}_i 根据下述关系分组到第 j 个局部分析器:

$$p(\mathbf{x}_i | j) = \max_m p(\mathbf{x}_i | m) \quad (7-7)$$

同时其局部切空间由下式给出:

$$\Theta_i = \text{span}(\mathbf{V}_j) \quad (7-8)$$

(2) 相似性矩阵

设数据点 $\mathbf{x}_i (i=1, \dots, N)$ 处的局部切空间为 Θ_i ，则两个数据点 \mathbf{x}_i 和 \mathbf{x}_j 的局部切空间之间的结构相似性可以定义为:

$$p_{ij} = p(\Theta_i, \Theta_j) = \left(\prod_{l=1}^d \cos(\theta_l) \right)^o \quad (7-9)$$

其中， $o \in \mathbb{N}^+$ 是一个可调参数， $0 \leq \theta_1 \leq \dots \leq \theta_d \leq \pi/2$ 是两个切空间 Θ_i ， Θ_j 之间的主角度。

数据点 \mathbf{x}_i 和 \mathbf{x}_j 之间的局部相似性简单地定义为:

$$q_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in Knn(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in Knn(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \quad (7-10)$$

其中 $Knn(\mathbf{x})$ 代表 \mathbf{x} 的 K 个近邻数据点，最终得到相似性权值:

$$s_{ij} = p_{ij}q_{ij} = \begin{cases} (\prod_{l=1}^d \cos(\theta_l))^o & \text{if } \mathbf{x}_i \in Knn(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in Knn(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} \quad (7-11)$$

(3) 算法实现流程



图 7-1 谱多流形聚类流程图

7.1.3 问题二分类效果及分析

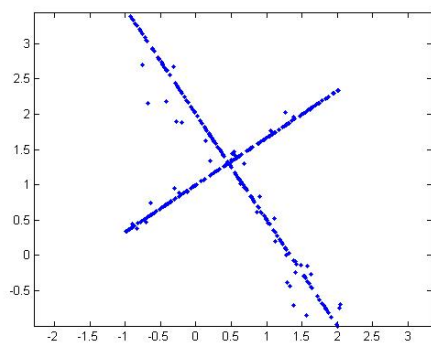
此部分首先以第 (2a) 小题为例，分别采用了模型一和模型三两种方法对其进行求解，并测出了 5.2 节中提到的三种评价测度，如表 7-1 所示，从中可以看出谱聚类算法对于该类问题的求解性能较差，对第 2b~2d 小题的求解采用模型三的算法。

表 7-1 两种算法对 2a 聚类时的评价测度

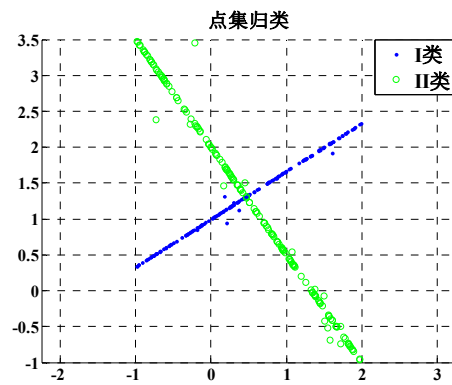
聚类算法	F-measure 值	RI 值	NMI 值
比值最优分割谱聚类算法	0.5782	0.5352	0.1975
谱多流形聚类方法	0.9588	0.9424	0.7919

7.1.3.1 分类效果

(2a) 交叉十字形数据分类：

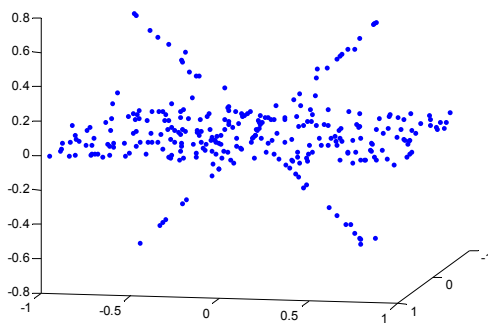


(a) 原图

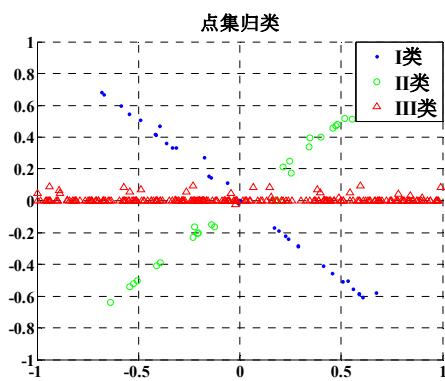


(b) 分类

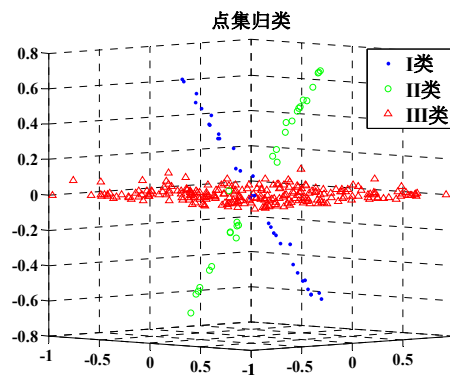
(2b) 平面与线相交叉的数据分类:



(c) 原图

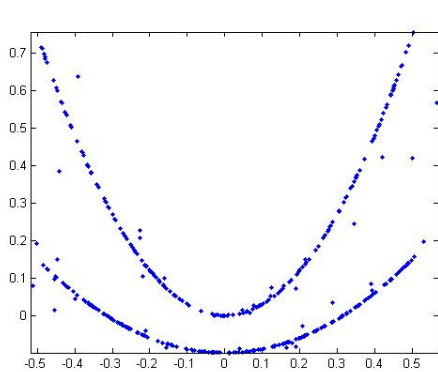


(d) 二维表示下的分类

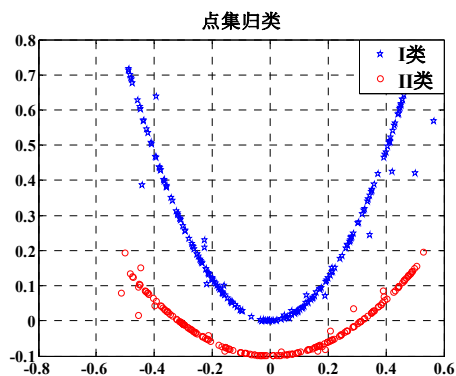


(e) 三维表示下的分类

(2c) 不相交的二次曲线数据分类:

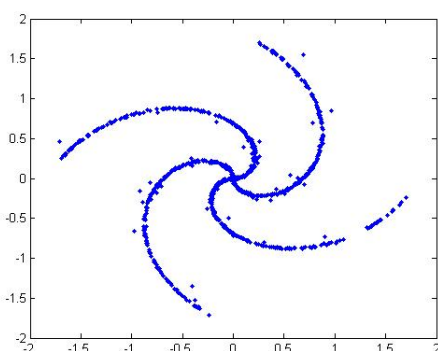


(f) 原图

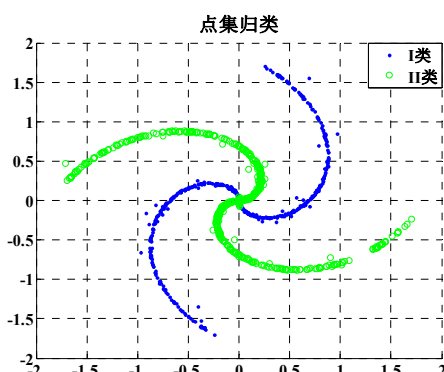


(g) 分类

(2d) 相交螺旋线数据分类:



(h) 原图



(I) 分类

图 7-2 问题二分类效果图

结果分析与讨论:

- 1) 模型一中的谱聚类模型不适合对具有相交结构的数据进行分类, 这说明当流形之间存在交叠时, 谱聚类算法的性能会明显变差; 由于谱聚类算法更多的是利用点与点之间的距离信息, 当有交叉区域时, 往往失去了有效信息, 因此无法分割;
- 2) 谱多流形聚类模型在四种情况下都能够得到较好的分离效果, 这说明该模型能够有效地将流形交叉处的点拆解开来得到不同流形结构的成分。谱多流形模型能够充分局部结构的几何信息, 从而克服了距离远近的干扰, 能够实现交叉区域的有效分割

7.1.3.2 参数影响

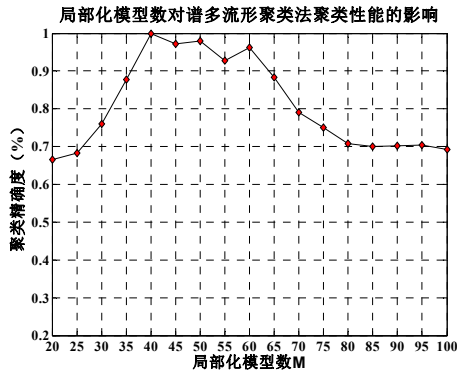
谱多流形聚类算法中有三个可调节参数, 即局部化模型数 M , 近邻点数 K 和

调节参数 α 。我们考察这些参数的设置对该算法聚类性能的影响。在(2d)中相交螺旋线数据上的实验结果如图 7-3 所示, 为算法在 10 次独立实验下的平均聚类结果, 图中给出了该算法聚类精确度随三种参数选取变化的结果。可以得到这样的结论:

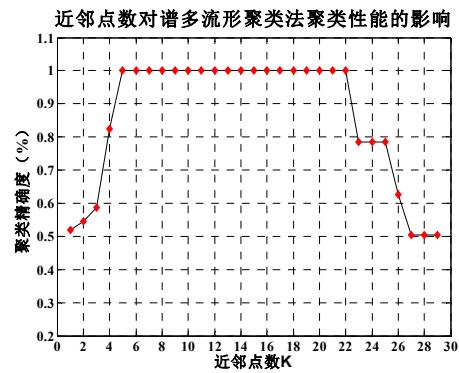
(1) 当局部化模型数 $M \in [\lceil N/(10d) \rceil, \lceil N/(2d) \rceil]$ 时, 聚类的效果较好, 这是因为在此取值区间内平均重构误差较小, 这意味着对每个数据点局部切空间的估计也越来越可靠, 进而使得谱多流形聚类算法具有更好的性能, 其中, d 为子空间的维度;

(2) 当近邻点数 K 既不太大也不太小时, 算法的性能在一段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。因此通过仿真实验, 我们认为 K 应该在 $[\lceil \log(N) \rceil, 3\lceil \log(N) \rceil]$ 内取值;

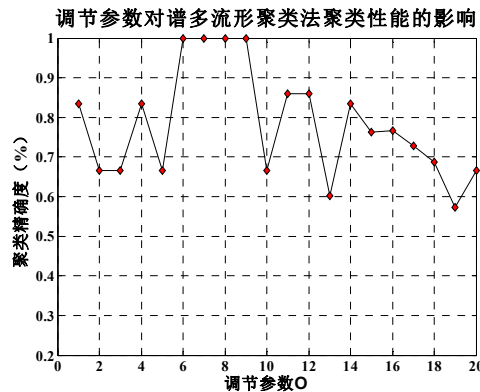
(3) 当调节参数 $\alpha \in [6, 10]$, 算法的性能很好。其原因在于, α 在此区间取值时, 来自不同流形的数据之间的可分性越好, 因为对 $x < 1$ 而言, 当 α 变大时 x^α 趋向于 0。



(a)



(b)



(c)

图 7-3 参数设置对谱多流形聚类算法聚类性能的影响

7.1.3.3 计算复杂度分析

由于聚类方法的计算复杂度直接影响到它的可适用性, 因而对谱多流形算法的计算复杂度做简要分析。

谱多流形算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、计算相似性矩阵和利用谱方法进行聚类。 N 个局部切空间 $\Theta_i, i=1, \dots, N$ 是通过 M 个混合的 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 $O(NDM(t_1+t_2))$, 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局部切空间之间结构相似性的复杂度为 $O(N^2Dd^2)$, 而搜索每个数据点的 K 个近邻点的复杂度为 $O((D+K)N^2)$ 。第三部分对 S 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 $O((N+k)N^2)$, 而 K-means 在 k 维投影数据上的复杂度为 $O(Nk^2t_3)$ (t_3 为该过程中 K-means 收敛所需的迭代步数)。因此谱多流形聚类算法方法总的计算复杂度为:

$$O(N^3 + N^2(Dd^2 + K + k) + N(DM(t_1 + dt_2) + k^2t_3)) \quad (7-12)$$

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 $d < D$, $K \ll N$, $k \ll N$, $M \ll N$, 因此谱多流形算法的复杂度主要由数据点数 N 和数据维数 D 决定。

8 问题三：子空间聚类在实际中的应用

对于上述四个低维空间中的目标分类问题，利用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的谱多流形聚类方法或者通过迭代更新属类概率矩阵的节点加权多维缩放方法，在一定情况都能够有效地实现目标的正确分类。在实际生活中，需要解决的往往是低维空间的扩展问题，一般的解决思路是在上述模型的基础上，通过扩展模型的应用维数，从而解决实际应用中高维数据的相关性分析、聚类分析等基本问题。

8.1 宽交叉十字形数据点群的分类问题

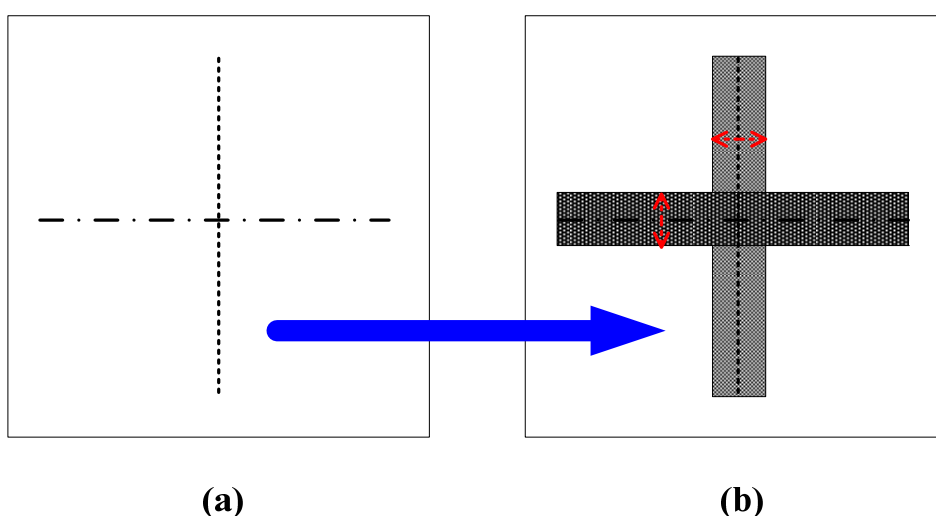
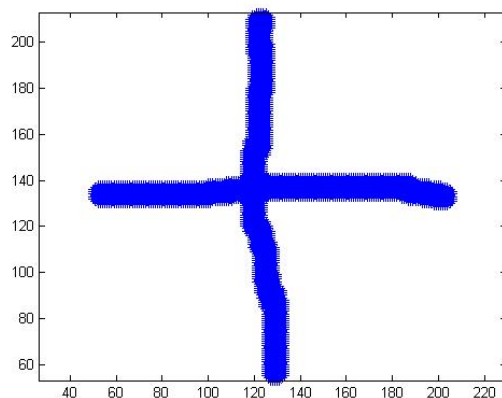
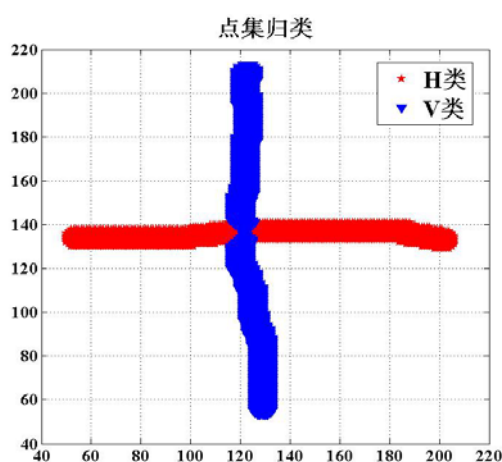


图 8-1 过程演化示意图

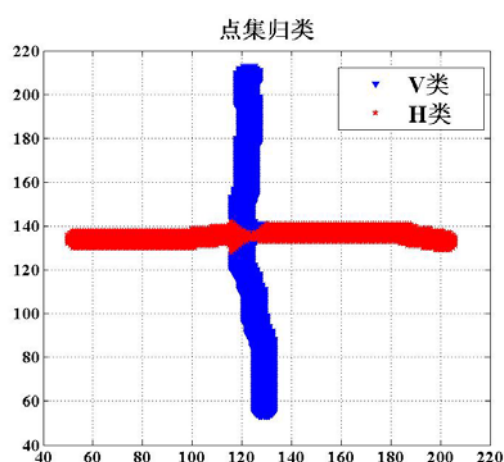
如图 8-1 所示，宽交叉十字形数据点群的分类问题本质上与问题二中互相垂直的两条直线分类问题相同，即区分有交互区域的两类点集。事实上，在缺乏先验知识的情况下，相交区域的点集归属是模棱两可的，不同的分类结果都具有足够充分局部几何结构信息来支撑其分类的正确性；当由相交的直线扩展为相交的矩形区域时，问题二（a）由分离两类一维子空间问题转化为分离两类二维子空间问题，在各自空间维数增加的同时，相交区域的扩大造成了局部几何结构信息的不确定性。为了表述方便，将十字区域的水平区域定义为“H”类，将十字区域的垂直区域定义为“V”类。下面将继续采用问题二（a）中的基于谱多流形聚类模型，在解决实际问题的同时，能够验证该模型解决交叉分类问题的鲁棒性。



(a) 原图



(b) 分类 1



(c) 分类 2

图 8-2 分类结果对比图

对十字交叉点集采用基于谱多流形聚类模型进行求解，图 8-2（a）为原始数据，图 8-2（b）和图 8-2（c）为同一算法的运行两次的不同结果。

结果分析与讨论：

- 1) 两类一维子空间问题转化为分离两类二维子空间问题，在各自空间维数增加的同时，基于谱多流形模型具有**强健的鲁棒性**，能够将两类点集有效地分离开，对于交叉区域的“干扰”具有不敏感性；
- 2) 该模型能够将非交叉点集部分有效地分离，但是对于中间公共交叉区域，算法并不能分离开来，每次计算都会产生不同的结果，事实上在误差范围内这种现象是合理存在的。

8.2 多帧图像运动特征点轨迹分类问题

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(由于同一物体具有完整性,因此可以认为每一个物体都是由相对集中的点群构成的),通过分割出来不同特征点轨迹(即分离点群的轨迹),能够实现对动态场景的理解和重构。如图 8-3 所示,图中给出了某一物体的运动轨迹示意图,物体用若干特征点表征,不同的位置代表了物体的不同帧的图像情景。

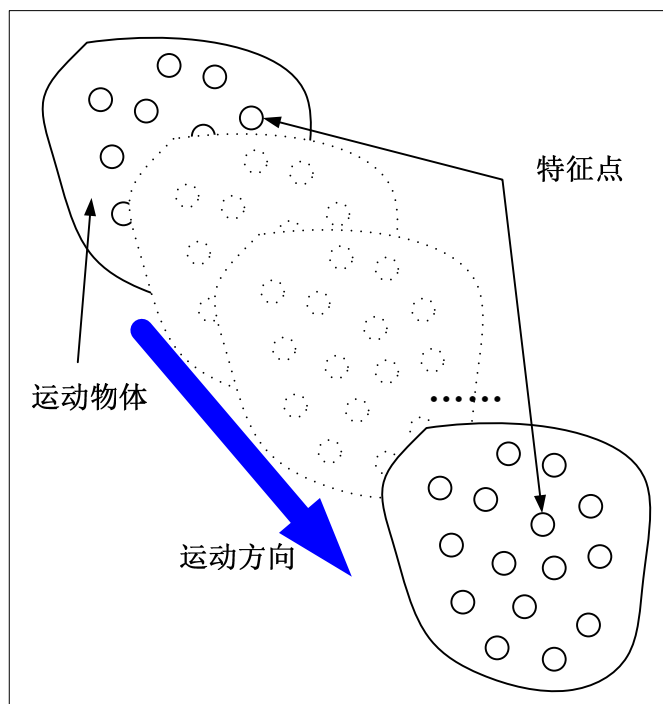


图 8-3 特征点运动轨迹示意图

在一些良好的环境下,在运动初始帧中,不同物体之间的差异性比较大,可以采用聚类算法将物体很好的分离开,通过监督流形学习算法,将可能得到样本的类别标记信息,在搜索近邻点过程中利用数据样本的类别标记信息,使得每一个数据点的近邻点都来自同一类别,这样数据点只与本类数据点相连接,不同类别的数据点不连接,形成若干个互不连接的子图。然而,在一些恶劣情况下,不同物体之间的差异不明显,仅利用初始帧的信息无法分离不同的物体,因而无法继续采用监督流形学习算法,如图 8-4 所示,图中的交叉区域,特征点类别无法区分,因此从逐帧计算的角度分析,这种情况显然是无法将分类进行下去的。

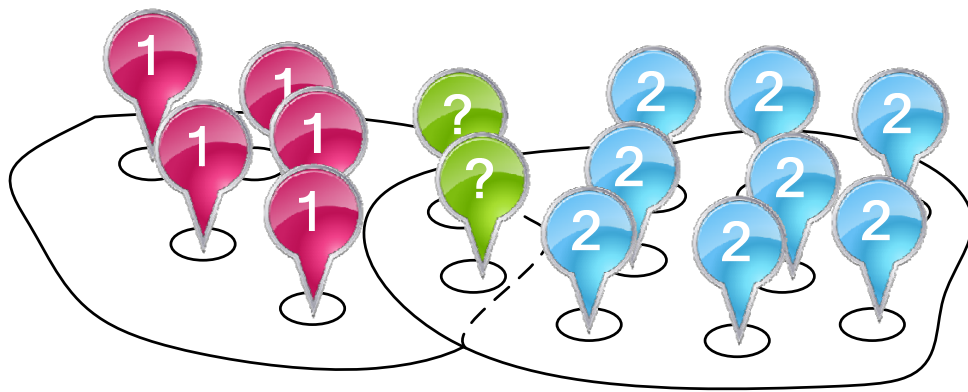


图 8-4 恶劣情形下特征点分类示意图

除了逐帧挖掘特征点信息以外，一种合理的方式是将每个单独的点作为一个研究对象，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，即一个物体的特征点轨迹在同一个线性流形上，因此可以按照每个特征点运动点轨迹进行分类，如图 8-5 所示：

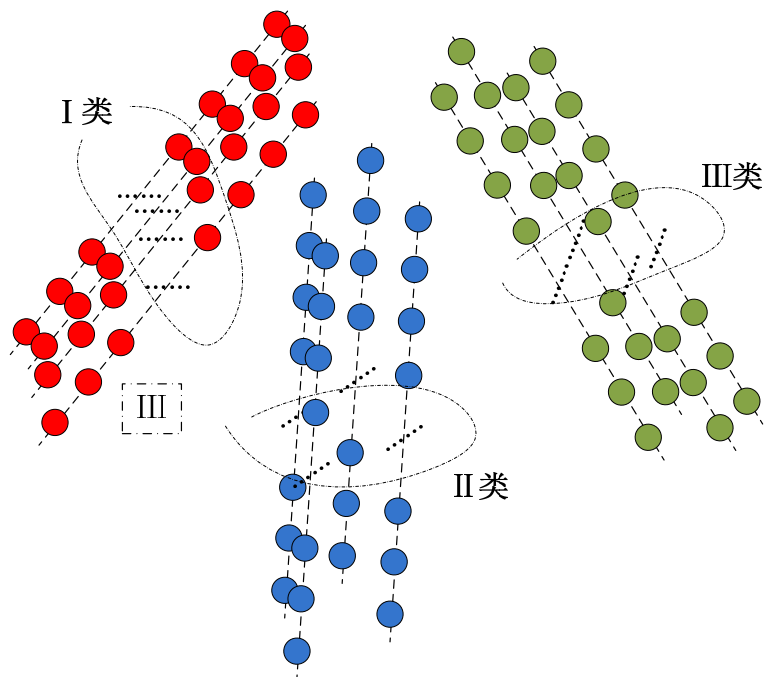


图 8-5 基于特征点运动流行聚类示意图

8.2.1 模型四：基于最小矢量差的谱多流形聚类模型

与一般的非线性流行聚类方式类似，首先需要构建邻接图 G 并计算 G 上两点之间的最短路径，从第二帧开始，每个点的信息要素为该点的位移矢量，通过计算当前时刻的坐标与前一时刻的坐标差得到，因此邻接图 G 上两点之间的最短路径为两点的矢量差，可以发现，位于同一物体上的不同特征点，由于其具有相同

的位移矢量，因此两点的矢量差为 0，根据矢量差进行聚类，能够充分利用特征点的时间序列信息，将具有相同运动特征的特征点进行聚类，并能有效地预测或者描述物体的运动规律。模型如下：

$$\min \phi(V) = \sum_{i=1}^N \sum_{j=1}^N w_i w_j ((d_{ij}^G) - \|\mathbf{v}_i - \mathbf{v}_j\|)^2 \quad (8-1)$$

其中 d_{ij}^G 是 \mathbf{v}_i 与 \mathbf{v}_j 之间的最短距离， \mathbf{v}_i 、 \mathbf{v}_j 为第 i 、 j 特征点的位移矢量。其元素 w_{ci} 表示数据 i 属于流形类别 c 的概率。

8.2.2 模型求解与讨论分析：

可以发现，上述模型以谱多流形聚类算法为基础，将最短路径距离这一优化指标替代为最小矢量差，能够充分的发掘同一物体特征点运动的相似性，由于上面问题里详细解释叙述了问题求解的过程，这里不再赘述，具体过程见下面对应的流程图 8-6。

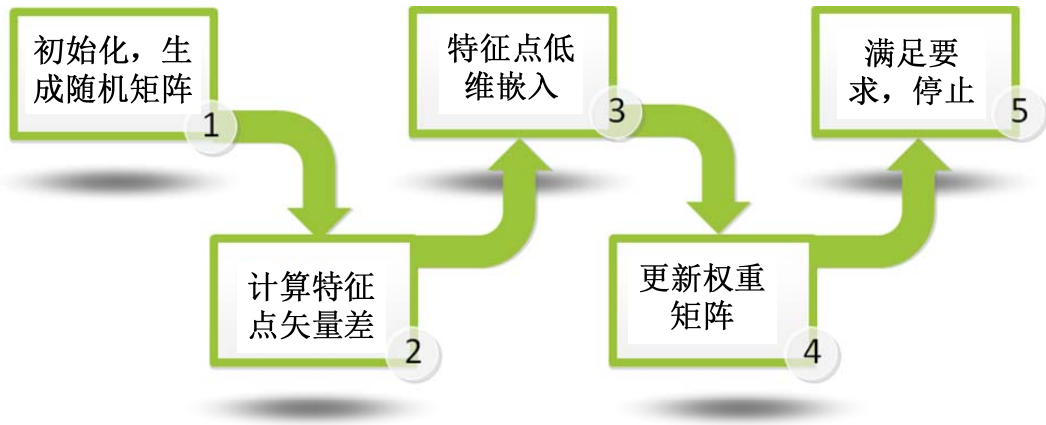


图 8-6 模型求解流程图

为了说明该模型解决运动点轨迹分割的稳定性，同时验证该算法的有效性不依赖于起始帧特征点信息，分别绘制了特征点分组结果随帧数的变化情况以及截取不同的帧作为起点的情况下关于特征点分组结果随帧数的变化情况，如图 8-7 所示。图中给出了不同帧的特征点分类情况，以最小矢量差为聚类标准，可以克服单帧信息中出现点群交叉的计算情况。进一步观察图 8-8 中特征点分布情况可以发现，公交车以及周围树后的物体在做类直线运动，而出租车在经过一点时间的类直线运动之后开始做变速转弯运动。在最后一帧中，不同类别的特征点的个数为（138，92，67）。

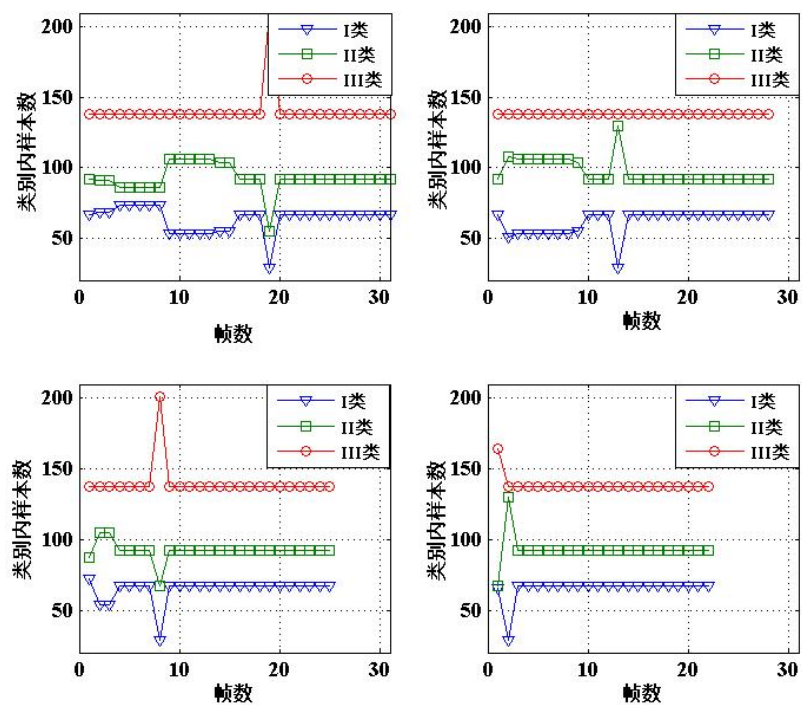


图 8-7 不同帧起点情况下分类结果随帧数增加的变化曲线图

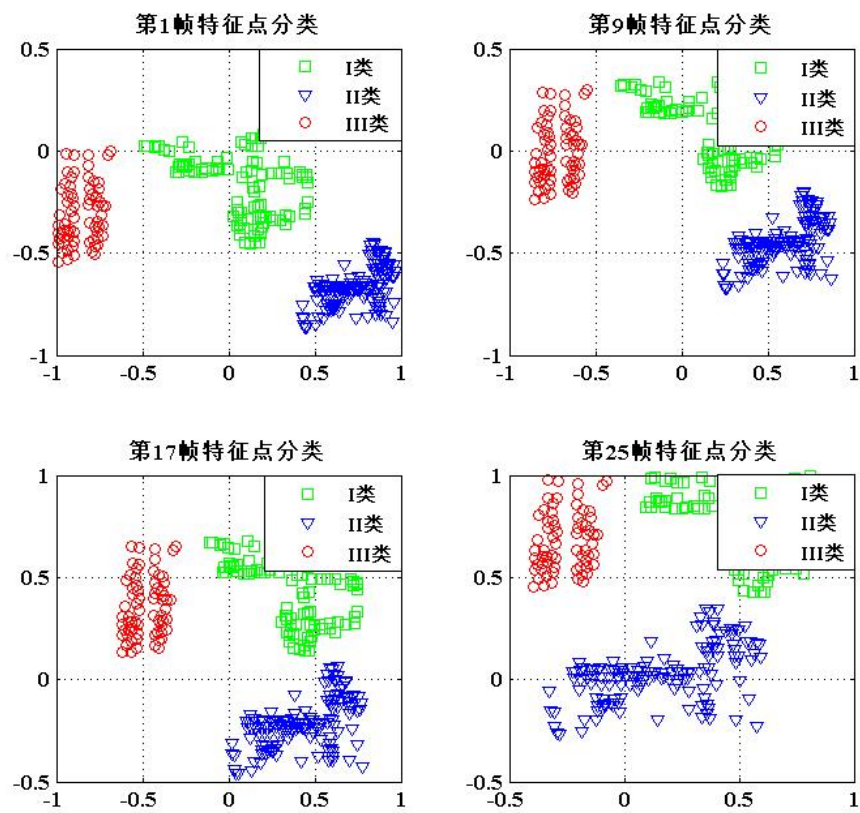


图 8-8 不同帧下的特征点分类结果示意图

结果分析与讨论:

- a) 对于同一起始帧, 当统计帧数较少时, 聚类结果表现出一定的抖动性, 但保持了其大致的浮动范围, 在一些恶劣情况下, 如运动物体转弯时等, 会是原本稳定的结果产生较大的抖动; 但是, 这种受迫抖动很快就会消失, 根本上在于采用最小矢量差模型, 可以充分地运用物体运动的连续性性质, 有效地消除异常结果;
 - b) 当采用不同起始帧时, 结果表现出很大的相似性, 即经过一定帧数的抖动后都到达稳定状态, 从而有力地说明了该模型能够克服对于起始帧特征点的依赖性, 具有更加普遍的适用性。
-

8.3 人脸图像分类问题

人脸识别是一个多学科交叉的研究方向, 涉及的学科领域包括人工智能、模式识别、计算机视觉、图像分析与处理、生理学、心理学及认知科学等。从本质上来说, 作为一个非常理想的研究对象, 人脸为模式识别、人工智能和计算机视觉等学科领域提供了用于验证相关方法和算法是否有效的研究素材。对人脸识别问题的研究和解决, 有助于这些研究领域的发展以及相关识别问题的解决, 具有重要的理论研究价值。另一方面, 作为一个非常具体的研究对象, 人脸识别为以和谐自然的方式实现身份认证提供了依据和途径, 它在和身份认证相关的诸应用领域具有广阔的应用前景。

8.3.1 数据预处理

数据为两个人在不同光照下的人脸图像共 20 幅 (矩阵的每一列为拉成向量的一幅人脸图像), 用列向量来表示人脸图像, 尽管降低了图像的维度, 但实际上是以损失图像信息为代价。人脸图像上的任何一点, 在二维平面内跟其他周围的临近点具有天然的联系性, 如亮度的连续性等, 因此首先需要将表示人脸图像的一维列向量 X_i 扩展为二维矩阵 Y_i ; 同时为了加速算法的收敛性, 需要对所有数据进行去均值化处理。图 8-9 给出了经过数据预处理后的人脸图像结果, 图中按照原数据中的顺序分别给出了数据处理后的人脸图像:

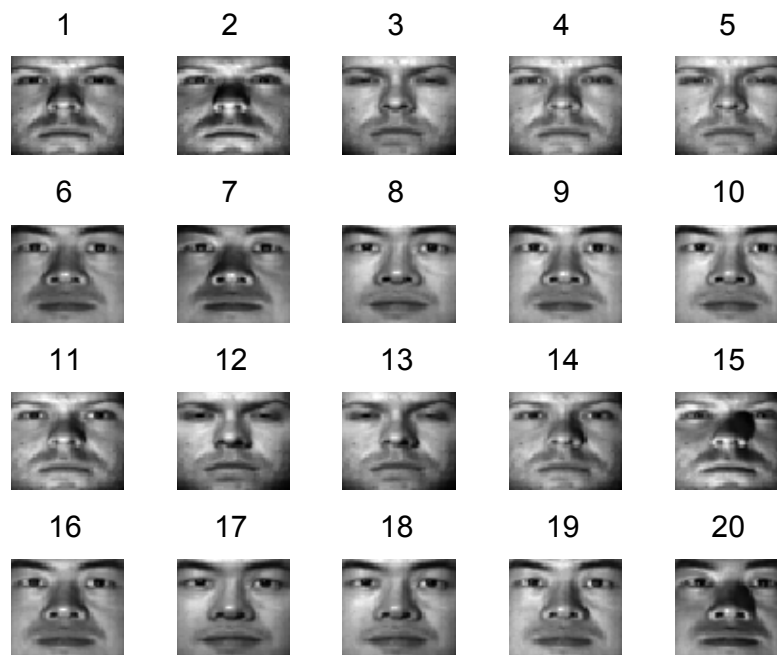


图 8-9 数据预处理后的人脸图像图

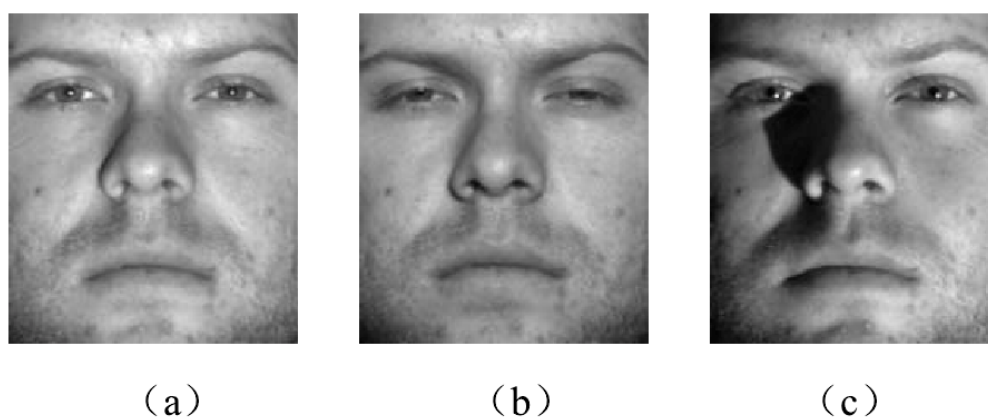


图 8-10 人脸图像由于表情和光照引起的局部的变化





















人脸由于光照条件、面部表情等因素引起的变化，往往只体现在图像的部分区域，而其它部分的变化很少甚至无变化。如图 8-10 所示，在此图中，（a）到（b）之间的表情变化只体现在人眼部分，而由（a）到（c）之间的光照变化主要体现在图像的左半部分。

8.3.2 问题求解

我们知道，尽管在不同光照或者表情等情况下，但同一人脸具有固有的本质特征，因此在全局范围内进行聚类，能够充分地利用这一本质特征，实现不同人脸图像的归类。本文题仍以谱多流形聚类算法为基础，在这一具体问题中，实为全局的谱多流形聚类。表 8-2 给出了全局谱多流形求解的聚类结果，可以发现

20 个人脸图像被区分为两类，每一个中均有 10 个人脸图像；进一步观察可以发现，不同类别中的人脸图像具有明显的地缘特征，具体来说，具有欧美地缘特征的人脸图像被归为一类，而具有亚洲地缘特征的人脸图像被归为另一类。从人的感官角度出发，不同地缘特征的人总能简单直接的识别出其他地缘特征的人，这与我们上述求解的结果具有一致性，从而验证了上述求解思路的有效性。

表 8-1 人脸图像分类结果

类别	分类结果				
I 类	1	2	3	4	5
					
	11	12	13	14	15
					
	6	7	8	9	10
					
	16	17	18	19	20
					
II 类					

8.3.3 求解过程的简化：数据再处理

谱多流行聚类算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、计算相似性矩阵 W 、利用谱方法进行聚类，可以表示为：

$$O(N^3 + N^2(Dd^2 + K + k) + N(DM(t_1 + dt_2) + k^2t_3)) \quad (8-2)$$

算法收敛所需的迭代步数通常较低(少于 50), 在数据维数基本确定的情况下，复杂度主要由数据点数 N 决定。

问题中每一个人脸图像由 42×48 个像素构成，随着现代储存技术的发展以及人们对于图像的真实性要求，构成图像的像素数量越来越多，一张分辨率为 640×480 的图片，那它的像素就达到了 307200，也就是我们常说的 30 万像素，而一张分辨率为 1600×1200 的图片，它的像素就是 200 万，在这种像素量的情

况下，上述算法复杂度的增加是不可接受的，因此需要对原始数据进行再处理，在损失少量信息的情况下，实现算法复杂度的有效降低。

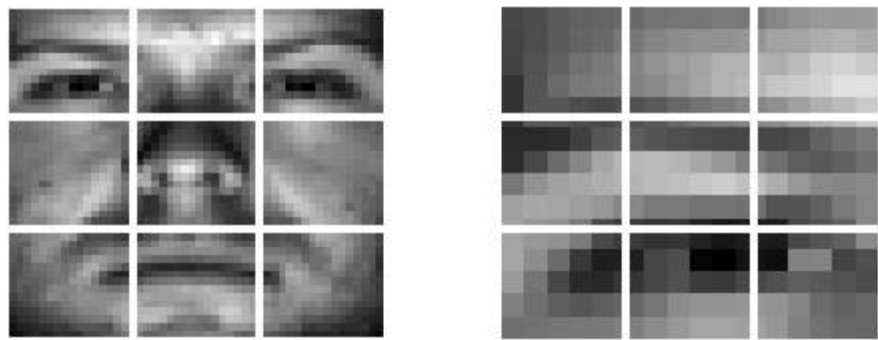


图 8-11 人脸图像的数据再处理示意图

图 8-11 中给出了单个人脸图像再处理过程，为了降低原始数据的像素数量，同时保留其基本的特征信息，可以将相邻的像素单元看成是一个新的像素单元，用处理后的像素单元信息代表融合前所有像素单元的信息。如图所示，所描述的人脸图像像素点个数可以由最初的 42×48 降低为 $3 \times 3, 9 \times 9 \dots$ ，这对于预算复杂度的降低是极为有益的。需要注意的是，在不同的情形下，需要采取不同的降维方式来满足分类需求。如在差异性不是很明显的情况下，降维的幅度要适当控制来保证信息的有效性。图 8-12 给出了采用了降维方式后，人脸图像的初始数据结果，表 8-2 给出了数据经过 $1/9$ 压缩处理后的分类结果，相比之下，人脸头像的分辨率明显降低许多；可以发现，经过适当降维后，在明显降低算法复杂度的同时，实现了与前述相同的分类结果，从而验证了该数据再处理方式的有效性。

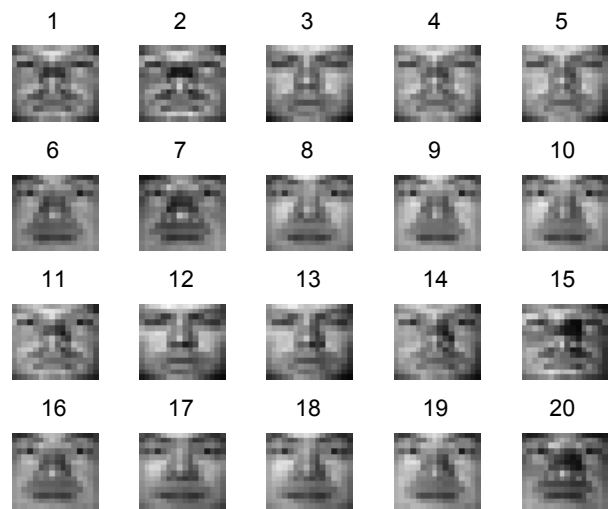






















图 8-12 人脸图像的数据 1/9 压缩再处理示意图

表 8-2 人脸图像的数据 1/9 压缩再处理后分类表

类别	分类结果				
I 类	1	2	3	4	5
					
	11	12	13	14	15
					
II 类	6	7	8	9	10
					
	16	17	18	19	20
					

9 问题四：实际应用中的多流形聚类问题的探讨

多个流形数据形成的空间结构可以细分为良分离结构、相互交叠结构和混合结构(既有良分离的结构又有相互交叠的结构)。当聚类子集可以用一组具有线性结构的流形(线性或仿射子空间)来表示时,由于可以建立模型来显式地描述线性流形,通常可以不管流形数据的空间结构,例如 GPCA、KF、LSA 和 SCC 等线性流形聚类方法。然而当数据分布高度非线性时,由于数据分布的非线性特性和流形结构的复杂性,直接建立显式的模型进行求解变得不可能,通常需要将数据结构分为良分离的或相互交叠的来分别处理,例如传统的谱聚类方法和 K-manifolds 方法。然而真实数据中混合模型才是最常见的,也是最合理最理想的数据结构假设。同时混合模型也是对传统良分离模型和相互交叠模型的自然理论。完整的混合流形聚类问题可以形式地描述为:

定义 5.1 给定来自于 k 个不同潜在流形(线性/非线性) $\Omega_j \subseteq \mathbb{R}^D (j=1, \dots, k)$ 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 $\mathbb{Z} = \{x_i \in \mathbb{R}^D\}_{i=1}^N$ 。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 k 及其本征维数 $d_j, j=1, \dots, k$;
2. 将给定的数据采样划分到其所属的潜在流形。

问题四中给出了实际应用场景,三维圆台的点云分类以及机器工件轮廓线的分类,两种情形采样于混合流形结构的数据集,两个均是非线性流形结构的组合。

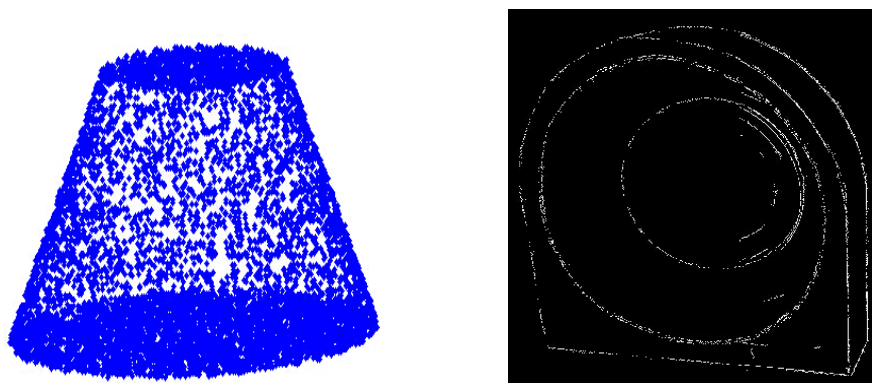


图 9-1 两个实际场景示意图

9.1 三维点云的分类问题

三维点云问题的难点在于，顶、底面点云与侧面点云不仅相互交叠而且交叠的区域非常多，前文已经分析过相互交叠的区域在分类时具有一定的随机性，而且如果处理不好会使分类结果产生严重的偏差。一种直接的方法是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。然而该模型交叠的部分区域非常大，交叠的区域已经构成了其中一个分类的界限，因此希望通过拆开流形结构来实现混合流形聚类的方式是不合适的。

9.1.1 谱多流形聚类方法分类圆台点云

采用常规的算法难以区分交叠区域的一个根本原因在于，交叠区域点之间的距离非常接近，无法构成分类的有效指标；因此需要进一步挖掘数据的有效信息。正如前文所述，谱多流形聚类方法正是基于这一出发点，在设计相似性矩阵时充分挖掘了点集之间的潜在联系，其特征简要概括如下：

-
- a) 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；
 - b) 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；
 - c) 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。
-

值得注意的是，只有当下面的两个条件同时满足时,我们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。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(称为局部相似性，又要考虑数据点局部切空间之间的相似性。这两个相似性融合在一起来决定最后的相似性权值：

$$w_{ij} = f(p_{ij}, q_{ij}) \quad (9-1)$$

其中 f 是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。下面给出算法的具体求解过程：

圆台点云的分类过程:

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 α 。

算法过程:

- 确定每个点的局部切空间;
- 计算两个局部切空间之间的结构相似性;
- 计算相似性矩阵 W , 并计算对角矩阵 D ;
- 计算广义特征矩阵 $(D-W)u = \lambda Du$ 最小 k 个特征值对应的特征向量;
- 利用 K-means 将行向量分组为 k 个聚类。

输出: 原始数据对应的聚类结果。

9.1.2 实验结果与分析

实验平台为: Intel 8 核系统、4*2.3GHz CPU 8GB 内存。本节的谱聚类采用谱多流形聚类算法, 其中相似性矩阵的构造采用热核, 热核的尺度参数设置在一个较大的范围内(即从 10^{-5} 到 10^5 , 步长为 $10^{0.5}$), 实验中汇报上述参数范围内谱聚类的最好结果^[9]。图 9-2 中给出了聚类原始数据与最终聚类结果对比图, 观察图中结果发现, 采用谱多流形聚类算法能够将圆台的点集有效地分成三类。完成分类的点集可以准确地实现对原始物体的重构和分析, 从而有效地完成如摄像头监测、轨迹预测等功能。

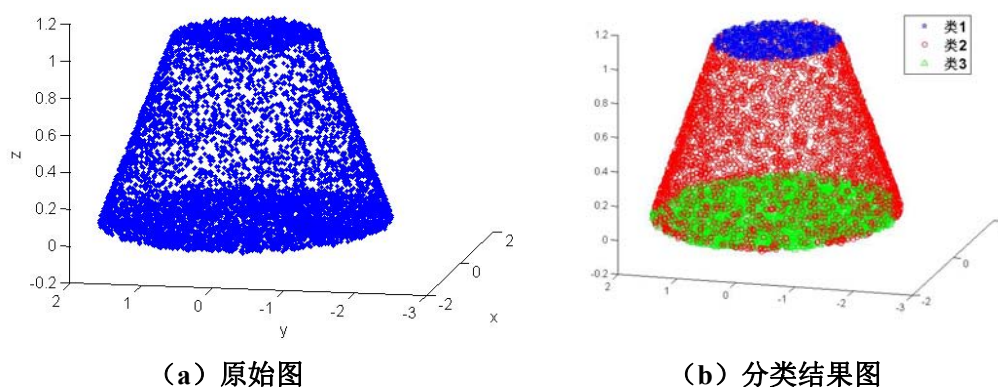


图 9-2 圆台表面点集的分类结果对比示意图

9.2 工件轮廓线的分类问题

首先，采用谱多流形聚类算法对工件轮廓线进行分类，设定将其分为 5 类，可得到如图 9-3 所示的分离效果。图中四个圆圈处位置标记的数据分类发生了混淆，这些数据虽属于不同的流形，但却被近邻图连通在了一起。这是因为当两个数据点 \mathbf{x} 和 \mathbf{y} 非常靠近时，即使它们来自于不同的流形，构造出的局部切空间 Θ_i 和 Θ_j 也会非常相似，其原因在于，在这种情况下 \mathbf{x} 和 \mathbf{y} 的基于欧氏距离度量的局部近邻 $N(\mathbf{x})$ 和 $N(\mathbf{y})$ 会严重地交叠在一起，从而导致了相似的局部协方差矩阵 $\Sigma_{\mathbf{x}}$ 和 $\Sigma_{\mathbf{y}}$ 。

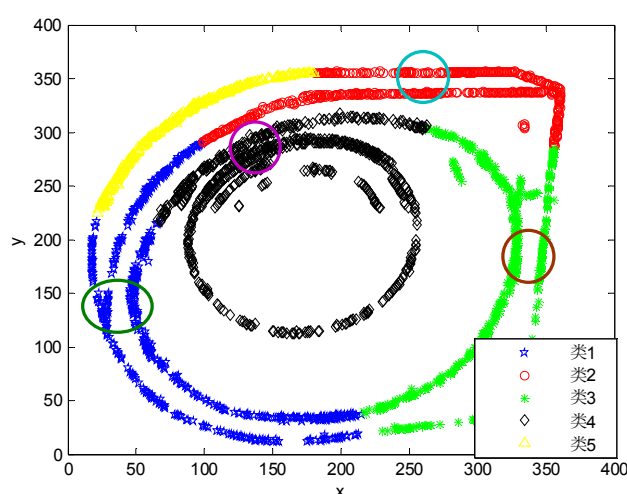


图 9-3 谱多流形聚类算法对工件轮廓线的分类示意图

基于上述分析，为了利用谱聚类方法来分组混合结构数据，需要构造更符合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。由于不可靠的无向近邻图通常来自于相互交叠的流形结构。因此，采用分部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，然后再重点解决混合结构的困难部分。具体地说，首先从整个数据中分离出不同的连通或可分离子集，从而将由单一流形构成的纯粹子集和由相交流形构成的交叠子集区分开来，然后进一步将相交的子集分割为相交区域和非相交区域，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可靠的无向近邻图，最后用谱聚类方法得到最后的聚类结果。该算法的基本流程如图 9-4 所示，其具体过程如下：

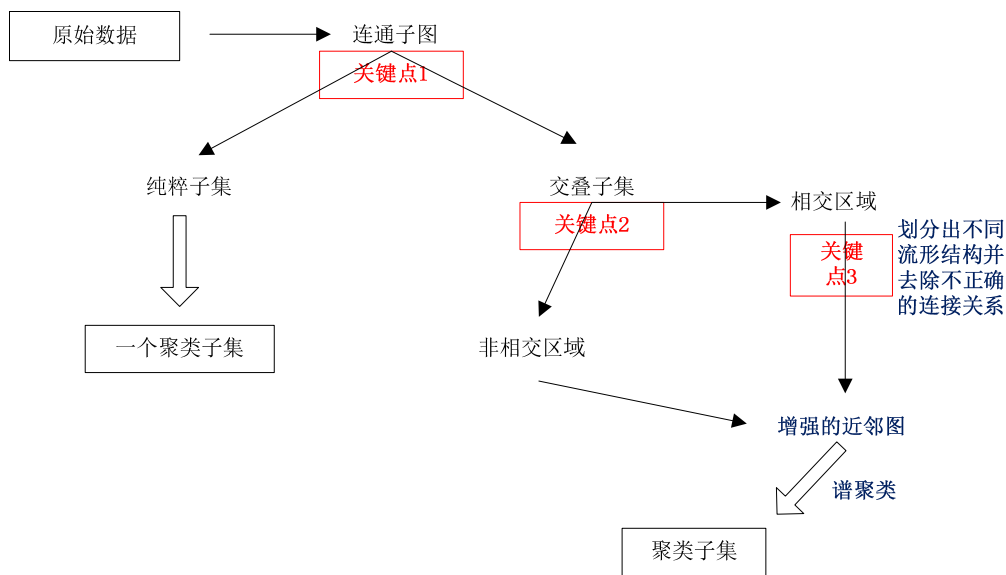


图 9-4 分部处理混合流形聚类的基本流程

(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,另一些是由相交流形构成的交叠子集。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)。

(2) 确定连通子集的结构: 在得到了不同的连通子集后, 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。

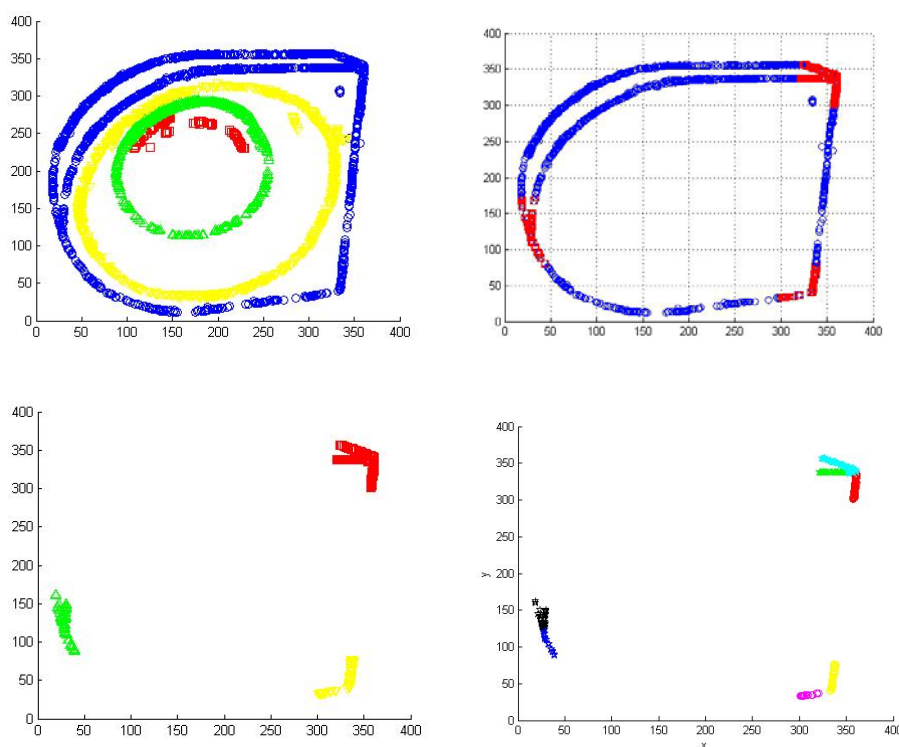
(3) 确定交叠子集的相交区域和非相交区域: 如果一个连通子集是纯粹的, 那么我们实际上已经得到了一个流形聚类。然而, 更困难的问题在于交叠子集, 我们需要进一步处理它以得到该子集中的不同流形聚类。这里的另一个关键点是如何找出数据的相交区域和非相交区域。通常, 相交区域的数据点由于有其它流形结构上的数据点存在, 其估计的维数会大于流形的真实维数。因此, 我们将具有最高估计维数的数据点作为相交区域中的点。

(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和非相交聚类, 同样需要找出这些不同的子聚类并进行分别处理。

(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)。

(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。

第一步粗聚类将独立的子集分离出来,第二步将重叠部分提取出来单独进行细聚类, 剩余的点集又可以构成 N 个独立子集便于聚类, 最后将两部分聚类结果进行整合。



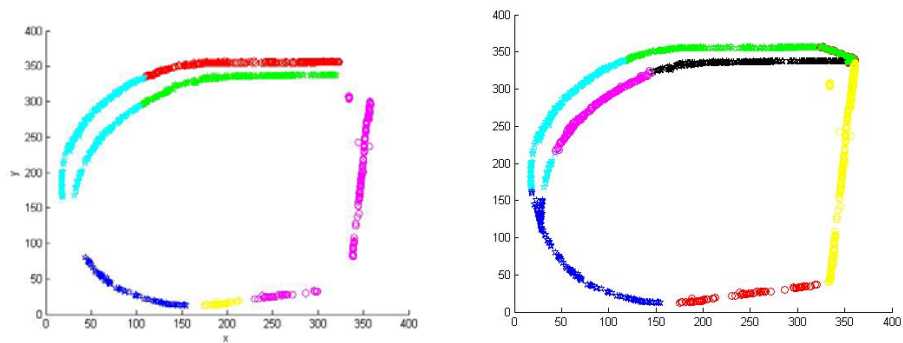
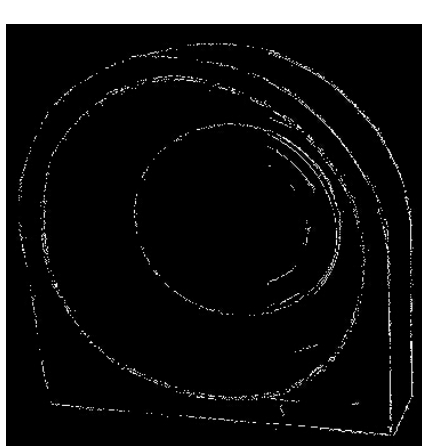
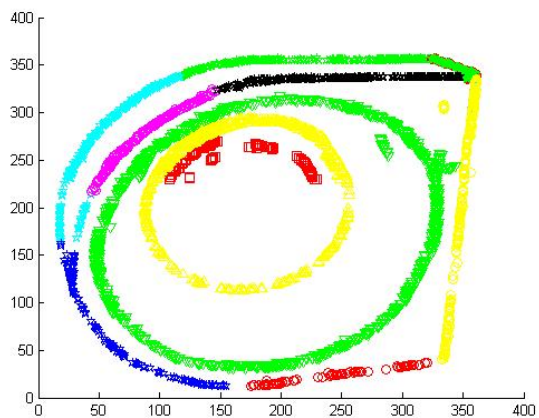


图 9-5 分部聚类的基本过程

分部谱聚类算法对工件轮廓线分类结果如图 9-6 所示，最终可将其分为 10 类，包括 4 类直线和 6 类圆弧线。



(a) 原图



(b) 分类

图 9-6 分部谱聚类算法分类结果图

10 总 结

本文主要围绕数据聚类问题展开,从经典的谱聚类算法处理高维数据问题入手,为了满足不同的应用需求,不断完善谱聚类算法的适用性,主要工作如下:

1. 改进相似矩阵构造过程,采用共享近邻的自适应谱聚类模型完成对高维数据的分类,克服了传统谱聚类算法依赖于参数的手工设置;
2. 采用谱多流形聚类算法实现对交叉图形的有效分割,研究了算法中主要参数对分类性能的影响,并分析了算法的复杂度;
3. 解决了宽十字交叉的分割问题,验证了谱多流形聚类算法对于交叉区域的鲁棒性;建立了基于最小矢量差的谱多流形聚类模型,完成了多帧图像运动特征点的分割;最后,基于谱多流形聚类模型完成了对不同人脸图像的分割,并提出了降低算法复杂度的有效办法;
4. 最后,采用谱多流形聚类模型解决了立体圆台的多流形聚类问题,同时为了解决工件的轮廓线分类问题,在现有模型的基础上,提出了提高正确率的改进意见。

参考文献

- [1] Von Luxburg U. A tutorial on spectral clustering [J]. Statistics and Computing, 2007, 17(4): 395-416.
- [2] 李庆扬. 数值分析[M]. 清华大学出版社, 2008.
- [3] 贾建华. 谱聚类集成算法研究[M]. 天津大学出版社, 2001.
- [4] 蔡晓妍, 戴冠中. 谱聚类算法综述[J]. 计算机科学, 2008, 35(7): 14-18.
- [5] Von Luxburg U. A tutorial on spectral clustering [J]. Statistics and Computing, 2007, 17(4): 395-416.
- [6] David J. Marchette et al. Random Graphs for Statistical Pattern Recognition [M]. A John Wiley&Sons, Inc, Publication, 1998.
- [7] Amigó E, Gonzalo J, Artiles J, et al. A comparison of extrinsic clustering evaluation metrics based on formal constraints [J]. Information retrieval, 2009, 12(4): 461-486.
- [8] 王勇. 基于流形学习的分类与聚类算法及其应用研究[D], 国防科技大学博士学位论文, 2011.
- [9] 薛定宇, 陈阳泉. 高等应用数学问题的 MATLAB 求解[M], 清华大学出版社, 2004.

附 录

附录 1:

表 1- 高维数据点分类表

编号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
标记	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
编号	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
标记	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
编号	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
标记	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
编号	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
标记	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
编号	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
标记	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
编号	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
标记	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
编号	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140
标记	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
编号	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
标记	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
编号	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
标记	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
编号	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
标记	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2




















附录 2:

表 2- 特征点轨迹分类表.

编号	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.
标记	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.
编号	21.	22.	23.	24.	25.	26.	27.	28.	29.	30.	31.	32.	33.	34.	35.	36.	37.	38.	39.	40.
标记	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.
编号	41.	42.	43.	44.	45.	46.	47.	48.	49.	50.	51.	52.	53.	54.	55.	56.	57.	58.	59.	60.
标记	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.
编号	61.	62.	63.	64.	65.	66.	67.	68.	69.	70.	71.	72.	73.	74.	75.	76.	77.	78.	79.	80.
标记	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.
编号	81.	82.	83.	84.	85.	86.	87.	88.	89.	90.	91.	92.	93.	94.	95.	96.	97.	98.	99.	100.
标记	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.
编号	101.	102.	103.	104.	105.	106.	107.	108.	109.	110.	111.	112.	113.	114.	115.	116.	117.	118.	119.	120.
标记	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.
编号	121.	122.	123.	124.	125.	126.	127.	128.	129.	130.	131.	132.	133.	134.	135.	136.	137.	138.	139.	140.
标记	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	2.	2.	2.
编号	141.	142.	143.	144.	145.	146.	147.	148.	149.	150.	151.	152.	153.	154.	155.	156.	157.	158.	159.	160.
标记	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
编号	161.	162.	163.	164.	165.	166.	167.	168.	169.	170.	171.	172.	173.	174.	175.	176.	177.	178.	179.	180.
标记	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
编号	181.	182.	183.	184.	185.	186.	187.	188.	189.	190.	191.	192.	193.	194.	195.	196.	197.	198.	199.	200.
标记	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
编号	201.	202.	203.	204.	205.	206.	207.	208.	209.	210.	211.	212.	213.	214.	215.	216.	217.	218.	219.	220.
标记	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	3.	3.	3.	3.	3.	3.	3.
编号	221.	222.	223.	224.	225.	226.	227.	228.	229.	230.	231.	232.	233.	234.	235.	236.	237.	238.	239.	240.
标记	3.	3.	3.	3.	3.	3.	3.	3.	3.	2.	3.	2.	3.	3.	3.	3.	2.	3.	2.	3.
编号	241.	242.	243.	244.	245.	246.	247.	248.	249.	250.	251.	252.	253.	254.	255.	256.	257.	258.	259.	260.
标记	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	2.	2.	3.	3.	3.	3.	3.	3.	3.	3.
编号	261.	262.	263.	264.	265.	266.	267.	268.	269.	270.	271.	272.	273.	274.	275.	276.	277.	278.	279.	280.
标记	3.	3.	3.	2.	3.	3.	2.	2.	3.	2.	3.	2.	3.	3.	2.	2.	3.	3.	3.	3.
编号	281.	282.	283.	284.	285.	286.	287.	288.	289.	290.	291.	292.	293.	294.	295.	296.	297.			
标记	3.	2.	3.	3.	3.	3.	2.	3.	3.	3.	3.	3.	3.	3.	3.	2.				

附录 3:

表 3- 人脸图像分类表

编号	1	2	3	4	5
					
类别	1	1	1	1	1
编号	6	7	8	9	10
					
类别	2	2	2	2	2
编号	11	12	13	14	15
					
类别	1	1	1	1	1
编号	16	17	18	19	20
					
类别	2	2	2	2	2

