

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 中国矿业大学

参赛队号 10290012

	1.	徐云靖
队员姓名	2.	冯 乐
	3.	师庆民

参赛密码 _____
(由组委会填写)



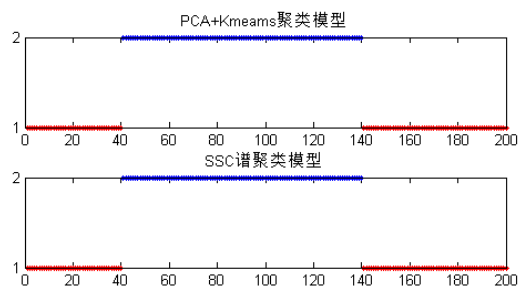
第十二届“中关村青联杯”全国研究生 数学建模竞赛

题 目 数据的多流形结构分析

摘 要：

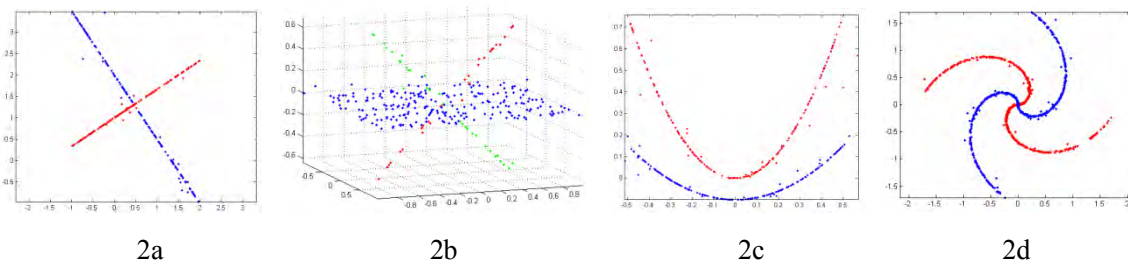
几何结构分析是进行数据处理的重要基础，基于谱聚类算法的多流形结构分析是解决几何结构分析的主要方法之一，本论文充分探讨了主成分分析（Principal Component Analysis, PCA）+K-means 聚类模型、共享近邻谱聚类（Shared Nearest Neighbors, SNN）、稀疏子空间聚类（Sparse Subspace Clustering, SSC）、多流形谱聚类（Spectral Multi-manifolds Clustering, SMMC）、稀疏流形聚类与嵌入模型（Sparse Manifold Clustering and Embedding, SMCE）和变色龙聚类模型（CHAMELEON Clustering）等聚类模型，并针对各个问题，分析具体分类目标与各个模型算法特点，对四个问题分别进行了求解，最终对模型进行评判并提出改进意见。

针对问题一，将采样于两个独立子空间的高维数据（200*100）进行聚类。本论文采用了传统的主成分分析（PCA）+Kmeans 聚类算法，将 $N=200$ ， $D=100$ 的高维数据用前 20 维（贡献度为 98.65）标记，并分为两类；并与稀疏子空间聚类（SSC）模型进行对比，二者分类结果如下图所示，结果一致；



1

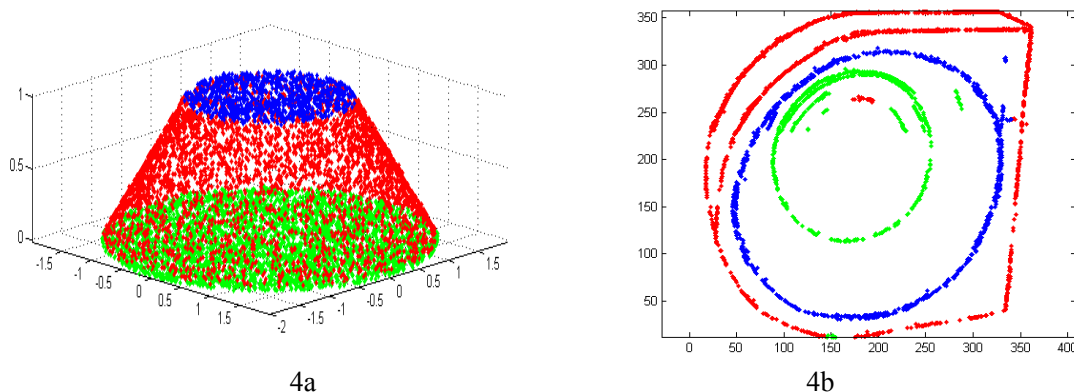
针对问题二，主要是对四个低秩子空间和多流形低秩子空间进行聚类。分别利用 SSC、SMMC 及 SNN 聚类方法，成功解决低维空间下非线性子空间的聚类问题（如下图所示），并通过模型特点分析了各自的优劣。认为 SSC 在解决直线子空间交叉问题更占优势，而 SMMC 及 SNN 在解决流形聚类上效果显著。同时，SMMC 更适合解决具有交叉的子空间问题。



针对问题三，在问题二对典型子空间和多流形问题成功聚类的基础上，解决实际问题。题 3.1 基于 SSC 的谱聚类算法，成功将特征提取环节中处理得到的十字上的点位置信息提取并分成两类；题 3.2 基于特征点轨迹追踪对视频的 31 帧运动进行运动分割，定义并基于原始数据生成了一个基于偏移量的运动轨迹特征矩阵，采用 SNN 谱聚类算法，将 297 个特征点准确分成 3 类，并还原运动轨迹；题 3.3 针对 2016 维人脸向量图，分别采用不降维和利用 PCA 降维至 9 维（贡献度为 99.67）的方法，基于 SSC 谱聚类算法对 20 张人脸进行分类和标记，并取得了一致的结果（标签列表如下所示）。

1	1	1	1	1	2	2	2	2	2	1	1	1	1	1	2	2	2	2	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

针对问题四，深入探索了谱聚类解决更加复杂的几何结构问题的适用性。分析题 4.1 中图形的几何特征，通过模型调整几何结构中更具有决定性的几何量的权重，采用 SNN 谱聚类算法，成功对图形进行聚类（如下图所示）；分析题 4.2 中图形的几何特征，选用 Chameleon 算法和 SMCE 谱聚类算法分别对图形轮廓进行聚类，其结果表明 SMCE 谱聚类算法不仅能够图形轮廓线中同一流形的连续点进行聚类，并且有噪声干扰的情况下，对于较外的轮廓线，其分类能力较 Chameleon 聚类算法更加显著（如下图所示）。



最后，基于 4 个问题的求解与分析，本文认为现阶段谱聚类在图形数据分类中仍然是十分有效的方式，通过将来自不同子空间的高维数据分割到所属的低维子空间中进行聚类，解决了维度高带来的复杂与稀疏问题。但是由于所处低维空间的流形特征与子空间相交的问题，给聚类带来极大困难，目前并没有统一模型同时对多流形与交叉子空间分类进行合理的解决。不论通过低维度的同流形判断还是通过局部密度的相似矩阵建立，都受到了噪声点与数据点缺失以及奇异样本等的影响。算法改进思想在于对噪声和缺失点的判断与消除，通过多次降维或是依据统计分布情况，筛除缺失值和噪声点，同时正则项的适当设定，让数据可以有先验信息，来达到更为准确的判断。

关键词：谱聚类；流形；稀疏子空间；降维；运动分割；正则项

目 录

一 问题重述.....	1
二 基本假设及说明.....	2
三 基本符号说明.....	2
四 问题分析.....	3
4.1 问题一分析.....	3
4.2 问题二分析.....	3
4.3 问题三分析.....	3
4.4 问题四分析.....	4
五 模型特点介绍.....	5
5.1 PCA+K-means 聚类模型.....	5
5.2 谱聚类 (Spectral Clustering).....	5
5.2.1 基于共享近邻的谱聚类 (SNN).....	6
5.2.2 稀疏子空间聚类 (SSC).....	6
5.2.3 多流形谱聚类模型 (SMMC).....	7
5.2.4 稀疏流形聚类与嵌入模型 (SMCE).....	8
5.3 CHAMELEON 聚类.....	8
六 问题求解.....	9
6.1 问题一求解.....	9
6.2 问题二求解.....	10
6.2.1 图 a 求解.....	10
6.2.2 图 b 求解.....	11
6.2.3 图 c 求解.....	12
6.2.4 图 d 求解.....	13
6.3 问题三求解.....	15
6.3.1 题 a 求解.....	15
6.3.2 题 b 求解.....	15
6.3.3 题 c 求解.....	18
6.4 问题四求解.....	19
6.4.1 图 a 求解.....	19
6.4.2 图 b 求解.....	20
7 模型评价与改进.....	23
参考文献.....	25
附 录.....	26

数据的多流形结构分析

一 问题重述

在信息爆炸的新时代，数据挖掘作为新兴交叉学科在机器学习、模式识别等领域取得了丰富的研究成果。

在对数据间隐藏关系、模型进行识别时，聚类分析是数据挖掘中非常有效和普遍的分析方法。聚类分析将数据分类到不同的类或者簇，使同一簇的对象间有很大的相似性，而不同簇的对象间有很大的相异性^[1]。聚类分析在人脸识别、手写体数字识别、图像分割、运动分割等计算机视觉、图像处理中应用广泛。聚类的过程按是否存在指导性督促学习等知识，可将聚类划分为无监督和有监督学习聚类。从概念可知，无监督聚类由于数据样本的类属信息，仅仅依靠样本间的相似性进行类属划分，因此具有比较差的预测结果。

传统的聚类算法有 K-means 和 EM 算法，但当样本空间非凸时，此类算法陷入局部的最优解^[2]。与传统的聚类分析相比，谱聚类在处理样本空间不为凸时，因其实现简单，对任意形状的数据空间表现出鲁棒性，且能收敛与全局最优而备受关注^[3]。谱聚类的基本思想是利用样本数据的相似矩阵进行特征分解后得到的特征向量进行聚类。

由于谱聚类算法本身存在许多适用性问题，将其应用到图像识别具有一定的挑战性。其中两个关键问题为相似矩阵构建方法和图像分割，相似矩阵构建算法对谱聚类性能起着重要的影响。

本文提出的问题包括：

问题一：将采样于两个独立子空间的高维数据进行聚类，分成两类。

问题二：对四个低维空间的子空间和多流形进行聚类。图 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 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 1(c)为两条不相交的二次曲线，请将其分为两类。图 1(d) 为两条相交的螺旋线，请将其分为两类。

问题三：（1）如图 2(a)所示，十字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，。请使用适当的方法将十字中的点按照“横”和“竖”分为两类。（2）图 2(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。

（3）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。

问题四：（1）图 3(a)分别显示了圆台的点云，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)。（2）图 3(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。

二 基本假设及说明

- 1、空间上相近的数据点应具有较高的相似度
- 2、位于同一个簇中的数据点应该具有较高的相似度。
- 3、稀疏子空间聚类假设高维空间中的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。
- 4、针对本题均为无监督学习聚类，因此没有很好的方法检验分类模型的好坏。由于题目中图形数据和分类要求已知，因而更多采用模型分类图形与对投点图形的人为分类进行对比，来评判模型的优劣。

三 基本符号说明

符号	说明
F_p	主成分分析中所导出的主成分形成的新综合指标，表示第 p 个综合指标
a_i	主成分分析中 X 的协方差阵 Σ 的特征值所对应的特征向量
Z_{xi}	主成分分析中原始变量经过标准化的值
x_i	谱聚类中的第 i 个数据点 x
w_{ij}	谱聚类中数据点 x_i 和 x_j 的相似度
$\ x_i - x_j\ $	谱聚类中数据点 x_i 和 x_j 的欧氏距离
δ	谱聚类中的尺度参数
$W(X_i, \bar{X}_i)$	谱聚类中邻近的关联矩阵
S_N	共享近邻谱聚类中的相似度
$\ Z\ _1$	系数矩阵 Z 的 l_1 -范数
$F(E)$	稀疏子空间谱聚类中的数据项或保真项
$R(Z)$	稀疏子空间谱聚类中的正则项或惩罚项
$\ X\ _2$	X 的欧几里得范数
$RI(C_i, C_j)$	CHAMELEON 聚类中 C_i 和 C_j 两簇数据的互连性
$RC(C_i, C_j)$	CHAMELEON 聚类中 C_i 和 C_j 两簇数据的相对紧密度
$ Z $	矩阵 Z 的元素取绝对值得到的矩阵

四 问题分析

4.1 问题一分析

问题一要求将附件一中 1.mat 中有一组高维数据分为两类。

在这组高维数据中，每个数据点具有 100 个属性，即 100 个维度。但有意义的属性仅存在于少数维度所组成的子空间中，因此可以采用主成分分析（Principal Component Analysis, PCA）过滤冗余属性，而后通过 K-means 基于欧氏距离进行聚类。

此外，稀疏子空间谱聚类（Sparse Subspace Clustering, SSC）假设高维空间中的数据本质上属于某个低维子空间，可以在低维子空间中进行线性表示。因此，也可以有效表示高维数据的子空间聚类^[4]。

4.2 问题二分析

问题二的第一小问要求将图 1(a)为两条交点不在原点且互相垂直的两条直线分为两类；第二小问要求将图 1(b)为一个平面和两条直线。两个问题均为子空间相交或者关联问题，使得一个点的邻域包含其他其它子空间的点，采用传统的谱聚类算法很容易使不同的簇在交集点被无向图连接，而稀疏子空间谱聚类（SSC）利用数据自表示可以很好的解决直线的和不同子空间数据间有关联的问题，可以合理的选择数据点邻域大小和处理交集附近的点^[5]。

问题二的第三小问要求将图 1(c)两条不相交的二次曲线分为两类。该问题为曲线不相交流形问题，共享近邻谱聚类（Shared Nearest Neighbors, SNN）通过表征两点间局部密度构建出相似度，从而可以很好的获得同一流形上的数据点的相似度高于不相交的不同流形上的数据点间的相似度^[6]。问题二的第四小问要求将图 1(d) 的两条相交的螺旋线分为两类。混合空间结合了第一小问的子空间关联或交集问题和第三小问的曲线流形问题，采用稀疏子空间聚类不能很好的解决曲线流形问题，而采用共享近邻聚类则不能很好的解决不同子空间数据点交集问题。因此，结合稀疏子空间谱聚类（SSC）的关联矩阵和共享近邻谱聚类（SSN）局部密度的方法，多流形谱聚类（Spectral Multi-manifolds Clustering, SMMC）可以解决相交流形分类问题^[7]。

值得说明的是，多流形谱聚类算法存在概率性问题，即同一条件下存在多个分类结果，在参数选取和方案选取中耗时较长。因此，针对本题，可以在比较成熟的共享近邻谱聚类的基础上将螺旋图空间数据划分为 4 类，然后对 4 组数据两两合并，并选取合适的分类。此方法相对多流形谱聚类更加简便省时。

4.3 问题三分析

问题三第一小问要求将 2(a)图中十字上的数据点分为“横”“竖”两类，该问题与问题二第一小问相似，为直线型混合子空间相交问题。因此，同样可采用稀疏子空间谱聚类的方法。

问题三第二小问要求将静态的树、运动的公交车和运动的小汽车三个不同运动物体的运动轨迹区分开来。该问题与上述问题不同，要求对运动物体进行分类，其本质是要求同一子空间的数据点具有相同的运动规律。因为部分不同子空间的数据点之间存在交集或欧氏距离很近的特征，即相似度可能很高，而同一子空间的数据点间可能因为欧氏距离较远而相似度较低，采用传统的静态谱聚类方法强行分类可能会造成同一运动物体上提取的数据点出现运动趋势不一致的情况。

同一物体上的数据点运动规律一致的本质在于同一子空间内的数据点具有相同的运动速度。因此，可以通过路径对时间积分的方式对混合空间数据点进行预处理，获得

速度向量，然后利用共享近邻谱聚类（SNN）的方法对速度向量进行聚类。进而将原始空间数据点分为三类。

问题三第三小问要求对 20 张人脸图像分为两类。该数据结构表征为 2016 维的向量图，属于高维混合子空间。因此，可利用主成分分析（PCA）去除原始数据中无关的属性，然后利用稀疏子空间谱聚类（SSC）进行求解。根据文献[5]，稀疏子空间谱聚类同样可直接处理高维数据，且结果精度更高。最终将分类方案与脸谱灰度图进行对应。

4.4 问题四分析

问题四第一小问要求将圆台点云的顶、底、侧面分为三类。侧面数据点与顶、底子空间数据点属于曲面交集问题，使用传统的谱聚类处理相对困难。因此，需要对数据进行预处理。分析图像可以发现，三簇数据点处于 Z 轴的不同空间位置点，聚类的实质是对 Z 轴进行分割，所以可通过对 Z 轴附权重而增大 Z 轴数据点所占权重，然后利用共享近邻谱聚类（SNN）进行聚类。

问题四第二小问要求对机器工件外部边缘轮廓中不同的直线和圆弧进行分类。根据直线和圆弧的特征，可将其划分为 3 类。但数据点存在明显的噪音、数据点缺失等问题，且多流形间距离较近，造成一般谱聚类出现误导。稀疏流形聚类与嵌入（Sparse Manifold Clustering and Embedding, SMCE）可以通过自动寻找数据点的最小邻域并附以合适的权重，来选择同一流形上的数据点，进而对距离较近的多个流形进行判断。因此，可以解决该多流形问题。

此外，Chameleon 聚类法先将数据点分成很小的簇，而后根据相近程度进行合并。因此，相对一般谱聚类而言，也可以相对较好的对多流形问题进行分类。

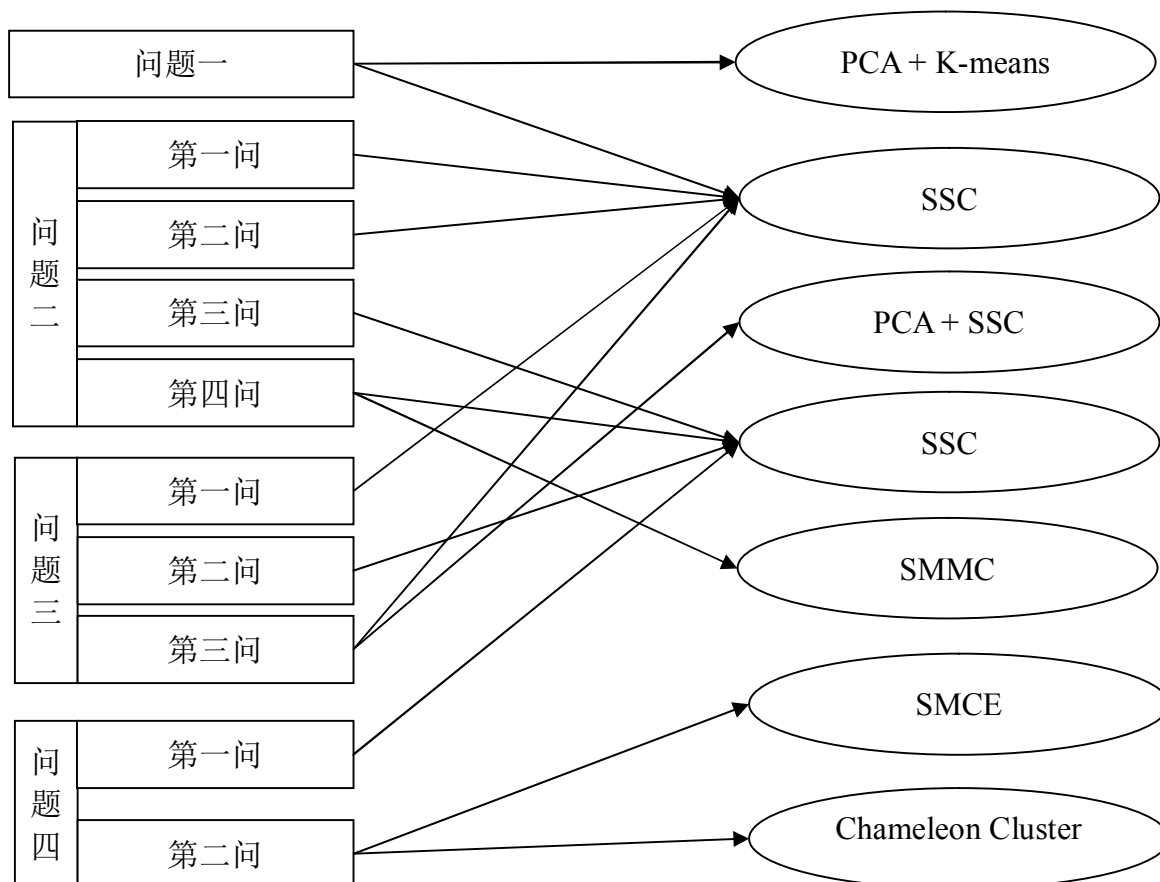


图 4-1 技术框架图

五 模型特点介绍

5.1 PCA+K-means 聚类模型

对于高维样本空间，首先采用 PCA 主成分分析方法降维，后用 K-means 聚类。

PCA（主成分分析法）是对高维样本空间引入随机变量，将多个变量通过线性变换选出较少重要变量的一种多元统计方法。其核心思想在将具有相关性的 P 个指标中，重新组合一组相互无关的综合指标。利用公式（5.1）：

$$F_p = a_{1i} * Z_{x1} + a_{2i} * Z_{x2} + \dots + a_{pi} * Z_{xp} \quad (5.1)$$

其中 $a_{1i}, a_{2i} \dots a_{pi} (i = 1, 2, 3)$ 为 X 的协方差阵 Σ 的特征值所对应的特征向量，

$Z_{x1}, Z_{x2}, \dots, Z_{xp}$ ，是原始变量经过标准化处理的价值，因为在实际应用中，往往存在指标的量纲不同，所以在计算之前须先消除量纲的影响

Kmeans 算法接受参数 k；然后将事先输入的 n 个数据对象划分为 k 个聚类以便使得所获得的聚类满足：同一聚类中的对象相似度较高；而不同聚类中的对象相似度较小。聚类相似度是利用各聚类中对象的均值所获得一个“中心对象”（引力中心）来进行计算的。

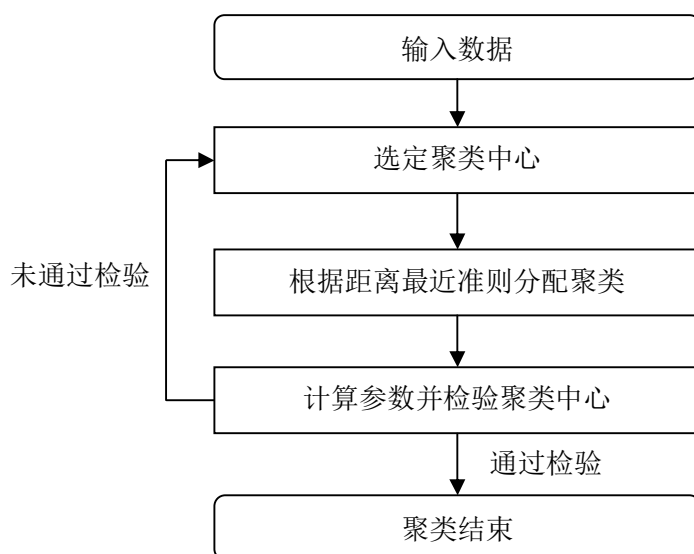


图 5-1 Kmeans 的流程图

5.2 谱聚类（Spectral Clustering）

建立在谱图理论基础上，与传统的聚类算法相比，它具有能在任意形状的样本空间上聚类且收敛于全局最优解的优点。其本质是将聚类问题转化为图的最优划分问题，将数据集中的每个对象看作是图的顶点 V，将顶点间的相似度量作为相应顶点连接边 E 的权值，这样就得到一个基于相似度的无向加权图 $G(V, E)$ ，于是聚类问题就可以转化为图的划分问题。基于图论的最优划分准则就是使划分成的子图内部相似度最大，子图之间的相似度最小，基于公式（5.2）构造相似矩阵 w_{ij} ，其中 $\|\cdot\|$ 为欧式距离

$$w_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) & i \neq j \\ 0 & i = j \end{cases} \quad (5.2)$$

后利用公式 (5.3) N-cut 进行剪边,

$$\text{Ncut}(X_1, \dots, X_k) \triangleq \frac{1}{2} \sum_{i=1}^k \frac{W(X_i, \bar{X}_i)}{\text{vol}(X_i)} \quad (5.3)$$

$X_1, \dots, X_k$ 是 $X(X_1 \cup \dots \cup X_k = X, X_i \cap X_j = \emptyset, i \neq j \text{ and } X_i \neq \emptyset, i = 1, \dots, k)$ 的一

部分, $W(A, B) \triangleq \sum_{X_i \in A, X_j \in B} \omega_{ij}$, $\text{vol}(A) \triangleq \sum_{X_i \in A, j \in \{1, \dots, N\}} \omega_{ij}$, $\bar{A}$ 是 A 的补集。为解决 NP-难问题, 将求拉普拉斯矩阵 L 的 K 个最小特征值及其对应的特征向量的问题, 转化为求矩阵 E 的 K 个最大的特征值及其对应的特征向量

最后, 利用 K-means 或其它经典聚类算法对特征向量空间中的特征向量进行聚类。

在谱聚类基础上, 针对流形问题、混合子空间问题等具体问题, 前人提出大量的算法, 下述为本论文用到的几类算法核心内容简述。

5.2.1 基于共享近邻的谱聚类 (SNN)

该模型设与点 X_i 最近的前 kd 个点构成集合, $N(x, i)$ 与点 x_j 最近的前 kd 个点构成集合 $N(X_j)$, 则点 X_i 和点 X_j 的共享 kd 近邻 $SNN(X_i, X_j) = N(X_i) \cap N(X_j)$, 利用 SNN 表征局部密度, 将这一信息用于相似度度量, 在自适应高斯核函数的基础上的相似度度量——基于共享近邻的自适应高斯核函数 (5.4)。

$$S_N(X_i, X_j) = \begin{cases} \exp\left(-\frac{\|X_i - X_j\|^2}{\sigma_i \sigma_j (SNN(X_i, X_j) + 1)}\right) & i \neq j \\ 1 & i = j \end{cases} \quad (5.4)$$

其中, σ_i 和 σ_j 分别为点 X_i 和点 X_j 到各自第 P 个近邻的距离。

然后, 通过 N-Cut 方式剪边后用 kmeans 进行聚类的方式

5.2.2 稀疏子空间聚类 (SSC)

对分布于多个低维子空间的并产生的子空间分割问题该方法是 Elhamifa 等于 2009 年基于一维稀疏性提出的, 其子空间模型用公式 (5.5) 表示为^[8]:

$$\min_Z \|Z\|_1 \quad (5.5.1)$$

$$\text{s.t. } X = XZ, Z_{ii} = 0 \quad (5.5.2)$$

其中, Z 具有对角结构, 揭示了数据的子空间属性[稀疏子空间聚类综述]。该模型利用稀疏表示(SR)迫使每个数据仅用同一子空间中其他数据的线性组合来表示.在数据所属的子空间相互独立的情况下, 模型(4)的解 Z 具有块对角结构, 这种结构揭示了数据的子空间属性:块的个数代表子空间个数, 每个块的大小代表对应子空间的维数, 同一个块的数据属于同一子空间。

在实际应用中，数据往往受到各种噪声或者奇异样本的影响，这时，数据 X 表示为 $X=DZ+E$ ，其中 E 为噪声或者奇异样本，通常 D 取为数据 X 本身或者干净字典 D 。一般地，稀疏子空间聚类模型可以统一描述为如下 (5.6) 优化问题^[9]

$$\min_Z J(Z) = F(E) + \lambda R(Z) \quad (5.6.1)$$

$$\text{s.t. } Z \in C \quad (5.6.2)$$

式中， C 为表示系数矩阵 Z 的约束集合， $\lambda > 0$ 为正则化参数； $F(E)$ 称为数据项或保真项，刻画了数据的表示 DZ 与数据 X 之间的逼近程度，针对数据中噪声的不同分布， $F(E)$ 采用不同的矩阵范数来度量误差； $R(Z)$ 称为正则项或惩罚项。稀疏子空间聚类通过对表示系数矩阵 Z 采用不同的稀疏度量作为正则项，迫使 Z 具有理想结构。

模型求解共分两个步骤：首先，利用全局的稀疏最优化，在所有点中寻找其它几个在同一子空间中的点；其次，利用稀疏系数，在谱聚类框架中对数据进行分类。

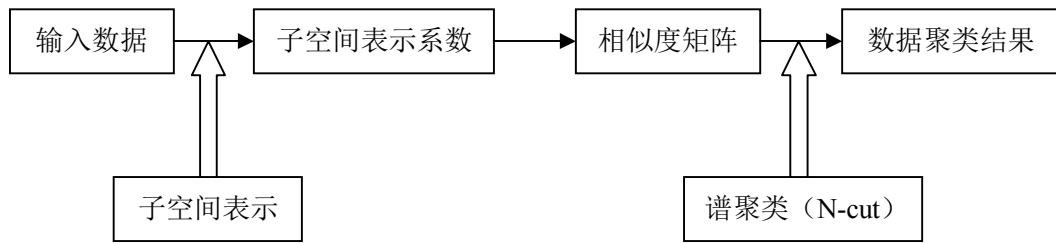


图 5-2 稀疏子空间聚类的基本的基本框架

5.2.3 多流形谱聚类模型 (SMMC)

由 YongWang 等提出，解决具有交叉子空间的流形聚类问题。首先，运行传统谱聚类方式，来揭示所有的链接组分，然后，对每一链接组分运行 K-manifolds 来进一步解开交叉聚类。其关键思想为以下两点：

(1) 构造关联矩阵：需要判断相距较远的两点是否属于同一流形，此时，不仅需要进行全局考虑，作者将目标放在同一趋于的两点判断是否足够近，并且有相似的切空间，藉此判断是否在同一流形上，因此需要构建关联矩阵，利用欧式距离， $q_{ij} =$

$q \|x_i - x_j\|$ (named localsimilarity) 及两点在切空间的相关性构成关联矩阵， $w_{ij} =$

$f(p_{ij}, q_{ij})$ 其中， f 为单调递增的融合函数。最终得到该关联值（公式 5.7），

$$w_{ij} = p_{ij} q_{ij} = \begin{cases} \left(\prod_{l=1}^d \cos(\theta_l) \right)^\circ, & \text{if } x_i \in Knn(x_j) \\ & \text{or } x_j \in Knn(x_i) \\ 0 & \text{otherwise} \end{cases} \quad (5.7)$$

其中， $Knn(x)$ 为 K-nearest neighbors of x 。

(2) 切空间的定义，通过全局非线性流形可以近似的被一系列局部线性流形所近似^[10,11]，并通过主成分来过滤交叉线性流形，从而将交叉点成功分开。

求解过程围绕：

参数选取原则： $Knn(x)$ ，越多，准确度越高； K 取值适中，太小的化会形成许多不连续的自聚类，太小局部的区分会丢失； P 值代表了不同流形间的可分性。

5.2.4 稀疏流形聚类与嵌入模型（SMCE）

由 Ehsan Elhamifar (2011) 等提出的一种多重非线性流形同步聚类 and 降维的模型，主要是能够通过解决稀疏优化问题，能自动选择同一流形下的近邻点，并近似的跨越一个低维仿射子空间。在最初给定邻近图同时，自动获取权重^[12]。

假定 $x_i \in M_1$ 包含最小球， $B_i \subset R^D$ ，此球包括了 M_1 中最邻近的 $d_i + 1$ 的最邻近 x_i ，通常，这一近邻也都在 M_1 中，公式 5.8。

$$\|\sum_{j \in N_i} c_{ij}(x_j - x_i)\|_2 \leq \epsilon \text{ and } \sum_{j \in N_i} c_{ij} = 1 \quad (5.8)$$

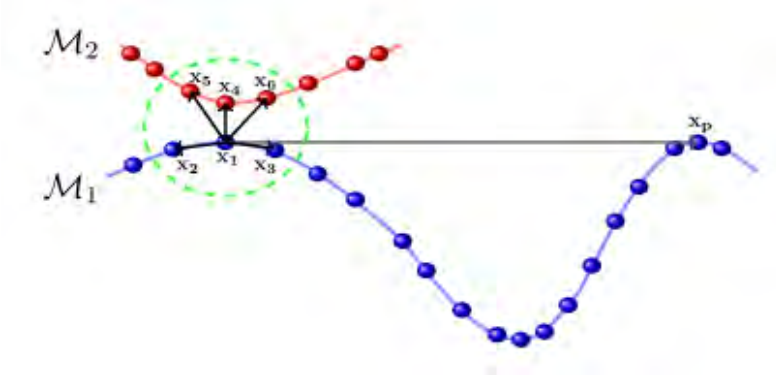


图 5-3 其中 $x_i \in M_1$ ，最近的临近点同时包括了 M_1, M_2 ，然而，通过 x_i 周围的一维子空间临近点判断其仅在 M_1 中（文献[12]）

其后，根据 d_l 维流形聚合 M_l ，最后通过 K-means 得到多流形聚类问题结果。取得不错的结果。

5.3 CHAMELEON 聚类

CHAMELEON 是一种两阶段聚类法。第一阶段把点分成很多小的簇；第二阶段根据相近程度合并这些小的簇。第一阶段采用 K 最邻近法，即把一个点和它最邻近的 K 个点连接起来。第二阶段计算任意两个簇的互连性 RI 和紧密性 RC，当两个指标都比较大时才合并这两个簇。

相对互连度 (5.9)：

$$RI(C_i, C_j) = \frac{2 * |EC(C_i, C_j)|}{|EC(C_i)| + |EC(C_j)|} \quad (5.10)$$

相对紧密度 (5.11)：

$$RC(C_i, C_j) = \frac{(|C_i| + |C_j|) |EC(C_i, C_j)|}{|C_j| |EC(C_i)| + |C_i| |EC(C_j)|} \quad (5.11)$$

$|C_i|$ 表示簇 i 内数据点的个数； $|EC(C_i)|$ 表示簇 i 内所有边的权重和； $EC(C_i, C_j)$ 表示跨越两个簇的所有边的权重和。

类似于谱聚类，Chameleon 算法也可以稳定地对数据进行聚类，不会对 k 的选择过分敏感。

六 问题求解

6.1 问题一求解

根据本题数据特征，提出分别采用 PCA + K-means 聚类算法和 SSC 谱聚类算法两种算法方案。其中，PCA + K-means 聚类算法采用常规方式求解。

通过我们将样本矩阵降至 20 维的矩阵，贡献度为 98.65，得到降维后的主成分向量，如下表所示。

表 6-1 PCA 降维主成分分量表

1	2	3	4	5	6	...	100
-0.02031	0.000551	0.080444	0.307225	0.232104	0.083194	...	0.081108
-0.03829	-0.13685	-0.12343	-0.14683	0.166359	0.015892	...	0.152605
...							
0.01573	0.0654	-0.0233	0.1554	-0.1235	-0.0235	...	0.0572

即，若降维前每组数据由 $[x_1, x_2, \dots, x_{100}]$ 表示，而降维后的数据由 $[y_1, y_2]$ 表示，则有

$$y_1 = -0.02031x_1 + 0.000551x_2 + \dots + 0.081108x_{100}$$

$$y_2 = -0.03829x_1 - 0.13685x_2 + \dots + 0.152605x_{100}$$

于是，200 个样本点降维后的数据如下表（6-2）所示

表 6-2 降维后数据

1	2	3	4	5	6	...	200
-1.03662	-0.72365	-1.04396	-1.34421	-0.90859	-1.25715	...	-1.23512
-0.08344	-0.09354	-0.10959	0.056814	-0.09308	-0.28506	...	0.203761
...							
-1.63552	0.166229	-0.0637	-0.07775	-0.03878	0.026501	...	-2.52E17

通过 Matlab 对数据进行 K-means 聚类，我们将 200 个样本点分为两类（如下图 6-1 所示）。

SSC 谱聚类算法参数设置如表 6-3 所示。

表 6-3 问题一 SSC 谱聚类算法参数设置

参数	R	Affine	Alpha	Outlier	Ro	n
参数值	0	true	20	true	1	2

R-代表降维后的主成分个数，r=0 则表示不需要降维；

Affine-判断子集之间是否为关联矩阵，默认为 false；

Alpha-代表正则化参数；

Outlier-判断是否有异常值或噪音，默认为 false；

Ro-确定参数矩阵是否全部可用，若 Ro<1，代表取部分，默认为 1；

n-代表分类数目。

根据题目要求，将样本聚类为 2 类，以 1 和 2 作为类别标签，两种聚类算法的分类结果如图 6-1 所示。

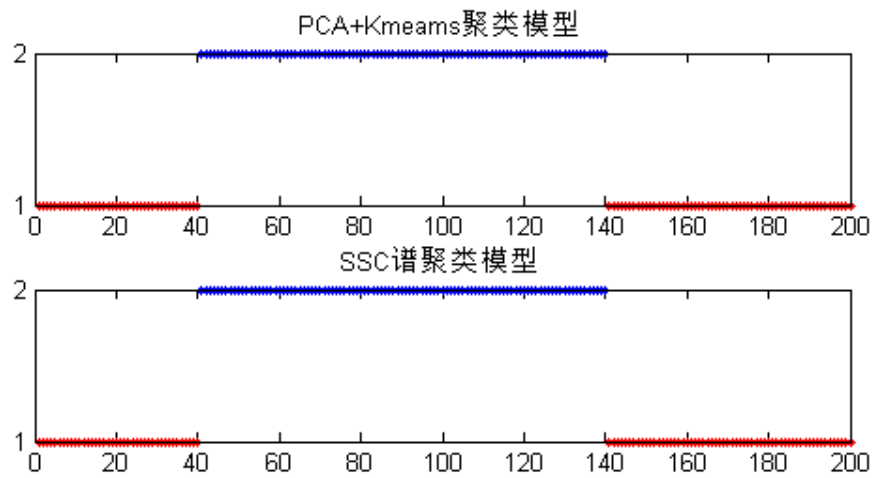


图 6-1 PCA + K-means 聚类模型与 SSC 谱聚类模型分类对比图

如图 6-1 所示，基于 PCA + K-means 聚类模型与 SSC 谱聚类模型对问题一的样本分类结果一致，样本的分类标签表格如表 6-4 所示。

表 6-4 问题一分类标签

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
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	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
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

6.2 问题二求解

6.2.1 图 a 求解

分析问题二图中图 a 的图形特点，采用 SSC 谱聚类算法（参数设置见表 6-5），分别用红色点与蓝色点标记样本点分类结果，其聚类的结果见图 6-2。

表 6-5 问题二图 a 的 SSC 谱聚类算法参数设置

参数	R	Affine	Alpha	Outlier	Ro	n
参数值	0	true	100	false	1	2

R-代表降维后的主成分个数，r=0 则表示不需要降维；
Affine-判断子集之间是否为关联矩阵，默认为 false；
Alpha-代表正则化参数；
Outlier-判断是否有异常值或噪音，默认为 false；
Ro-确定参数矩阵是否全部可用，若 Ro<1，代表取部分，默认为 1；
n-代表分类数目。

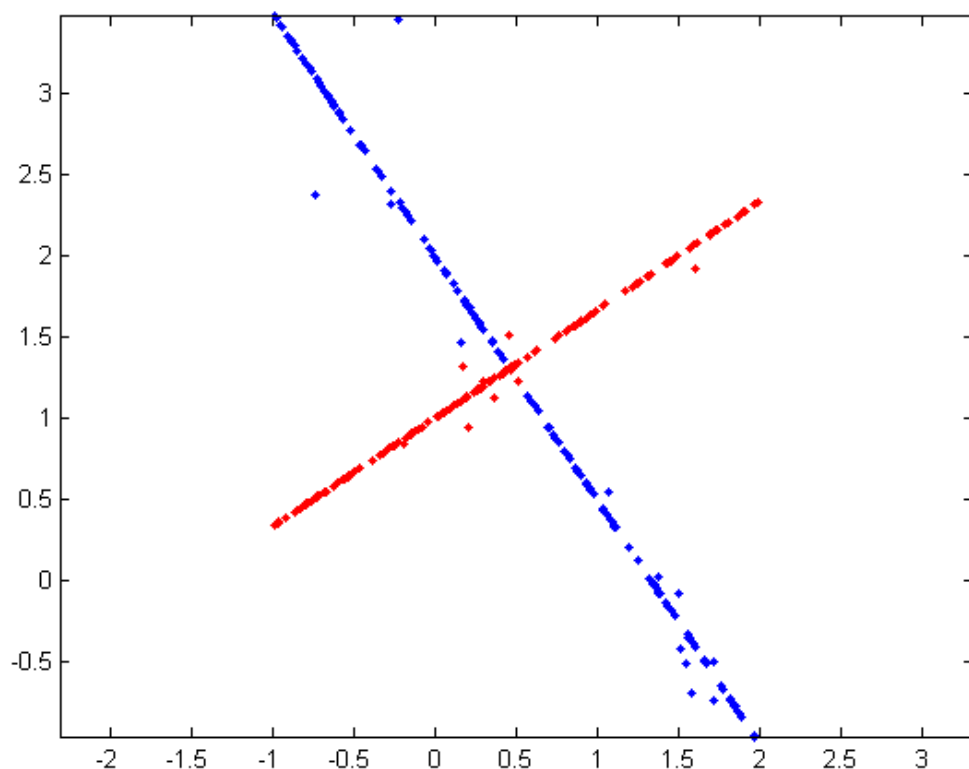


图 6-2 问题二图 a 聚类结果

如图 6-2 所示，SSC 谱聚类算法成功地将图 a 中的两条相交线段通过谱聚类进行了分类。其误差参数为 $\text{err1}=0.0002$ ， $\text{err2}=0.0194$ ， $\text{err3}=0.0001$ （其中， err1 代表两个系数矩阵元素间的最大误差（C-Z）； err2 代表线性系统的列之间残值的最大欧式距离残差 $Y-YZ$ ； err3 代表 Z 系数矩阵的最大误差）。

6.2.2 图 b 求解

分析问题二图 b 的图形特点，采用 SSC 谱聚类算法（参数设置见表 6-6），分别用红色点、蓝色点和绿色点标记样本点分类结果，其聚类的结果见图 6-3。

表 6-6 问题二图 b 的 SSC 谱聚类算法参数设置

参数	R	Affine	Alpha	Outlier	Ro	n
参数值	0	true	20	false	1	3

R-代表降维后的主成分个数， $r=0$ 则表示不需要降维；

Affine-判断子集之间是否为关联矩阵，默认为 false；

Alpha-代表正则化参数；

Outlier-判断是否有异常值或噪音，默认为 false；

Ro-确定参数矩阵是否全部可用，若 $Ro < 1$ ，代表取部分，默认为 1；

n-代表分类数目。

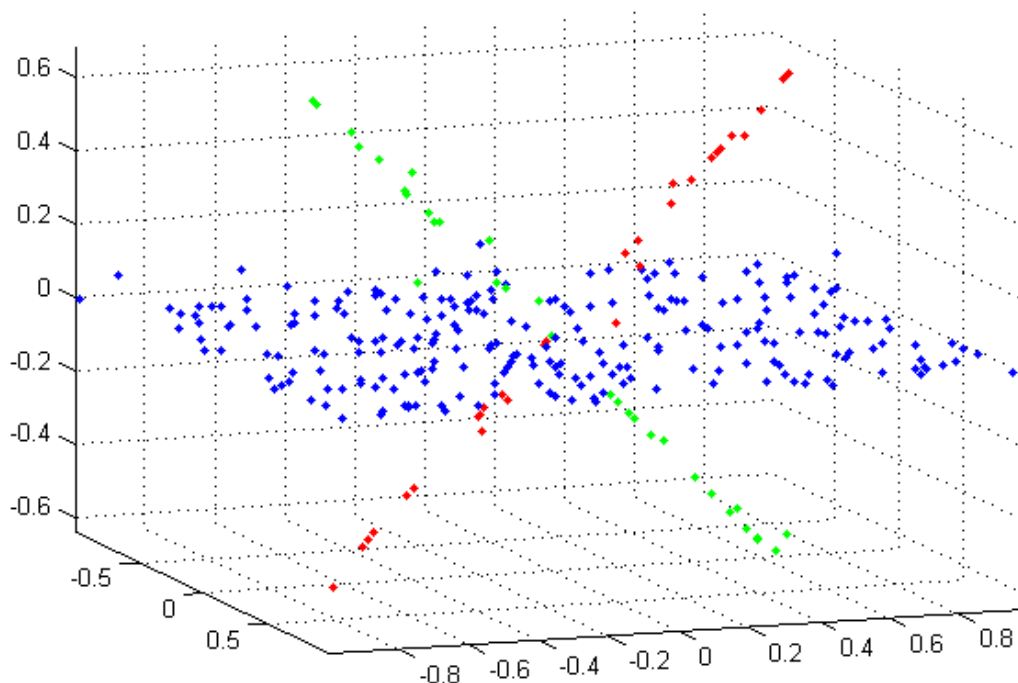


图 6-3 问题二图 b 聚类结果

如图 6-3 所示，SSC 谱聚类算法成功地将图 b 中的相交的两条线段和一个平面通过谱聚类进行了分类。其误差参数为 $\text{err1}=0.0009$ ， $\text{err2}=0.0549$ （其中， err1 代表两个系数矩阵元素间的最大误差（C-Z）； err2 代表线性系统的列之间残值的最大欧式距离残差 Y-YZ）。

6.2.3 图 c 求解

分析问题二图 c 的图形特点，选择 SNN 谱聚类算法（参数设置见表 6-7），分别用红色点与蓝色点标记样本点分类结果，其聚类的结果见图 6-4。

表 6-7 问题二图 c 的 SNN 谱聚类算法参数设置

参数	num_neighbors	block_size	sigma	num_clusters
参数值	20	5	10	2

num_neighbors-代表共享临近分析的临近点数量；

block_size-代表数据矩阵的块的大小；

sigma-用于相似度计算的参数，若 $\text{sigma}=0$ ，则自适应调整；

num_clusters -代表分类数目。

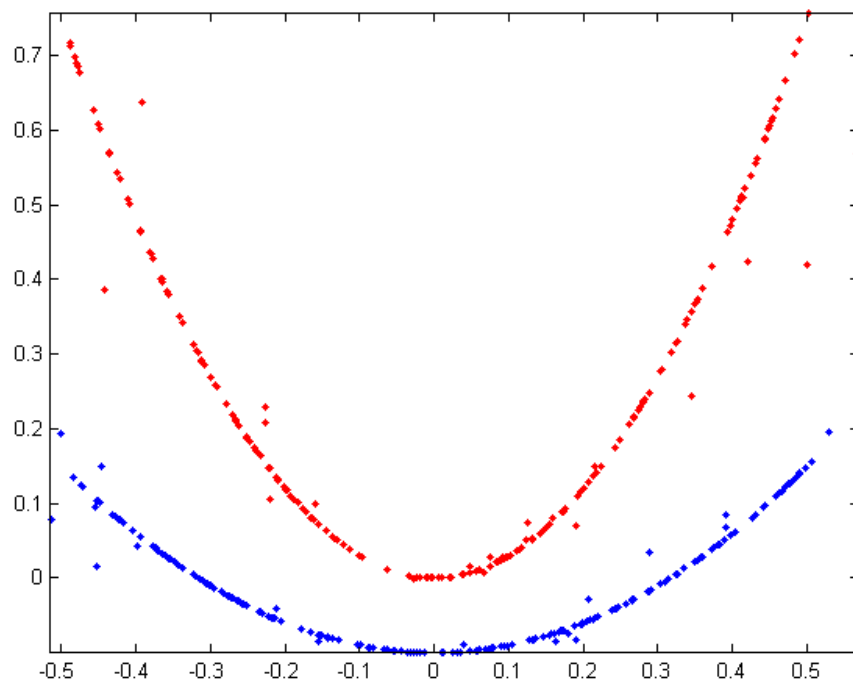


图 6-4 问题二图 c 聚类结果

如图 6-4 所示, SNN 谱聚类算法成功地将图 c 中的两条不相交的二次曲线进行了分类。

6.2.4 图 d 求解

分析问题二图 d 的图形特点, 选择 SNN 谱聚类算法和 SMMC 谱聚类算法进行求解。SNN 谱聚类算法和 SMMC 谱聚类算法的参数设置如表 6-8 和表 6-9 所示, 分别用红色点与蓝色点标记样本点分类结果, 两种算法的聚类的结果分别见图 6-5 和图 6-6。

表 6-8 问题二图 d 的 SNN 谱聚类算法参数设置

参数	num_neighbors	block_size	sigma	num_clusters
参数值	20	5	10	2

num_neighbors-代表共享临近分析的临近点数量;

block_size-代表数据矩阵的块的大小;

sigma-用于相似度计算的参数, 若 sigma=0, 则自适应调整;

num_clusters -代表分类数目。

通过对文献的阅读, 可知 SMMC 参数选取原则为: $Knn(x)$, 越多, 准确度越高; K 取值适中, 太小的化会形成许多不连续的自聚类, 太小局部的区分会丢失; P 值代表了不同流形间的可分性。鉴于此, 本文认为中心点个数 20-30 之间, 不宜过小, 否则 cut 不到交叉点。也不宜太大, 否则 cut 会超过交叉点。中心点个数是最关键的因素。kn 最邻近个数 20-30 左右。affinity (各子空间的关联度) 的 power 需要有一定的强度。体现出两个子空间之间具有交叉、关联。选取了 SMMC 的以下参数。

表 6-9 问题二图 d 的 SMMC 谱聚类算法参数设置

参数	nClusts	ppca_dim	ncentres	knn	power
参数值	2	1	30	24	10

nClusters-代表分类数目；
ppca_dim-代表主成分空间的维度；
ncentres-混合模型的中心点数目；
knn-邻域大小；
Power-关联强度。

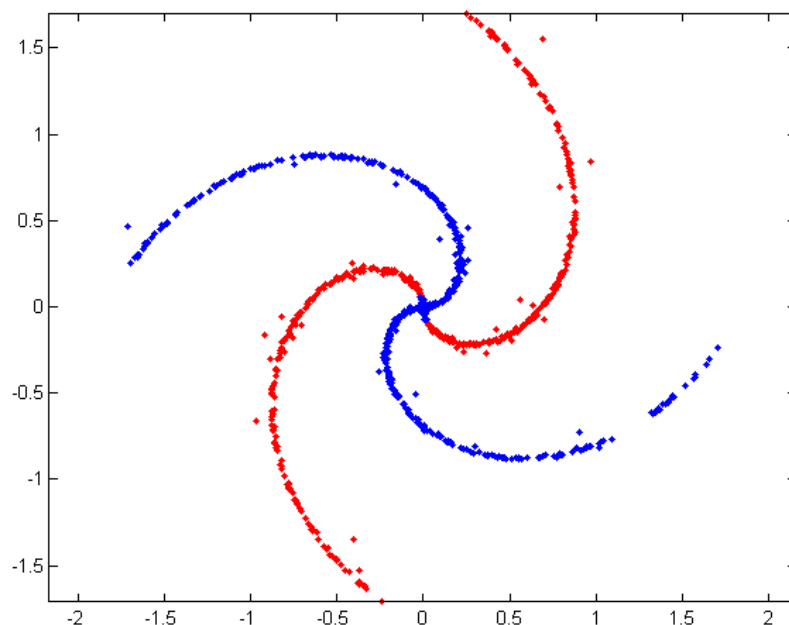


图 6-5 问题图 d 的 SSC 谱聚类结果

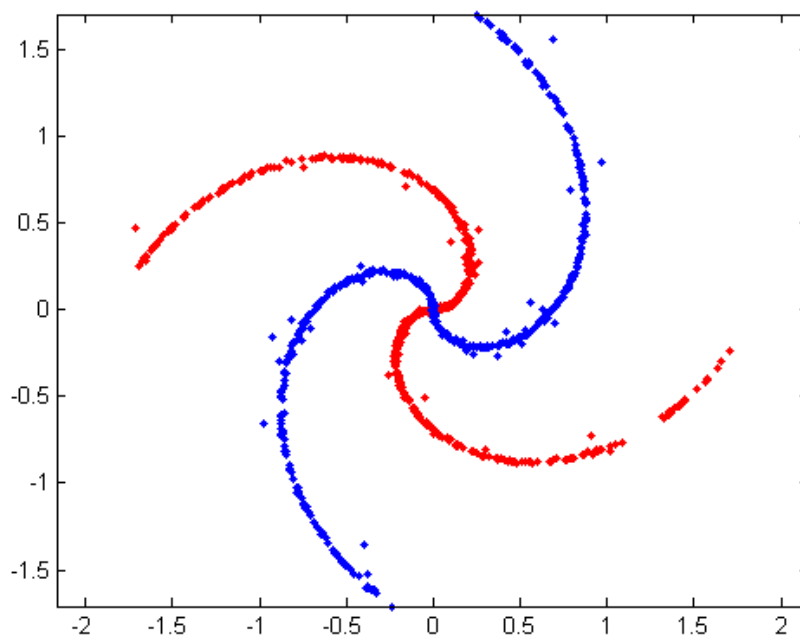


图 6-6 问题图 d 的 SMMC 谱聚类结果

如图 6-5 和图 6-6 所示，SNN 谱聚类算法和 SMMC 谱聚类算法都成功地将图 d 中的两条相交的螺旋线进行了分类，且通过图 6-5 和图 6-6 的对比可发现，SMMC 算法在两条螺旋线的交点处分类的效果比 SSC 算法的效果更加显著。

6.3 问题三求解

6.3.1 题 a 求解

分析问题三图 b 的图形特点, 采用 SSC 谱聚类算法 (参数设置见表 6-10), 分别用红色点和蓝色点标记样本点分类结果, 其聚类的结果见图 6-7。

表 6-10 问题三题 a 的 SSC 谱聚类算法参数设置

参数	R	Affine	Alpha	Outlier	Ro	n
参数值	0	false	800	false	1	2

R-代表降维后的主成分个数, $r=0$ 则表示不需要降维;

Affine-判断子集之间是否为关联矩阵, 默认为 false;

Alpha-代表正则化参数;

Outlier-判断是否有异常值或噪音, 默认为 false;

Ro-确定参数矩阵是否全部可用, 若 $Ro < 1$, 代表取部分, 默认为 1;

n-代表分类数目。

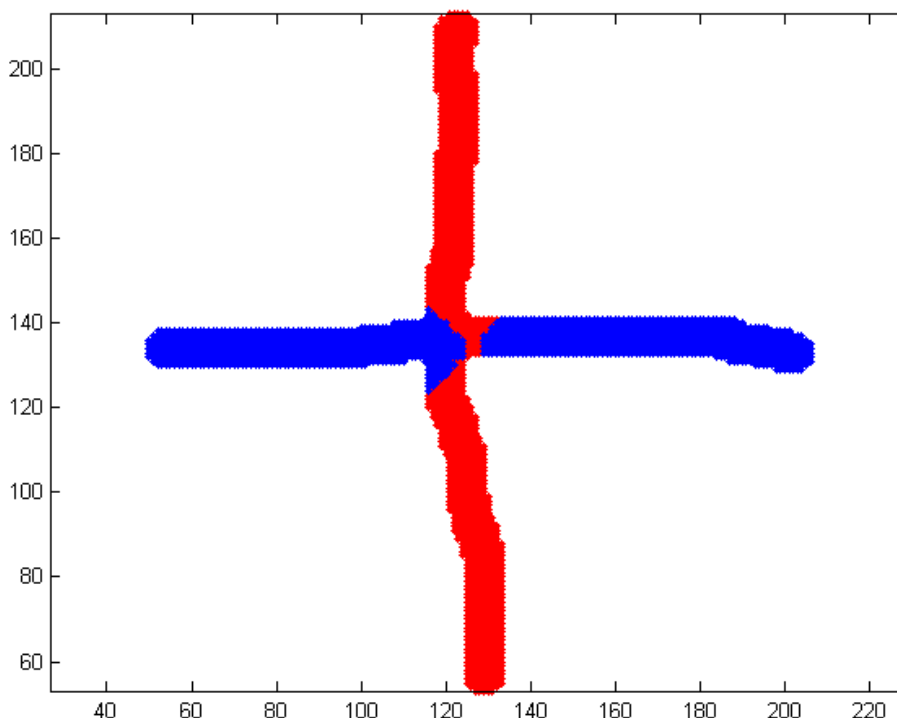


图 6-7 问题三题 a 聚类结果

如图 6-7 所示, SSC 谱聚类算法成功地将题 a 中的十字上的点进行了分类。其误差参数为 $err1=0.0002$, $err2=0.0015$ (其中, $err1$ 代表两个系数矩阵元素间的最大误差 ($C-Z$); $err2$ 代表线性系统的列之间残值的最大欧式距离残差 $Y-YZ$)。在中心处仍然出现了分类情况不好的情况, 可见仍然需要进一步调试。

6.3.2 题 b 求解

分析题 b 的数据可知, 样本点数据行数为 62 行, 分别对应轨迹点 31 帧数据的坐标信息, 令 Z 为特征点轨迹矩阵, X_{ij} , Y_{ij} 分别为第 j 列点在第 i 帧的坐标信息, 则

$$X_{i,j} = Z_{2i-1,j} \quad (6.1)$$

$$Y_{i,j} = Z_{2i,j} \quad (6.2)$$

定义 $V_{i,j}$ 为第 j 列特征点在第 i 帧与下一帧的偏移量

$$V_{i,j} = (X_{i+1,j} - X_{i,j}, Y_{i+1,j} - Y_{i,j}), i = 1, \dots, 30 \quad (6.3)$$

则基于运动轨迹的特征矩阵定义为 $A = V_{i,j}$ 。

选择 SNN 谱聚类算法（参数设置见表 6-11），以 1、2、3 作为类别标签，其聚类结果的标签表格见表 6-12。

表 6-11 问题三题 b 的 SNN 谱聚类算法参数设置

参数	num_neighbors	block_size	sigma	num_clusters
参数值	20	5	10	3

num_neighbors-代表共享临近分析的临近点数量；

block_size-代表数据矩阵的块的大小；

sigma-用于相似度计算的参数，若 sigma=0，则自适应调整；

num_clusters -代表分类数目。

表 6-12 问题三题 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	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	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	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	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

依据类别标签，分别用红色点、蓝色点和绿色点标记样本点分类结果，绘制出特征点数据的 31 帧聚类结果，如图 6-8 所示。

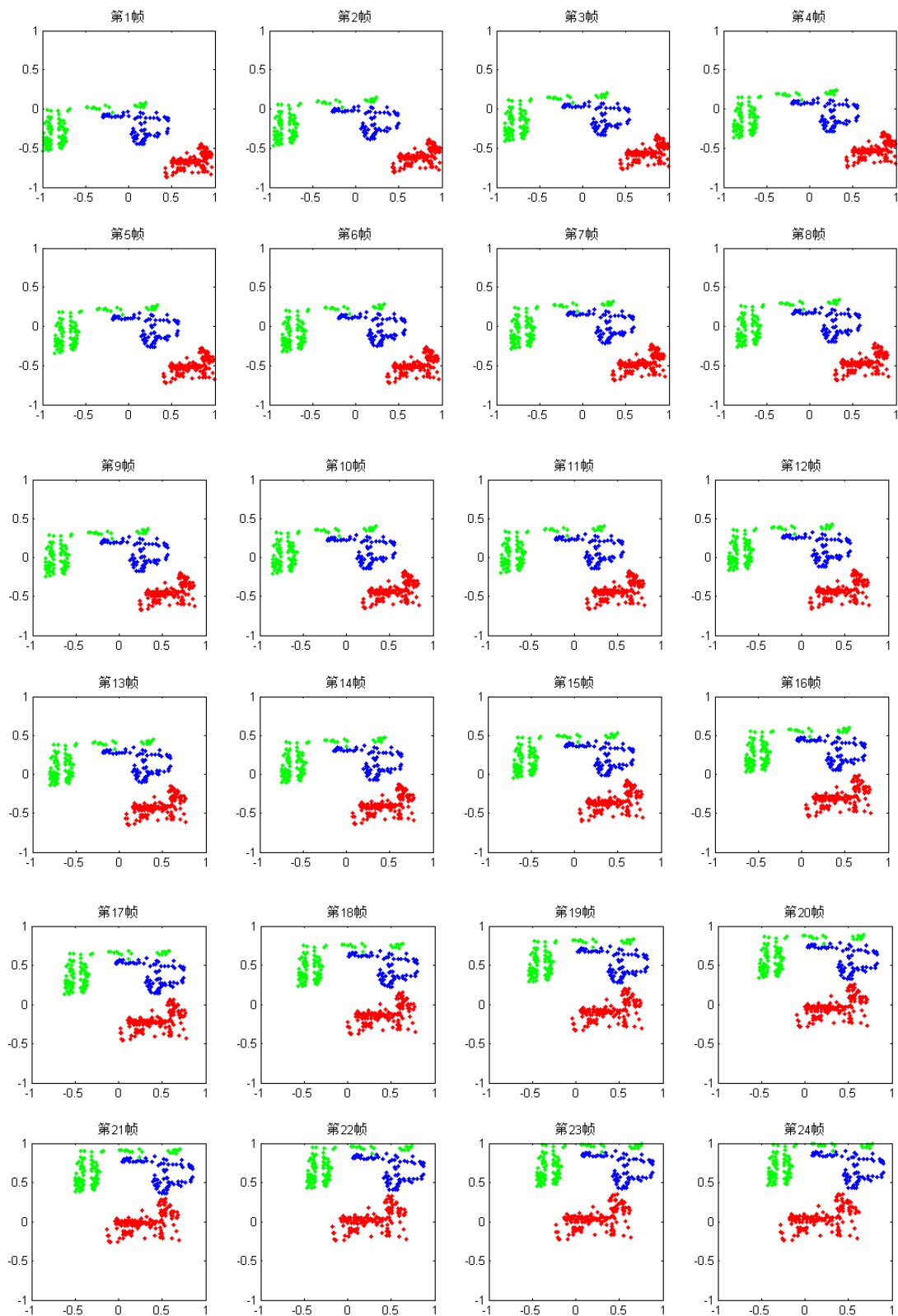


图 6-8 问题三题 b 的 31 帧聚类结果图

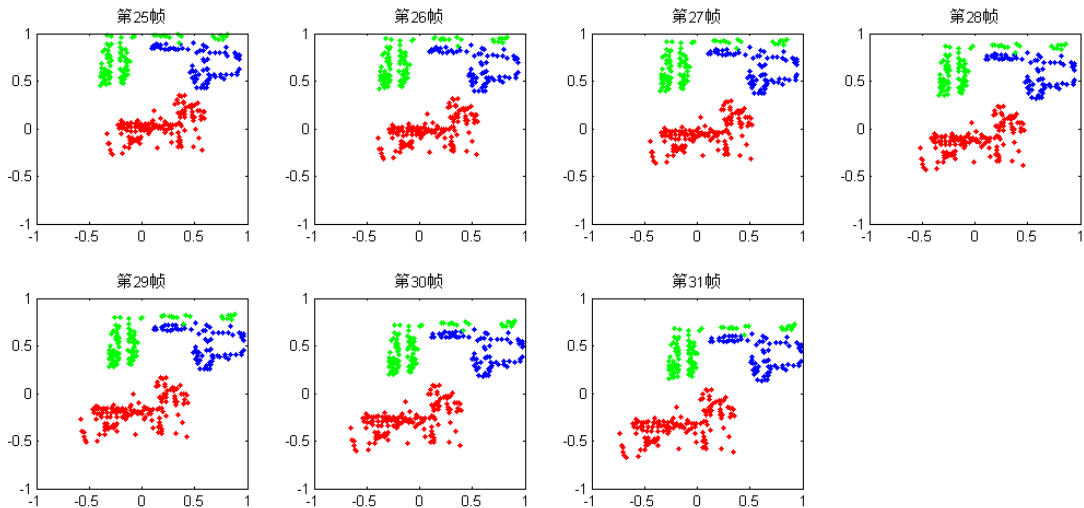


图 6-8 问题三题 b 的 31 帧聚类结果图（续）

6.3.3 题 c 求解

针对问题中数据的高维特征，采用 SSC 谱聚类算法（参数设置见表 6-13），以 1 和 2 作为类别标签，其聚类结果的标签表格如表 6-14 所示。

表 6-13 问题三题 c 的 SSC 谱聚类算法参数设置

参数	R	Affine	Alpha	Outlier	Ro	n
参数值	0	true	10	false	1	2

R-代表降维后的主成分个数， $r=0$ 则表示不需要降维；

Affine-判断子集之间是否为关联矩阵，默认为 false；

Alpha-代表正则化参数；

Outlier-判断是否有异常值或噪音，默认为 false；

Ro-确定参数矩阵是否全部可用，若 $Ro < 1$ ，代表取部分，默认为 1；

n-代表分类数目。

表 6-14 问题三题 c 分类标签（降维前后）

r=0	1	1	1	1	1	2	2	2	2	2	1	1	1	1	1	2	2	2	2	2
r=9	1	1	1	1	1	2	2	2	2	2	1	1	1	1	1	2	2	2	2	2

根据类别标签，将 20 幅人脸分成两类的结果如图 6-9 所示，其中用红色标记的序号为 1-5,11-15 的 10 张照片为类别 1，用蓝色标记的序号为 6-10,16-20 的 10 张照片为类别 2，根据图中的人脸可知，分类结果正确无误。

鉴于脸部特征维数达到 2016，故而采用 PCA 进行降维，提取面部特征，通过模型结果可知， $D=9$ 时可以贡献度已达到 99.67，故降维后再次运行 SSC，并将参数 r 设置为 9。对比二者误差（表 6-15）。

表 6-15 降维前后误差对比

误差	err1	err2	err3	ter
r=0	0.0013	0.0027,	0.0013	150
r=9	0.0016	0.0075	0.0037	150

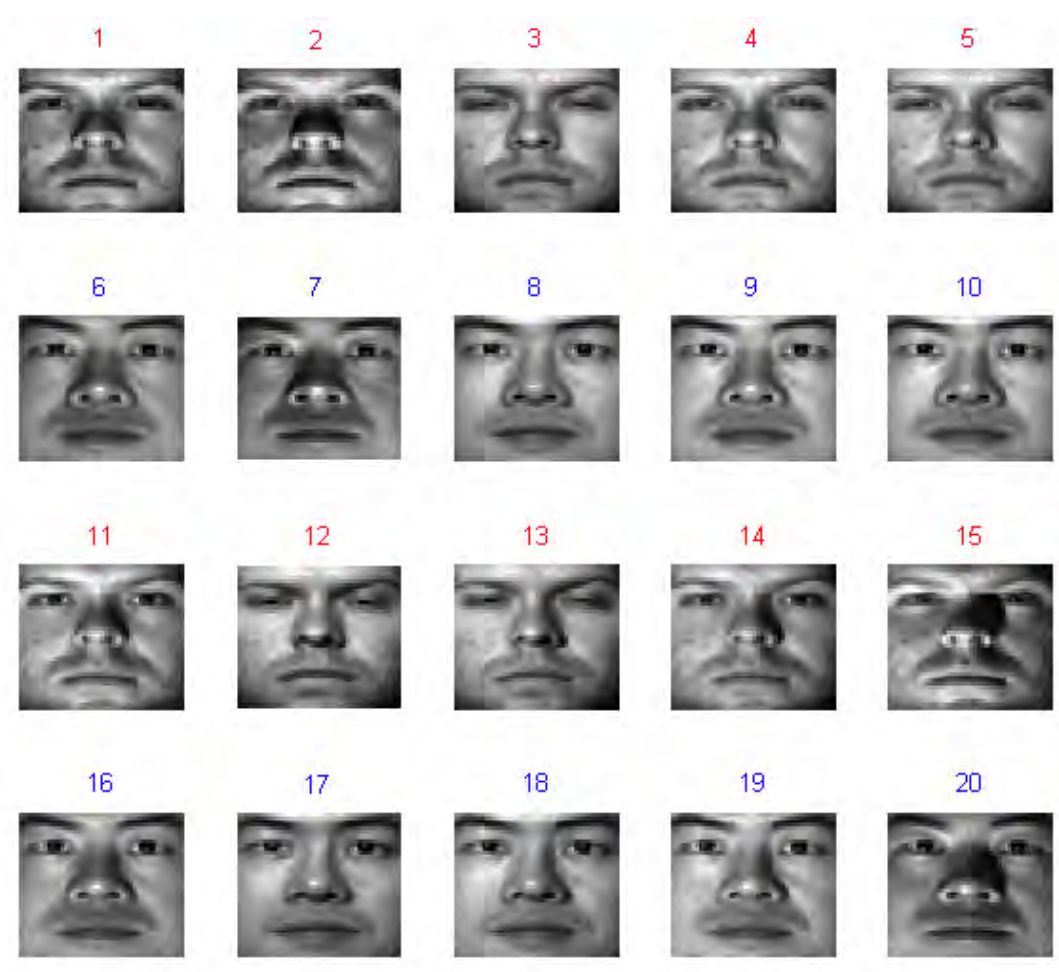


图 6-9 问题三题 c 的 20 张人脸聚类结果图

6.4 问题四求解

6.4.1 图 a 求解

分析问题中的特征点和要分成的上底面、下底面和侧面的几何特征，特征点在 Z 轴方向的取值是决定其在哪个面上的主要因素，选择 SNN 谱聚类算法(参数设置见表 6-16)，调整算法中 Z 轴的权重，以 1、2、3 作为类别标签，其聚类结果见图 6-10。

表 6-16 问题四图 a 的 SNN 谱聚类算法参数设置

参数	num_neighbors	block_size	sigma	num_clusters
参数值	20	5	10	3

num_neighbors-代表共享临近分析的临近点数量；

block_size-代表数据矩阵的块的大小；

sigma-用于相似度计算的参数，若 sigma=0，则自适应调整；

num_clusters -代表分类数目。

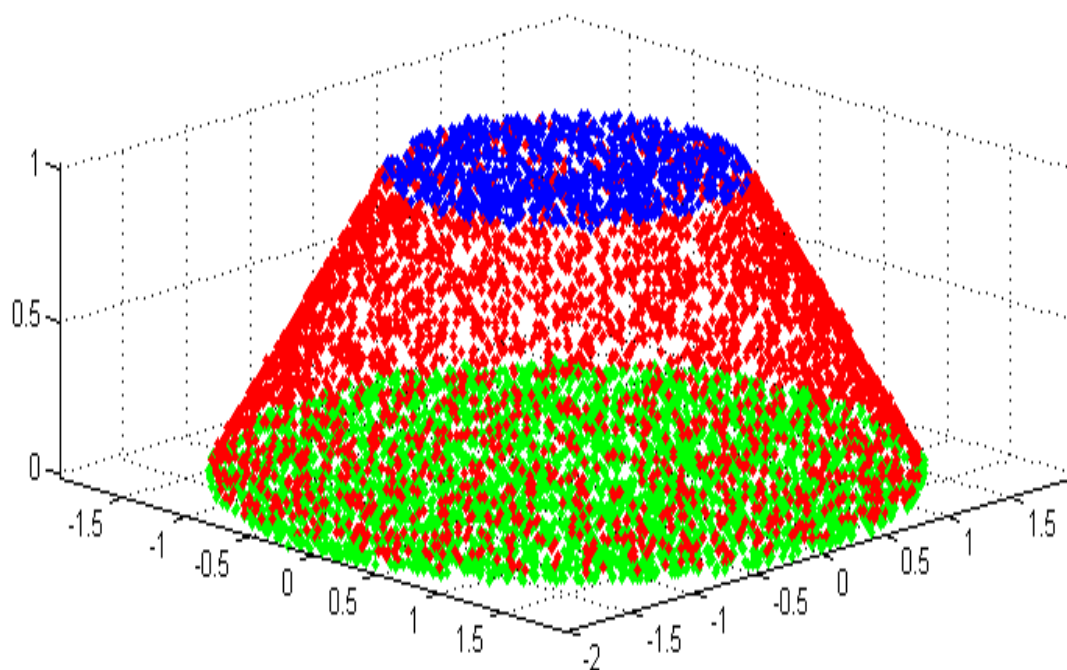


图 6-10 问题四图 a 的 SNN 算法分类结果

如图 6-10 所示，SNN 谱聚类算法成功地将图 a 中圆台上的点分成了上底面、下底面和侧面三类。

6.4.2 图 b 求解

分析问题四中图 b 的特征点的分布特征，选用 Chameleon 聚类算法和 SMCE 谱聚类算法对图 b 中的点进行聚类。

采用从下而上的层次聚类思想的 Chameleon 聚类算法，设置不同的聚类数目，对特征点聚类，其聚类结果如图 6-11 所示。

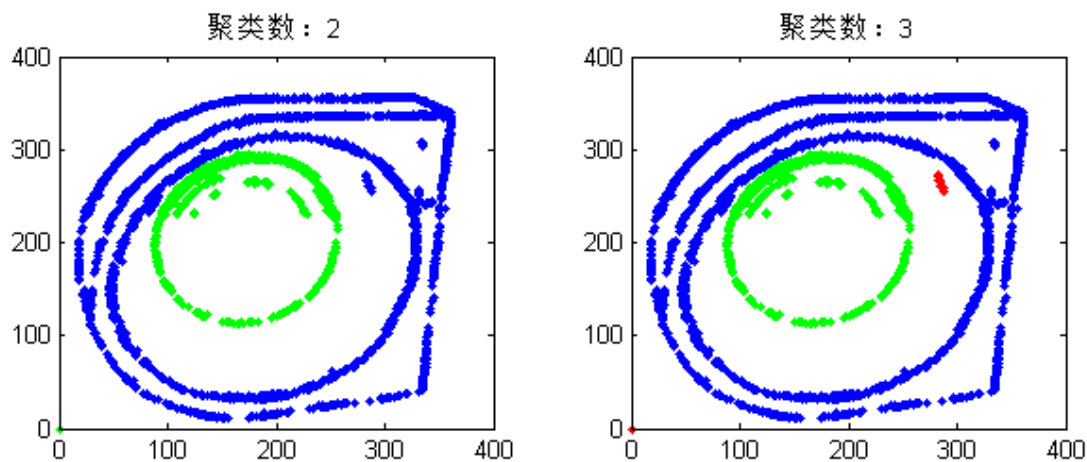


图 6-11 问题四图 b 的 Chameleon 算法聚类结果

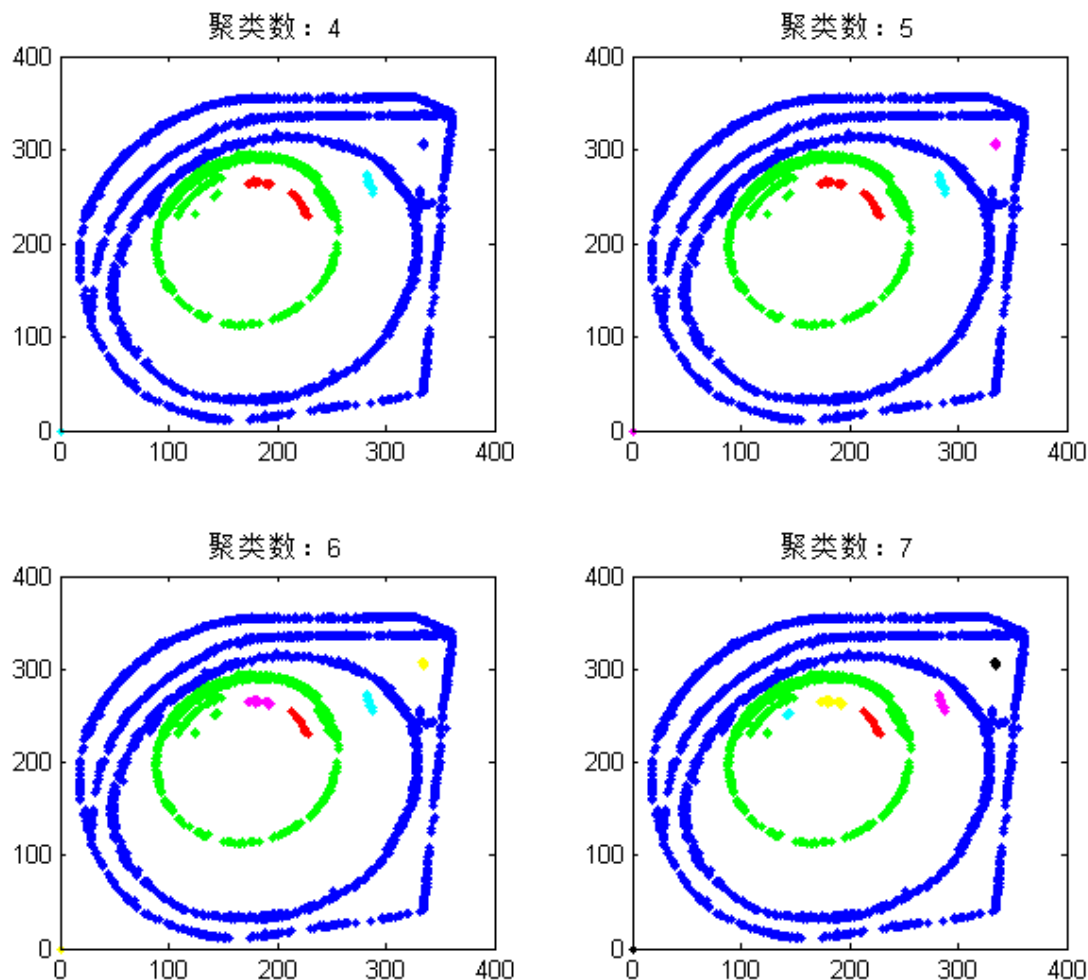


图 6-11 问题四图 b 的 Chameleon 算法聚类结果（续）

如图 6-11 所示，Chameleon 聚类算法能够对图 b 中的图形轮廓线进行聚类，将同一流形中的连续点进行聚类，然而对于较外的轮廓线其分类能力并不显著。

基于此，我们采用 EhsanElhamifar 于 2011 年提出的 SMCE 算法，该算法对多重非线性流形问题具有很好的解决效果。

采用 SMCE 谱聚类算法进行聚类(参数设置见表 6-17)，其聚类结果如图 6-12 所示。

表 6-17 问题四图 b 的 SMCE 谱聚类算法参数设置

参数	Lambda	KMax	Dim	Gtruth	Verbose	n
参数值	1	10	2	3	true	3

Lambda—平衡稀疏数据项和惩罚项的参数；

KMax—最大邻近点数量；

Dim—低维嵌入的维数；

Gtruth—流形关系向量的个数；

Verbose—确定是否返回优化信息；

n—代表分类数目；

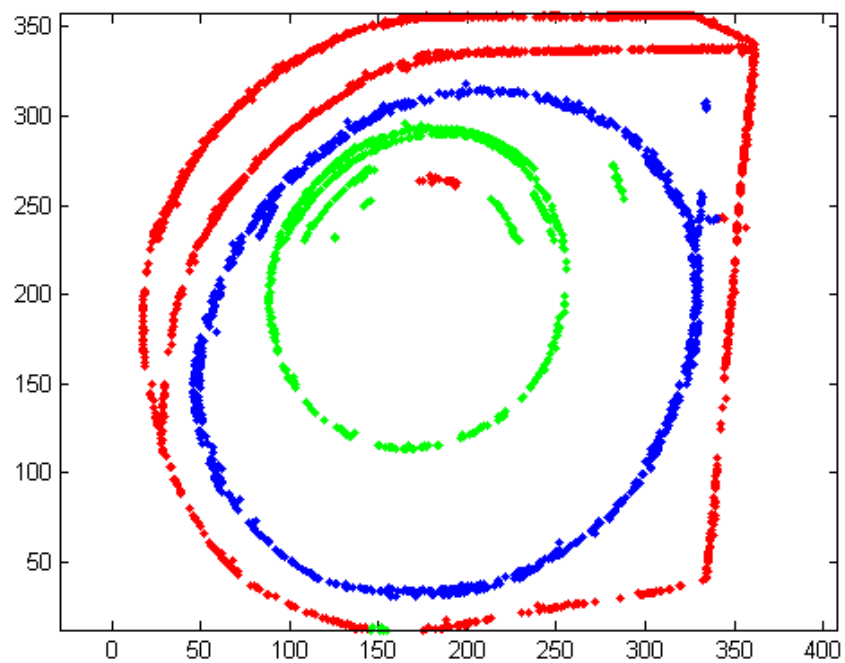


图 6-12 问题四图 b 的 SMCE 谱聚类结果

如图 6-12 所示，SMCE 谱聚类算法能够对图 b 中的图形轮廓线进行聚类，将同一流形中的连续点进行聚类，对于较外的轮廓线，其分类能力较 Chameleon 聚类算法更加显著。

7 模型评价与改进

谱聚类在图形数据的分类中具有非常有效的应用，本题重点从相似矩阵的构建和选择出发，对谱聚类算法性能进行的分析，不同相似矩阵的构建方法对谱聚类算法性能的影响性非常重要。但目前相似矩阵的构建没有统一的准则，且不同的相似矩阵对不同的子空间或流形特征适应性差别较大。

稀疏子空间谱聚类（SSC）通过数据点稀疏表达可以很好的解决数据点邻域大小的选择和子空间相交问题。由于其思想具有降维和消除数据稀疏性的特点，因此可以有效解决高维数据和具有稀缺性的数据分类问题。如问题一与传统的 PCA + K-means 聚类做对比，具有一致的效果。高维数据利用稀疏子空间谱聚类的方法，将分布于多个子空间的并从而进行分割，可以得到非常好的分类结果。但稀疏子空间不能很好的刻画非线性流形问题，对于同一弧线流形上的数据点不能很好的聚类。稀疏子空间谱聚类对噪声点的度量也存在一定的局限性，人为的聚类数目和阈值的设定对结果具有较大的敏感性，在解决曲线流形问题上具有较大的局限性。如利用该算法对问题二图（c）或图（d）无法使同一流形上的数据很好的聚为一簇。

共享近邻谱聚类（SNN）具有较好的识别能力和容错性，在不相交混合空间数据点的分类中具有较大的优越性。如在解决问题二图（c）两条不相交弧形流形数据时分类效果非常好。但共享近邻谱聚类算法对高斯核函数中的参数 σ 、聚类数目 k 等较为敏感，这些参数的人为设定直接关系到聚类结果的好坏。该方法处理由于不具有降维的效果，因而处理高维数据难度较大。此外，该方法在解决相交子空间数据分类上存在较大的局限性。如利用该算法对问题二图（a）或图（b）时，无法使同一线性子空间上的数据在交集附近很好的分类。

多流形谱聚类（SMMC）兼容了关联矩阵和共享近邻谱聚类的思想，对多种图形数据分类具有适用性。可以解决子空间相交的弧形流形数据分问题，利用向量余弦的思想可以对交集附近同一流形上的数据点进行降维后合理的判断。如解决问题二图（4）问题。但该方法对数据的聚类中心点多少、邻域大小、数据关联强度大小的人为设定非常敏感，需要对算法具有对结果影响很大。且数据输出结果具有一定的概率性，得到理想的分类结果需要人为选择，耗时较长。

稀疏流形聚类与嵌入（SMCE）多流形任务时具有较好的效果，可以很好的处理距离较近的不同流形数据点间的分类问题。如对问题四图（b）不同流形上的数据进行合理的分类。但该方法需要设定数据项与正则项之间的平衡参数，该参数对分类效果同样具有较大的影响。

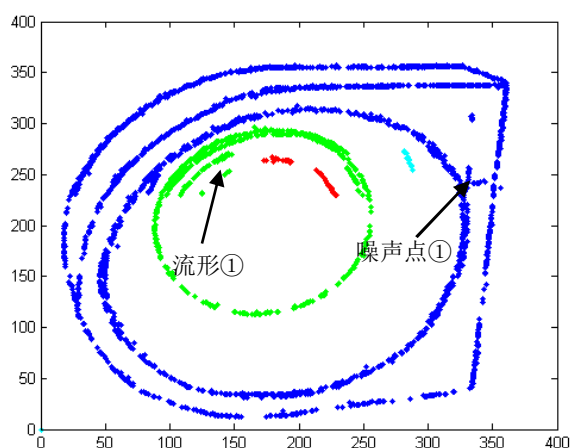


图 7-1 噪声点及数据点缺失对图形分类的影响

CHAMELEON 聚类也可以对多流形数据进行较好的分类，但其对噪声点、缺失点的优化效果较差，导致其对问题四图（b）不同流形上的数据分类出现误导。如图 7-1，噪声点①导致外围直线数据点与内部圈闭的弧线流形上的数据点不能分割。流形①由于数据点缺失，导致同一流形上的数据点无法建立同一簇内数据点欧氏距离最短，而表现为与临界流形上的数据点归为一类。

最后，本论文认为的算法改进思想在于对噪声和缺失点的判断与消除，首先，可以通过多次降维或是依据统计分布情况，筛除缺失值和噪声点，对复杂噪声进行误差建模，让稀疏子空间，有合适的的数据项；同时正则项的适当设定，让数据可以有先验信息，譬如 Lu^[13]提供的 Trace Lasso 正则项在 SSC 等模型中的应用，也是让数据具有更准确的信息。

参考文献

- [1] 百度百科, 聚类分析, http://baike.baidu.com/link?url=ZM1nhuaJOGwTijuN-j4R5Q4IE-VDKyJpCBb5vGJE_GVVzxyiZJaLJlvRXmnKPizmm-hzJKdQJ2ZQd0IR1QsgqK, 2015.09.19
- [2] 高琰, 谷士文, 唐琰等, 机器学习中谱聚类方法的研究, 计算机科学, 34(2): 201-203, 2007
- [3] 蔡晓妍, 戴冠中, 杨黎斌. 谱聚类算法综述, 计算机科学, 35(7): 14-18, 2008
- [4] 王卫卫, 李小平, 冯象初等, 稀疏子空间聚类综述, 自动化学报, 41(8): 1373-1384, 2015
- [5] 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
- [6] 李静伟, 基于共享近邻的自适应谱聚类算法, 大连理工大学, 2010
- [7] Y. Wang, Y. Jiang, Y. Wu, and Z. Zhou. Spectral clustering on multiple manifolds. IEEE Transactions on Neural Networks, 22(7):1149–1161, 2011
- [8] ParsonsL, HaqueE, LiuH, Subspace clustering for high dimensional data: are view, ACMSIGKDD Explorations Newsletter, 6(1): 90-105, 2004
- [9] Liu W. F., Pokharel P. P., Principe J. C.. Correntropy: properties and applications in non-Gaussian signal processing. IEEE Transactions on Signal Processing, 2007, 55(11):5286–5298
- [10]S. T. Roweis and L. K. Saul, Nonlinear dimensionlity reduction by locally linear embedding, Science, 290(5500): 2323-2326, 2000
- [11]L. K. Saul and S. T. Roweis, Think globally, fit locally: unsupervised learning of low dimensional manifolds, J Mach. Learn. Res. 4(2): 119-155, 2004
- [12]E. Elhamifar and R. Vidal. Sparse manifold clustering and embedding neural information processing systems(NIPS), 2011
- [13]Lu C Y, Min H, Zhao Z Q, Zhu L, Huang D S, Yan SC. Robust and efficient subspace segmentation via least squares regression. In: Proceedings of the the 2012 Computer Vision-European Conference on Computer Vision(ECCV). Florence, Italy: Springer Berlin Heidelberg,2012. 347–360

附 录

附件说明：文档附件中的代码为主程序代码，其余的 **function** 代码见压缩包附件。

附件 1：问题一主程序代码（不包含 **function** 代码）

```
clc;
clear;
load 1.mat;
figure
subplot(2,1,1)
if exist('Q1kmeans.xls','file')==2
    dos('del Q1kmeans.xls.xls');
end
x=data';
[coef,score,latent,tsquare]=princomp(x);%pca
latent=100*latent/sum(latent);
sum(latent(1:25)); %ans=98.6
x2=x(1:200,1:20);
cluster_labels=kmeans(x2,2);
ncol=size(data,2);
C1=[];
C2=[];
for i=1:ncol
    if mod(i,20)==0
        Q(ceil(i/20),20)=cluster_labels(i);
    else
        Q(ceil(i/20),mod(i,20))=cluster_labels(i);
    end
    if cluster_labels(i)==1
        C1=[C1;i,1];
    else
        C2=[C2;i,2];
    end
end
plot(C1(:,1),C1(:,2),'r.',C2(:,1),C2(:,2),'b.');
set(gca,'YTick',[ 1 2]);
title('PCA+Kmeans');
xlswrite('Q1kmeans.xls.xls',Q)

subplot(2,1,2)
if exist('Q1SSC.xls','file')==2
    dos('del Q1SSC.xls');
end
```

```

[Q1missrate,Q1Mat,Q1grps]=SSC(data,0,true,20,true,1,2);%
ncol=size(data,2)
C1=[];
C2=[];
for i=1:ncol
    if mod(i,20)==0
        Q(ceil(i/20),20)=cluster_labels(i);
    else
        Q(ceil(i/20),mod(i,20))=cluster_labels(i);
    end
    if cluster_labels(i)==1
        C1=[C1;i,1];
    else
        C2=[C2;i,2];
    end
end
plot(C1(:,1),C1(:,2),'r.',C2(:,1),C2(:,2),'b. ');
xlswrite('Q1SSC.xls',Q)
set(gca,'YTick',[ 1 2]);
title('SSCÆ×¼ÛÀÄÆÐí');

```

附件 2： 问题二主程序代码（不包含 function 代码）

A

```

clc;
clear;
load 2a.mat;
[Q2Amissrate,Q2ACMat,Q2Agrps]=SSC(data,0,true,100,false,1,2);%
ncol=size(data,2);
C1=[];
C2=[];
for i=1:ncol
    if Q2Agrps(i)==1
        C1=[C1;data(1,i),data(2,i)];
    elseif Q2Agrps(i)==2
        C2=[C2;data(1,i),data(2,i)];
    end
end
figure
plot(C1(:,1),C1(:,2),'r.',C2(:,1),C2(:,2),'b. ');
axis equal

```

B

```

clc;
clear;
load 2b.mat;
[Q2Bmissrate,Q2BCMat,Q2Bgrps]=SSC(data,0,false,20,false,1,3);%
ncol=size(data,2);
C1=[];
C2=[];
C3=[];
for i=1:ncol
    if Q2Bgrps(i)==1
        C1=[C1;data(1,i),data(2,i),data(3,i)];
    elseif Q2Bgrps(i)==2
        C2=[C2;data(1,i),data(2,i),data(3,i)];
    elseif Q2Bgrps(i)==3
        C3=[C3;data(1,i),data(2,i),data(3,i)];
    end
end
figure
scatter3(C1(:,1),C1(:,2),C1(:,3),'r');
hold on;
scatter3(C2(:,1),C2(:,2),C2(:,3),'b');
hold on;
scatter3(C3(:,1),C3(:,2),C3(:,3),'g');
axis equal
az=70;
el=10;
view(az,el);

```

```

C
clc;
clear;
load 2c.mat;
num_neighbors = 20;
block_size = 5;
X=data';
gen_nn_distance(X, num_neighbors, block_size, 0); %
filename = [num2str(num_neighbors), '_NN_sym_distance'];
load(filename)
num_clusters = 2;
sigma = 10;
cluster_labels = sc(A, sigma, num_clusters);
ncol=size(data,2);
C1=[];
C2=[];

```

```

for i=1:ncol
    if cluster_labels(i)==1
        C1=[C1;data(1,i),data(2,i)];
    elseif cluster_labels(i)==2
        C2=[C2;data(1,i),data(2,i)];
    end
end
figure
plot(C1(:,1),C1(:,2),'r.',C2(:,1),C2(:,2), 'b. ');
axis equal

```

D-SSC

```

clc;
clear;
load 2d.mat;
num_neighbors = 20;
block_size = 5;
X=data';
gen_nn_distance(X, num_neighbors, block_size, 0); %
filename = [num2str(num_neighbors), '_NN_sym_distance'];
load(filename)
num_clusters = 4;
sigma = 10;
cluster_labels = sc(A, sigma, num_clusters);
figure
plot(X(cluster_labels == 1, 1), X(cluster_labels == 1, 2), 'r.', X(cluster_labels == 2, 1), X(cluster_labels == 2, 2),
'b.', X(cluster_labels == 3, 1), X(cluster_labels == 3, 2), 'g.', X(cluster_labels == 4, 1), X(cluster_labels == 4, 2),
'c.')
axis equal

```

D-SMMC

```

clear;
load('2d.mat');
X=data;
ppca_dim=1;
[D,N] = size(X);
knn = 24; %100 200 300 400 500
ncentres=30;
[cluster_labels,ppca_label,mse,time_mppca,time_smmc,time_sc,W] = smmc(X,2,ppca_dim,ncentres,knn,10);
X1=X';
figure
plot(X1(cluster_labels == 1, 1), X1(cluster_labels == 1, 2), 'r.', X1(cluster_labels == 2, 1), X1(cluster_labels ==

```

```
2, 2), 'b.', X(cluster_labels == 3, 1), X1(cluster_labels == 3, 2), 'r.', X1(cluster_labels == 4, 1), X1(cluster_labels == 4, 2), 'b.')
axis equal
```

附件 3：问题三主程序代码（不包含 function 代码）

A

```
clc;
clear;
load 3a.mat;
x=zscore(data');
[Q3Amissrate,Q3ACMat,Q3Agrps]=SSC(x',0,false,800,false,1,2);%
ncol=size(data,2);
C1=[];
C2=[];
for i=1:ncol
    if Q3Agrps(i)==1
        C1=[C1;data(1,i),data(2,i)];
    elseif Q3Agrps(i)==2
        C2=[C2;data(1,i),data(2,i)];
    end
end
figure
plot(C1(:,1),C1(:,2),'r.',C2(:,1),C2(:,2),'b.');
axis equal
```

B

```
clc;
clear;
if exist('Q3B.xls','file')==2
    dos('del Q3B.xls');
end
load 3b.mat;
X=[];
for i=1:297
    sum1=0;
    sum2=0;
    for j=0:29
        sum1=sum1+data(2*j+1,i)-data(2*j+3,i);
        sum2=sum2+data(2*j+2,i)-data(2*j+4,i);
    end
    sum1=sum1/30;
```

```

        sum2=sum2/30;
        X=[X;sum1,sum2];
    end
    num_neighbors = 20;
    block_size = 5;
    gen_nn_distance(X, num_neighbors, block_size, 0); %
    filename = [num2str(num_neighbors), '_NN_sym_distance'];
    load(filename)
    num_clusters = 3;
    sigma = 10;
    cluster_labels = sc(A, sigma, num_clusters); % Æ×¼ÛÀà
    ncol=size(data,2);
    nrow=size(data,1);
figure
for j=1:nrow/2
    C1=[];
    C2=[];
    C3=[];
    for i=1:ncol
        if cluster_labels(i)==1
            C1=[C1;data(2*j-1,i),data(2*j,i)];
        elseif cluster_labels(i)==2
            C2=[C2;data(2*j-1,i),data(2*j,i)];
        elseif cluster_labels(i)==3
            C3=[C3;data(2*j-1,i),data(2*j,i)];
        end

        end
        subplot(5,7,j)
        plot(C1(:,1),C1(:,2),'r',C2(:,1),C2(:,2), 'b',C3(:,1),C3(:,2), 'g');
        set(gca,'YTick',[-1 -0.5 0 0.5 1 ],'XTick',[-1 -0.5 0 0.5 1]);
        axis([-1 1 -1 1 ]);
        title(['µÛ',num2str(j),'Öï'],'color','k');
        %axis equal;
    end
    for i=1:ncol
        if mod(i,20)==0
            Q(ceil(i/20),20)=cluster_labels(i);
        else
            Q(ceil(i/20),mod(i,20))=cluster_labels(i);
        end
    end
end
xlswrite('Q3B.xls',Q)

```

C

```
clc;
clear;
if exist('Q3C.xls','file')==2
    dos('del Q3C.xls');
end
load 3c.mat;
[Q3Cmissrate,Q3CMat,Q3Cgrps]=SSC(data,0,true,10,false,1,2);%
ncol=size(data,2);
C1=[];
C2=[];
for i=1:ncol
    if Q3Cgrps(i)==1
        C1=[C1;data(1,i),data(2,i)];
    elseif Q3Cgrps(i)==2
        C2=[C2;data(1,i),data(2,i)];
    end
end
xlswrite('Q3C.xls',Q3Cgrps);
ncol=size(data,2)
figure
for i=1:ncol
    subplot(4,5,i)
    a=data(:,i);
    b=reshape(a,42,48);
    handle=imshow(b,[]);
    if Q3Cgrps(i)==1
        title(i,'color','r');
    elseif Q3Cgrps(i)==2
        title(i,'color','b');
    end
end
end
```

附件 4： 问题四主程序代码（不包含 function 代码）

A

```
clc;
clear;
load 4a.mat;
num_neighbors = 20;
block_size = 5;
ncol=size(data,2);
centerl=0;
```

```

center2=1;
ee1=0.02;
ee2=0.02;
X=data';
z=0;
for nn=1:ncol
    z=X(nn,3);
    if z<=center2-ee2 && z>=center1+ee1
        X(nn,3)=X(nn,3)+100;
    end
end
gen_nn_distance(X, num_neighbors, block_size, 0); %
filename = [num2str(num_neighbors), '_NN_sym_distance'];
load(filename)
num_clusters = 3;
sigma = 10;
cluster_labels = sc(A, sigma, num_clusters); %  $\mathbb{R}^{n \times 3}$ 

C1=[];
C2=[];
C3=[];
for i=1:ncol
    if cluster_labels(i)==1
        C1=[C1;data(1,i),data(2,i),data(3,i)];
    elseif cluster_labels(i)==2
        C2=[C2;data(1,i),data(2,i),data(3,i)];
    elseif cluster_labels(i)==3
        C3=[C3;data(1,i),data(2,i),data(3,i)];
    end
end
figure
scatter3(C1(:,1),C1(:,2),C1(:,3),'g')
hold on; scatter3(C2(:,1),C2(:,2),C2(:,3),'b')
hold on; scatter3(C3(:,1),C3(:,2),C3(:,3),'r')
axis equal
az=45;
el=10;
view(az,el)

```

B

```

clear;
clc;
load 4b.mat

```

```

ncol=size(data,2);
img=zeros(ncol,ncol*2);
figure
subplot(3,2,1)
x=zeros(1,ncol);
y=zeros(1,ncol);

for i=1:ncol
    x(1,i)=data(1,i)
    y(1,i)=data(2,i);
end
processed=1000000;
for i=1:ncol
    for j=i+1:ncol
        dist(i,j)=(x(1,i)-x(1,j))*(x(1,i)-x(1,j))+(y(1,i)-y(1,j))*(y(1,i)-y(1,j));
        dist(j,i)=processed;
    end
    dist(i,i)=processed;
end

for i=1:ncol
    category(1,i)=i;
end
num_category=ncol;
while (num_category>2)
    mindist=1000000;
    for i=1:ncol
        for j=i+1:ncol
            if (dist(i,j)<mindist)
                mindist=dist(i,j);
                minp1=i;
                minp2=j;
            end
        end
    end
    dist(minp1,minp2)=processed;
    dist(minp2,minp1)=processed;
    for i=1:ncol
        if (category(1,i) == minp2)
            category(1,i)=minp1;
        end
    end
    for i=1:ncol
        if (dist(minp2,i)<dist(minp1,i))

```

```

        dist(minp1,i)=dist(minp2,i);
    end
    dist(minp2,i)=processed;
    dist(i,minp2)=processed;
end
num_category=num_category-1;
end
for i=1:ncol;
    labeled(1,i)=0;
end
categories=0;
for i=1:ncol
    kind=category(1,i);
    if (labeled(1,kind)==0)
        categories=categories+1;
        for j=i:ncol
            if(category(1,j)==kind)
                category(1,j)=categories;
            end
        end
        labeled(1,categories)=1;
    end
end

x1=zeros(1,ncol);
y1=zeros(1,ncol);
x2=zeros(1,ncol);
y2=zeros(1,ncol);
for i=1:ncol
    if(category(1,i)==1)
        x1(1,i)=x(1,i);
        y1(1,i)=y(1,i);
    elseif (category(1,i)==2)
        x2(1,i)=x(1,i);
        y2(1,i)=y(1,i);
    end
end

plot(x1,y1,'b.',x2,y2,'g.');
title('3/4ÛÀÊý£²');

subplot(3,2,2)
x=zeros(1,ncol);
y=zeros(1,ncol);

```

```

for i=1:ncol
    x(1,i)=data(1,i)
    y(1,i)=data(2,i);
end

processed=1000000;
for i=1:ncol
    for j=i+1:ncol
        dist(i,j)=(x(1,i)-x(1,j))*(x(1,i)-x(1,j))+(y(1,i)-y(1,j))*(y(1,i)-y(1,j));
        dist(j,i)=processed;
    end
    dist(i,i)=processed;
end
for i=1:ncol
    category(1,i)=i;
end
num_category=ncol;
while (num_category>3)
    mindist=1000000;
    for i=1:ncol
        for j=i+1:ncol
            if (dist(i,j)<mindist)
                mindist=dist(i,j);
                minp1=i;
                minp2=j;
            end
        end
    end
    dist(minp1,minp2)=processed;
    dist(minp2,minp1)=processed;
    for i=1:ncol
        if (category(1,i) == minp2)
            category(1,i)=minp1;
        end
    end
    for i=1:ncol
        if (dist(minp2,i)<dist(minp1,i))
            dist(minp1,i)=dist(minp2,i);
        end
        dist(minp2,i)=processed;
        dist(i,minp2)=processed;
    end
    num_category=num_category-1;
end

```

```

for i=1:ncol;
    labeled(1,i)=0;
end
categories=0;
for i=1:ncol
    kind=category(1,i);
    if (labeled(1,kind)==0)
        categories=categories+1;
        for j=i:ncol
            if(category(1,j)==kind)
                category(1,j)=categories;
            end
        end
        labeled(1,categories)=1;
    end
end

x1=zeros(1,ncol);
y1=zeros(1,ncol);
x2=zeros(1,ncol);
y2=zeros(1,ncol);
x3=zeros(1,ncol);
y3=zeros(1,ncol);

for i=1:ncol
    if(category(1,i)==1)
        x1(1,i)=x(1,i);
        y1(1,i)=y(1,i);
    elseif (category(1,i)==2)
        x2(1,i)=x(1,i);
        y2(1,i)=y(1,i);
    elseif (category(1,i)==3)
        x3(1,i)=x(1,i);
        y3(1,i)=y(1,i);
    end
end

plot(x1,y1,'b.',x2,y2,'g.',x3,y3,'r.');
```

title('4ÛÀËÿ£³');
 subplot(3,2,3)
 x=zeros(1,ncol);
 y=zeros(1,ncol);

```

for i=1:ncol

```

```

        x(1,i)=data(1,i)
        y(1,i)=data(2,i);
    end
    processed=1000000;
    for i=1:ncol
        for j=i+1:ncol
            dist(i,j)=(x(1,i)-x(1,j))*(x(1,i)-x(1,j))+(y(1,i)-y(1,j))*(y(1,i)-y(1,j));
            dist(j,i)=processed;
        end
        dist(i,i)=processed;
    end

    for i=1:ncol
        category(1,i)=i;
    end
    num_category=ncol;
    while (num_category>4)
        mindist=1000000;
        for i=1:ncol
            for j=i+1:ncol
                if (dist(i,j)<mindist)
                    mindist=dist(i,j);
                    minp1=i;
                    minp2=j;
                end
            end
            dist(minp1,minp2)=processed;
            dist(minp2,minp1)=processed;
            for i=1:ncol
                if (category(1,i) == minp2)
                    category(1,i)=minp1;
                end
            end
            for i=1:ncol
                if (dist(minp2,i)<dist(minp1,i))
                    dist(minp1,i)=dist(minp2,i);
                end
                dist(minp2,i)=processed;
                dist(i,minp2)=processed;
            end
            num_category=num_category-1;
        end
    end
    for i=1:ncol;

```

```

        labeled(1,i)=0;
end
categories=0;
for i=1:ncol
    kind=category(1,i);
    if (labeled(1,kind)==0)
        categories=categories+1;
        for j=i:ncol
            if(category(1,j)==kind)
                category(1,j)=categories;
            end
        end
        labeled(1,categories)=1;
    end
end

x1=zeros(1,ncol);
y1=zeros(1,ncol);
x2=zeros(1,ncol);
y2=zeros(1,ncol);
x3=zeros(1,ncol);
y3=zeros(1,ncol);
x4=zeros(1,ncol);
y4=zeros(1,ncol);

for i=1:ncol
    if(category(1,i)==1)
        x1(1,i)=x(1,i);
        y1(1,i)=y(1,i);
    elseif (category(1,i)==2)
        x2(1,i)=x(1,i);
        y2(1,i)=y(1,i);
    elseif (category(1,i)==3)
        x3(1,i)=x(1,i);
        y3(1,i)=y(1,i);
    elseif (category(1,i)==4)
        x4(1,i)=x(1,i);
        y4(1,i)=y(1,i);
    end
end

plot(x1,y1,'b.',x2,y2,'g.',x3,y3,'r.',x4,y4,'c.');
```

title('3/4ÛÀËÿ°4');

```

subplot(3,2,4)
x=zeros(1,ncol);
y=zeros(1,ncol);

for i=1:ncol
    x(1,i)=data(1,i)
    y(1,i)=data(2,i);
end
processed=1000000;
for i=1:ncol
    for j=i+1:ncol
        dist(i,j)=(x(1,i)-x(1,j))*(x(1,i)-x(1,j))+(y(1,i)-y(1,j))*(y(1,i)-y(1,j));
        dist(j,i)=processed;
    end
    dist(i,i)=processed;
end

for i=1:ncol
    category(1,i)=i;
end
num_category=ncol;
while (num_category>5)
    mindist=1000000;
    for i=1:ncol
        for j=i+1:ncol
            if (dist(i,j)<mindist)
                mindist=dist(i,j);
                minp1=i;
                minp2=j;
            end
        end
        dist(minp1,minp2)=processed;
        dist(minp2,minp1)=processed;
        for i=1:ncol
            if (category(1,i) == minp2)
                category(1,i)=minp1;
            end
        end
        for i=1:ncol
            if (dist(minp2,i)<dist(minp1,i))
                dist(minp1,i)=dist(minp2,i);
            end
            dist(minp2,i)=processed;
        end
    end
end

```

```

        dist(i,minp2)=processed;
    end
    num_category=num_category-1;
end
for i=1:ncol;
    labeled(1,i)=0;
end
categories=0;
for i=1:ncol
    kind=category(1,i);
    if (labeled(1,kind)==0)
        categories=categories+1;
        for j=i:ncol
            if(category(1,j)==kind)
                category(1,j)=categories;
            end
        end
        labeled(1,categories)=1;
    end
end
x1=zeros(1,ncol);
y1=zeros(1,ncol);
x2=zeros(1,ncol);
y2=zeros(1,ncol);
x3=zeros(1,ncol);
y3=zeros(1,ncol);
x4=zeros(1,ncol);
y4=zeros(1,ncol);
x5=zeros(1,ncol);
y5=zeros(1,ncol);
for i=1:ncol
    if(category(1,i)==1)
        x1(1,i)=x(1,i);
        y1(1,i)=y(1,i);
    elseif (category(1,i)==2)
        x2(1,i)=x(1,i);
        y2(1,i)=y(1,i);
    elseif (category(1,i)==3)
        x3(1,i)=x(1,i);
        y3(1,i)=y(1,i);
    elseif (category(1,i)==4)
        x4(1,i)=x(1,i);
        y4(1,i)=y(1,i);
    elseif (category(1,i)==5)

```

```

        x5(1,i)=x(1,i);
        y5(1,i)=y(1,i);
    end
end
plot(x1,y1,'b.',x2,y2,'g.',x3,y3,'r.',x4,y4,'c.',x5,y5,'m.');
```

title('3D Scatter Plot');

```

subplot(3,2,5)
x=zeros(1,ncol);
y=zeros(1,ncol);

for i=1:ncol
    x(1,i)=data(1,i)
    y(1,i)=data(2,i);
end
processed=1000000;
for i=1:ncol
    for j=i+1:ncol
        dist(i,j)=(x(1,i)-x(1,j))*(x(1,i)-x(1,j))+(y(1,i)-y(1,j))*(y(1,i)-y(1,j));
        dist(j,i)=processed;
    end
    dist(i,i)=processed;
end

for i=1:ncol
    category(1,i)=i;
end
num_category=ncol;
while (num_category>6)
    mindist=1000000;
    for i=1:ncol
        for j=i+1:ncol
            if (dist(i,j)<mindist)
                mindist=dist(i,j);
                minp1=i;
                minp2=j;
            end
        end
    end
    dist(minp1,minp2)=processed;
    dist(minp2,minp1)=processed;
    for i=1:ncol
        if (category(1,i) == minp2)
            category(1,i)=minp1;
        end
    end
end

```

```

end
for i=1:ncol
    if (dist(minp2,i)<dist(minp1,i))
        dist(minp1,i)=dist(minp2,i);
    end
    dist(minp2,i)=processed;
    dist(i,minp2)=processed;
end
num_category=num_category-1;
end

for i=1:ncol;
    labeled(1,i)=0;
end
categories=0;
for i=1:ncol
    kind=category(1,i);
    if (labeled(1,kind)==0)
        categories=categories+1;
        for j=i:ncol
            if(category(1,j)==kind)
                category(1,j)=categories;
            end
        end
        labeled(1,categories)=1;
    end
end

x1=zeros(1,ncol);
y1=zeros(1,ncol);
x2=zeros(1,ncol);
y2=zeros(1,ncol);
x3=zeros(1,ncol);
y3=zeros(1,ncol);
x4=zeros(1,ncol);
y4=zeros(1,ncol);
x5=zeros(1,ncol);
y5=zeros(1,ncol);
x6=zeros(1,ncol);
y6=zeros(1,ncol);

for i=1:ncol
    if(category(1,i)==1)
        x1(1,i)=x(1,i);
        y1(1,i)=y(1,i);
    end
end

```

```

elseif (category(1,i)==2)
    x2(1,i)=x(1,i);
    y2(1,i)=y(1,i);
elseif (category(1,i)==3)
    x3(1,i)=x(1,i);
    y3(1,i)=y(1,i);
elseif (category(1,i)==4)
    x4(1,i)=x(1,i);
    y4(1,i)=y(1,i);
elseif (category(1,i)==5)
    x5(1,i)=x(1,i);
    y5(1,i)=y(1,i);
elseif (category(1,i)==6)
    x6(1,i)=x(1,i);
    y6(1,i)=y(1,i);
end
end
plot(x1,y1,'b.',x2,y2,'g.',x3,y3,'r.',x4,y4,'c.',x5,y5,'m.',x6,y6,'y.');
```

title('ÛÀÄË£°6');

```

subplot(3,2,6)
x=zeros(1,ncol);
y=zeros(1,ncol);

for i=1:ncol
    x(1,i)=data(1,i)
    y(1,i)=data(2,i);
end
processed=1000000;
for i=1:ncol
    for j=i+1:ncol
        dist(i,j)=(x(1,i)-x(1,j))*(x(1,i)-x(1,j))+(y(1,i)-y(1,j))*(y(1,i)-y(1,j));
        dist(j,i)=processed;
    end
    dist(i,i)=processed;
end

for i=1:ncol
    category(1,i)=i;
end
num_category=ncol;
while (num_category>7)
    mindist=1000000;
    for i=1:ncol
```

```

        for j=i+1:ncol
            if (dist(i,j)<mindist)
                mindist=dist(i,j);
                minp1=i;
                minp2=j;
            end
        end
    end
    dist(minp1,minp2)=processed;
    dist(minp2,minp1)=processed;
    for i=1:ncol
        if (category(1,i) == minp2)
            category(1,i)=minp1;
        end
    end
    for i=1:ncol
        if (dist(minp2,i)<dist(minp1,i))
            dist(minp1,i)=dist(minp2,i);
        end
        dist(minp2,i)=processed;
        dist(i,minp2)=processed;
    end
    num_category=num_category-1;
end
for i=1:ncol;
    labeled(1,i)=0;
end
categories=0;
for i=1:ncol
    kind=category(1,i);
    if (labeled(1,kind)==0)
        categories=categories+1;
        for j=i:ncol
            if(category(1,j)==kind)
                category(1,j)=categories;
            end
        end
        labeled(1,categories)=1;
    end
end
x1=zeros(1,ncol);
y1=zeros(1,ncol);
x2=zeros(1,ncol);
y2=zeros(1,ncol);

```

```

x3=zeros(1,ncol);
y3=zeros(1,ncol);
x4=zeros(1,ncol);
y4=zeros(1,ncol);
x5=zeros(1,ncol);
y5=zeros(1,ncol);
x6=zeros(1,ncol);
y6=zeros(1,ncol);
x7=zeros(1,ncol);
y7=zeros(1,ncol);

for i=1:ncol
    if(category(1,i)==1)
        x1(1,i)=x(1,i);
        y1(1,i)=y(1,i);
    elseif (category(1,i)==2)
        x2(1,i)=x(1,i);
        y2(1,i)=y(1,i);
    elseif (category(1,i)==3)
        x3(1,i)=x(1,i);
        y3(1,i)=y(1,i);
    elseif (category(1,i)==4)
        x4(1,i)=x(1,i);
        y4(1,i)=y(1,i);
    elseif (category(1,i)==5)
        x5(1,i)=x(1,i);
        y5(1,i)=y(1,i);
    elseif (category(1,i)==6)
        x6(1,i)=x(1,i);
        y6(1,i)=y(1,i);
    elseif (category(1,i)==7)
        x7(1,i)=x(1,i);
        y7(1,i)=y(1,i);
    end
end
plot(x1,y1,'b.',x2,y2,'g.',x3,y3,'r.',x4,y4,'c.',x5,y5,'m.',x6,y6,'y.',x7,y7,'k. ');
title('3/4ÛÀÀËÿ£°7');

```

B-SMCE

```

clc, clear;
load('4b.mat');
Y = data;
lambda =1; KMax =10; dim = 2;

```

```

n = 3;
gtruth = 3;
verbose = true;
[Yc,Yj,clusters,missrate] = smce(Y,lambda,KMax,dim,n,gtruth,verbose);
X1=Y';
figure
plot(X1(clusters == 1, 1), X1(clusters == 1, 2), 'r.', X1(clusters == 2, 1), X1(clusters == 2, 2), 'b.', X1(clusters ==
3, 1), X1(clusters == 3, 2), 'g.', X1(clusters == 4, 1), X1(clusters == 4, 2), 'c.')
axis equal

```