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：

在如今的信息爆炸时代，大数据分析已经成为了热门话题。本文所阐述的内容就是数据处理的一个重要方法——多流形结构分析。本文根据不同的子空间聚类问题，使用稀疏子空间聚类(Sparse Subspace Clustering, SSC)[7],多流形谱聚类(Spectral Multi-Manifold Clustering, SMMC)[8]及其改进算法，共同解决了所有问题。本文所做的主要创新工作和结论如下：

第一，基于所给问题，验证和讨论了 SSC 和 SMMC 算法的适用范围。正如文献[7]中所说，SSC 算法适用于多个线性子空间聚类，并不适用于非线性子空间的聚类。而 SMMC 算法的适用范围就相对较广，不仅适用于线性子空间聚类，而且对非线性子空间同样适用。SMMC 的缺点也同样明显：

（1） 参数过多，对最佳算数的搜寻较为困难；

（2）由于 SMMC 中引入了概率主成分分析，所以相同的参数偶尔会产生不同的结果；

（3）SMMC 同样不适用于复杂的混合流形子空间。

针对（2）中 SMMC 的不稳定结果，由于 SMMC 产生准确结果的概率较大，文中使用了投票的方法：多次运行 SMMC 程序，根据产生的结果投票，若某个点的第i个聚类标签(记为Label(i))的票数最多，则该点属于Label(i)。

第二，基于所给问题，讨论了局部线性嵌入(Locally Linear Embedding, LLE)[9]对于高维子空间聚类的指导性意义。文献[9]中结论表明 LLE 能够降低数据维度，而不改变其局部结构。这说明通过 LLE 降低维度之后，若数据呈现线性分布，则数据在高维空间也是近似线性的。通过 LLE 可以处理高维子空间聚类。对于

高维数据，如第一题，第三题的(b)和(c)，文中会给出两类结果：直接使用高维数据聚类的结果和使用 LLE 降维（降到 3 维）之后聚类的结果。若两类结果不一样，如第三题的(b)，则以高维数据聚类的结果为准。、

第三，本文改进了 SMMC 算法，使得 SMMC 算法能够很好地解决第四题中的两个混合子空间模型。改进的基本思想如下：

（1）对于多个（大于两个）子空间的划分，首先分为 2 个子空间，然后根据划分的结果改进关系图的亲和矩阵，随后划分为 3 个子空间，依此类推，逐步划分为要求的子空间个数，命名为逐步多流形谱聚类(Gradually Spectral Multi-Manifold Clustering, GSMMC)。此方法的最主要思想就是把数据分为两个子空间比直接分为多个子空间更容易更准确；

（2）探测直线：由于第四题的(b)图中，直线和圆弧组成方式相当复杂，而直线又靠近噪声点，所以本文首先用 RANSAC 方法[12]将图(b)中的那条较长的直线检测出来，并从数据集中剔除，剩余的模型更为简单。或者将结果作为第一次分类结果，在 GSMMC 中用来优化亲和矩阵，不过此种方法并没有获得比直接剔除更好的效果，所以有待更进一步地研究；另外，去除直线或线性子空间的影响，对所有的混合子空间模型的聚类都有重大意义。

（3）增加数据维度：为了更好地区分数据，可以给数据添加上新的一维或者两维；也可以在 GMSSC 中每迭代一次，根据划分结果增加维度。由于时间限制本文只是采用了一种较为粗糙和低效的增加维度的方式。若想要获得更一般，更有效的增维方法，需要进一步地研究。

使用（1），就能很好地解决第四题的（a），使用（1）、（2）、（3）能够对第四题的（b）做出非常好的聚类。

最后，基于本文所做的所有工作、结论和思想，希望对相关研究人员有所启发。

关键词：稀疏子空间聚类，逐步多流形谱聚类，直线探测，增加数据维度

目录

更多数学建模资料请关注微店店铺“数学建模学习交流”
https://k.weidian.com/RHO6PSpA

一、 问题重述

1.1 引言

一个人在不同光照下的人脸图像可以被一个低维子空间近似[1]，由此产生大量的数据降维方法被用来挖掘数据集的低维线性子空间结构，这类方法假设数据集采样于一个线性的欧氏空间[15]。但是，在实际问题中很多数据具备更加复杂的结构。

针对单一子空间结构假设的后续讨论主要是两个方面，首先是从线性到非线性的扩展，主要的代表性工作包括流形（流形是局部具有欧氏空间性质的空间，欧氏空间就是流形最简单的实例）学习等。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。

其次是流形或子空间从一个到多个的扩展，即假设数据集采样于多个欧氏空间的混合。

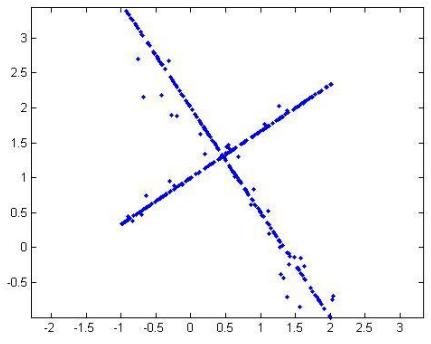
由于混合流形不全是子空间的情况，数据往往具有更复杂的结构，分析这种数据具有更大的挑战性。基于谱聚类的方法仍然是处理该类问题的流行方法。虽然这类数据本身无法使用相互表示的方式，但是数据的特征可相互线性表示且表示系数具有稀疏性或低秩性的特点。

1.2 问题的提出

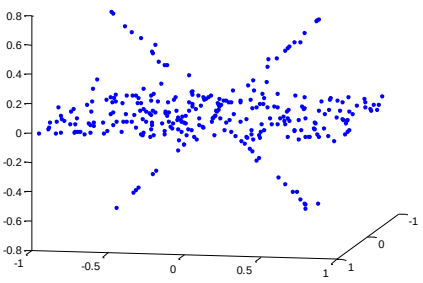
本几何结构分析问题中假设数据分布在多个维数不等的流形上，其特殊情况是数据分布在多个线性子空间上。

1.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。

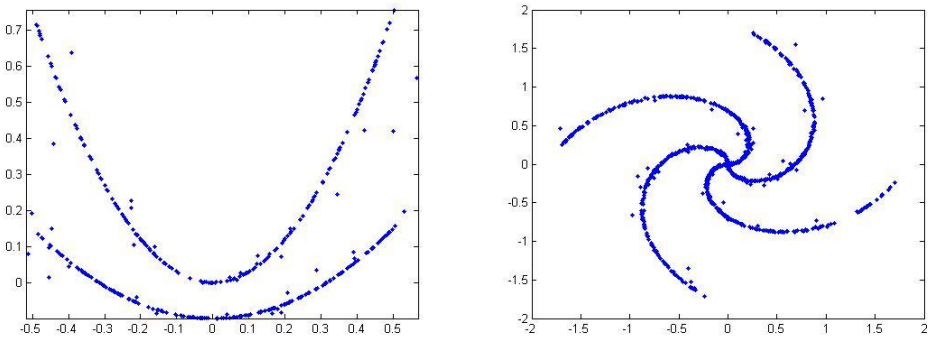
2.请处理附件二中四个低维空间中的子空间聚类问题和多流形聚类问题，如图 1.1 所示。图 1.1(a)为两条交点不在原点且互相垂直的两条直线，请将其分为两类；图 1.1 (b)为一个平面和两条直线，这是一个不满足独立子空间的关系的例子，请将其分为三类。图 1.1 (c)为两条不相交的二次曲线，请将其分为两类。图 1.1 (d) 为两条相交的螺旋线，请将其分为两类。



(a)



(b)



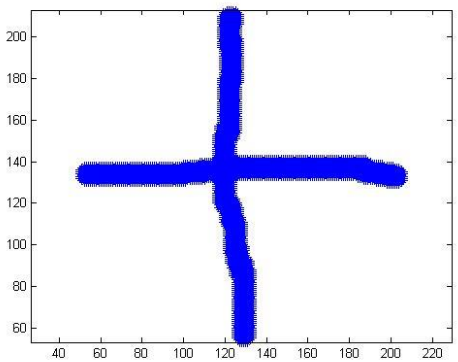
(c) (d)

图 1.1

3. 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 1.2（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 1.2（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 1.2（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。



(a) (b)

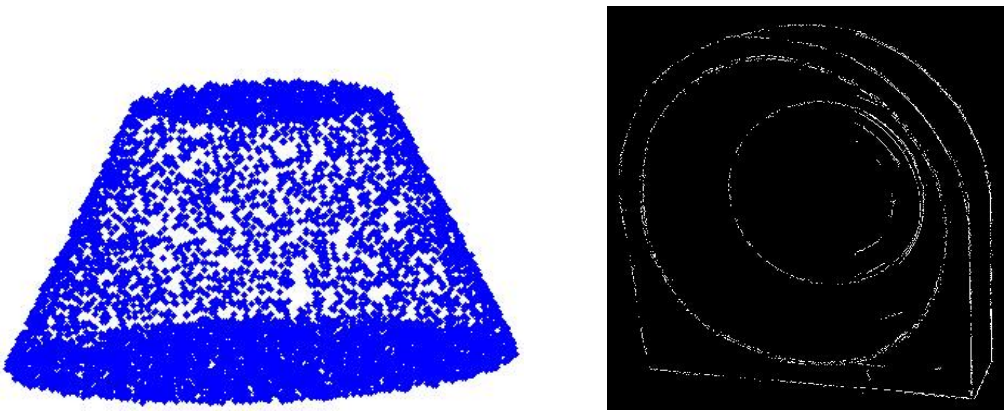
图 1.2

（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。

4. 请作答如下两个实际应用中的多流形聚类问题

图 1.3(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)。

图 1.3 (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。



(a) (b)

图 1.3

二、问题分析

2.1 问题一

题目要求将一组子空间相互独立的高维数据分为两类。此问题最大的难点在于高维（大于 3 维）数据不可观察性，即无法通过观察确定聚类是否正确。此时可以使用 LLE 算法降低维度，使之可以观察，然后根据是否线性使用 SSC 算法或者 SMMC 算法。

2.2 问题二

问题二有四个小问题，(a)和(b)是线性子空间聚类，可以使用 SSC 算法。而（c）和（d）是非线性子空间聚类，适合使用 SMMC 算法。值得一提的是，本文中也采用了 SSC 算法成功分割了图(c)。方法是首先通过 LLE 算法增加维度，然后就能够用 SSC 算法成功划分。由此可见，增加维度确实是一种值得思考的分割方法。

2.3 问题三

问题三分为 3 个聚类小问题。图(a)看似线性，然而实际上只是近似线性，并且两条线都具有相当的宽度，聚集着大量的点，难以用 SSC 成功划分；而 SMMC 的使用范围较广，除了参数难以调校，推荐使用 SMMC。图（b）和图（c）的数据属于高维数据，可以用类似于问题一的方法来实现划分。

2.4 问题四

问题四的两个小问模型相对复杂，属于混合子空间聚类，划分难度很大。单纯的使用 SSC 或者 SMMC 难以产生满意的结果。考虑到 SMMC 算法的适用性，本文考虑将 SMMC 算法加以改进，最后成功解决第四题。

三、符号说明

表 1 符号说明

符号	意义
R^D	D 维实空间
$R^{D\times K}$	D × K维矩阵实空间
Label(i)	第i个聚类标签
G	描述数据点之间相似度的加权图
W	衡量成对数据点相似度的亲和矩阵(Affinity Matrix)
w_{ij}	亲和矩阵 W 的第 i 行，第 j 列，衡量第 i 个数据点与第 j 个数据点的相似度；相似度越大，值越大，一般取[0，1]
x_i	需要被划分的数据点
X	总数据点集$\mathrm{X}=\{x_i\in R^D,\mathrm{i}=1,2,\dots,\mathrm{N}\}$
e	自然对数
$\|*\|_p$	l_p范数
Ncut	规范割集准则(Normalized Cut)
X_i	第 i 个数据点集，$\mathrm{X}=\cup_{i=i}^{k}X_i$，可以与Label(i)对应
Ω_j	第 j 个平滑流形

四、模型建立与改进

4.1 谱聚类

高维数据在生活中无处不在，例如影像，音频文件，图像处理，计算机视觉，模式识别。生物信息学等。高维数据不仅增加了算法的计算时间，占用了更多的内存空间，而且由于周围环境下采样率的不足以及噪声的存在，使得高维数据的性能不尽如人意，这被称作为“维数灾难”。然而，高维数据是镶嵌在低维数据之中的，恢复数据的低维结构不仅有助于减少计算量和占用的内存，也减少了高维噪声对实际性能的影响。

在许多现实问题上，数据的分类是在高维空间中提取出低维数据的典型。子空间聚类的方法在图像处理、运动分割等过程中对数据的分类有很好地应用。聚类，就是将数据对象划分成几个不同的类别，使得同一类的数据尽可能相似，不同类的数据差别尽可能大[2]，目前子空间聚类的算法主要有四种：迭代、代数、统计以及基于谱聚类的方法。其中基于谱聚类的方法在近几年较为流行，与传统的聚类算法相比，它具有能在任意形状的数据集上聚类的优点。

谱聚类算法建立在谱图理论基础上，广泛应用于图像分割，计算机视觉和模式识别之中。该算法通过构造数据点之间的加权图 G，根据图 G 定义一个描述成对数据点相似度的亲和矩阵W，其元素值一般取[0，1]之间。然后，计算矩阵W的特征值和特征向量，然后选择合适的特征向量聚类不同的数据点。矩阵的特征值称为谱，所以该方法命名为谱聚类。

对于一个未标记过的数据点集

$$\mathrm{X} = \{x_i \in R^D, i = 1, \ldots \mathrm{N}\} \tag{4-1}$$

聚类分析的目的是把这些点分配到k个不同的子集，使相似的点属于同一个族群，不同的点进入不同的聚类族群。在谱聚类中，数据点的邻近图通过一些标准来构建，该标准应当满足：若点x_i与点x_j在同一类的可能性越大，描述它们相似度的值w_{ij}越大。如

$$\mathrm{w}_{ij} = \begin{cases} e^{\frac{-\|x_i - x_j\|_2^2}{2\sigma^2}} & \mathrm{i} \neq \mathrm{j} \\ 0 & \mathrm{i=j} \end{cases} \tag{4-2}$$

其中，w_{ij}表示x_i与x_j之间的相似性。公式（4-2）只是通过点之间的距离来判断相似性，较为粗糙，对于复杂的子空间聚类问题不得得到很好的解。然后可以通过规范割集准则（Normalized Cut, Ncut）[3]进行划分：

$$\mathrm{Ncut}(\mathrm{X}_1, \cdots, \mathrm{X}_k) \triangleq \frac{1}{2}\sum_{i=1}^{k} \frac{W(\mathrm{X}_i, \bar{\mathrm{X}}_i)}{vol(\mathrm{X}_i)} \tag{4-3}$$

其中，$\mathrm{X}_1, \cdots, \mathrm{X}_k$是 X 的一个划分$(\mathrm{X}_1 \cup \cdots \cup \mathrm{X}_k = \mathrm{X}, \mathrm{X}_i \cap \mathrm{X}_j = \emptyset, i \neq j$并且$\mathrm{X}_i = \emptyset, i = 1, \cdots, k)$，$W(A, B) \triangleq \sum_{x_i \in A, x_j \in B} w_{ij}$，$vol(A) \triangleq \sum_{x_i \in A, j \in \{1, \ldots, N\}} w_{ij}$，其中，$\bar{A}$是$A$的补集。

注意到$W(\mathrm{X}_i,\bar{\mathrm{X}}_i)$的值小说明$\mathrm{X}_i$是一个较好集群$vol(\mathrm{X}_i)$的值大说明集群规模较大，这说明 Ncut 值越低，聚类效果越好。因此，目标是最小化 Ncut 算法。
最小化 Ncut 算法可以等价于[4]：

$$\min_{\mathrm{X}_1,\cdots,\mathrm{X}_k} Tr(H^T(E-W)H) \quad s.t.\ H^TEH=I \tag{4-4}$$

其中，E是$E_{ii}=\sum_j W_{ij}$的一个$N\times N$对角矩阵，I是单位矩阵，$H\in R^{N\times k}$是一个特殊的离散矩阵，Tr 是矩阵的迹。

然后需要构造拉普拉斯矩阵$L=D^{-1/2}AD^{-1/2}$。对其进行特征值分解，找出前 k 个最大特征值所对应的特征向量$u_1,u_2,\ldots,u_k$，然后构造矩阵$U=[u_1,u_2,\ldots,u_k]\in R^{N\times k}$，其中特征向量按列存储。对$U$的行向量归一化为$Y$，$Y_{ij}=U_{ij}/(\sum_j U^2{}_{ij})^{1/2}$。

把Y的每一行作为空间R^k中的样本（样本数为N，样本维数为k），然后对其用 K-means 聚类。最后把最初的样本点s_i划分为第j聚类，当且仅当矩阵Y的第 i 行被划分为第 j 聚类[3,5]。

4.2 稀疏子空间聚类 SSC

SSC（sparse subspace clustering）算法[6,7]是一种基于稀疏表示技术的算法，用于在低维子空间联合下聚集数据点，适用于多个线性子空间的聚类。SSC 算法利用了数据的“自表现性”，即子空间联合中的每个数据点可以充分的被表示为其他点的一个线性的或者放射性的组合，且这种表示的方法不唯一。一个数据点的稀疏表示为来自与它的子空间的一些点的组合，这种算法是一种全局谱聚类算法的优化算法，克服了局部谱聚类算法的一些局限性，对于一个给定的数据点，SSC 对于那些与之在同一子空间中但关系不大的点有自动屏蔽作用。

假定 $\{S_l\}_{l=1}^n$ 是 n 个 D 维向量 $\{d_l\}_{l=1}^n$ 子空间的排列，采集到了 N 个无噪声的数据点$\{x_i\}_{i=1}^N$ ，这些数据点存在于 n 个子空间联合中，表示包含了所有点的矩阵为：

$$\mathrm{X}\triangleq[x_1\ \ldots\ x_N]=[X_1\ \ldots\ X_n]\,\Gamma \tag{4-5}$$

$X_l\in R^{D\times N_l}$ 是一个在 S_l 中 $N_l>d_l$ 的点的 d_l 阶的矩阵，$\Gamma\in R^{N\times N}$ 是一个未知的置换矩阵。假设事先不知道子空间的信息，也不知道哪个点属于哪个子空间，子空间的聚类问题转化为了求解子空间的数量，它们的维数以及数据点的分割问题。解决这个子空间聚类的问题分为两个步骤，第一步，对于每个点，我们找到一些与之在同一个子空间的点，这样就找到了对于每一个点来说潜在的子空间；第二步，使用第一步得到的谱聚类框架的潜在信息来推断数据的聚类。根据数据的“自表现性”，每一个数据点 $x_i\in\cup_{l=1}^n S_l$ 可以被写为

$$x_i=Xc_i,\ c_{ii}=0 \tag{4-6}$$

其中 $c_i\triangleq[c_{i1}\ \ldots\ c_{iN}]^T$ 以及 $c_{ii}=0$ 消除了数据点与它自身的线性对结果的影响，X表示了每个点被其他点线性表示出来系数，且不唯一。式（4-6）存在一个稀疏解 C_i ，其对应于来自同一子空间的数据点的非 0 项与 x_i 一致，称之为子空间的稀疏表示。具体来说，一个在 d_l 维子空间 S_l 中的点 x_i ，可以被子空间中的其他点线性表示出来。一个点的稀疏表示可以从来自与同一子空间中非零元素对应的潜在子空间维数中获得。

然而，式（4-6）这样的系统方程有无数解，一种合适的约束条件如下：

$$\min\|c_i\|_q \qquad \text{s.t.} \qquad x_i=Xc_i,\ c_{ii}=0 \tag{4-7}$$

c_i 的 l_q 范数定义为 $\|c_i\|_q\triangleq\left(\sum_{j=1}^N|c_{ij}|^q\right)^{\frac{1}{q}}$

选择不同的 q 会得到不同的方案，随着 q 从无穷大到零，稀疏解递增。极端情况当 q=0 时，会得到一个最稀疏的解。但，l_0 范数最小化是 NP 完全问题，无法有效求解。因此，用l_1 范数最小化来近似代替，

$$\min\|c_i\|_1 \quad \text{s.t.} \quad x_i = Xc_i, c_{ii} = 0 \tag{4-8}$$

式（4-8）可以运用稀疏解的凸规划算法求解。

考虑在矩阵中的所有点 $\mathrm{i} = 1, \ldots, \mathrm{N}$ ，重写（4-8）式，有：

$$\min\|C\|_1 \quad \text{s.t.} \quad X = XC, \mathrm{diag}(C) = 0 \tag{4-9}$$

其中矩阵 $\mathrm{C} \triangleq [c_1 \ldots c_N] \in R^{N\times N}$ 的第 i 列 c_i表示了x_i的稀疏表示， $\mathrm{diag(C)} \in R^N$ 是矩阵 C 的对角元素的向量。可以用式（4-9）的子空间稀疏表示推断数据的不同聚类。

得到优化算法（4-9）的解后，就有了每一个数据点的稀疏表示。构造权重图$\mathrm{G} = (\nu, \varepsilon, \mathrm{W})$，其中 ν 表示一组 G上的 N 个节点与 N 个数据点的对应，$\varepsilon \subseteq \nu \times \nu$ 表示一组节点间的边界，$\mathrm{W} \in R^{N\times N}$ 是一个对称非负相似矩阵，表示了边界的权重值，节点 i 与节点 j 相连的边界权重为w_{ij}。理想情况下，来自与同一子空间的节点相互连接而来自与不同子空间的节点之间没有连接。由于非零项对应的点来自于同一个子空间，有$\mathrm{W} = |C| + |C|^T$，于是节点 i 与节点 j 的连接权重为 $|c_{ij}| + |c_{ji}|$。

类似地对于 n 个子空间，有：

$$W = \begin{bmatrix} W_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & W_n \end{bmatrix} \Gamma \tag{4-10}$$

其中 W_l 是 S_l 中数据点的相似矩阵。对于G 运用谱聚类的方法可以将数据点进行分类。

算法(4-1)总结了 SSC 算法，只要不同子空间之间点的边界连接比较弱，谱聚类算法可以正确的进行分类。

算法(4−1)：稀疏子空间聚类算法（SSC）[7]

输入：n 个线性子空间 $\{S_i\}_{i=1}^n$ 中的 N 个点 $\{x_i\}_{i=1}^N$

1 求解优化方程（4-9）

2 对 C 进行归一化

3 构建相似矩阵 W，$\mathrm{W} = |C| + |C|^T$

4 运用谱聚类的方法求解相似矩阵 W

输出： 分类数据： $X_1, \ldots, X_n$

4.3 多流形谱聚类 SMMC

对于一个未标记过的数据点集

$$\mathrm{X} = \{x_i \in R^D, i = 1, \ldots, \mathrm{N}\} \tag{4-11}$$

其中数据点采样自$k > 1$的不同的平滑流形

$$\{\Omega_j \subseteq R^D, j = 1, \ldots, \mathrm{k}\} \tag{4-12}$$

平滑流形之间会产生交集，而流形聚类算法的目的就是把流形的每一个样本分

配到它们所属的位置。所以可以假定所有的流形都有同样的维度$d(0 < d < D)$，同时对应的子空间个数k都是已知的。然后，我们可以通过以下步骤来解决混合的非线性流形的聚类问题：

首先，通过经典谱聚类找出所有的连接部分。

其次，分开这些连接部分的交叉丛。

但是在此过程中无法判断连接部分是由单一流形还是多流形组成，此外，若为多流形交叉组成，也无法确定连接部分存在多少流形。在此基础上，文献[8]提出了多流形谱聚类的方法方法(Spectral Multi-Manifold Clustering, SMMC)。SMMC 算法是一种典型的谱聚类算法，其关键在于亲和矩阵的构造，至于之后利用 Ncut 算法分割，与其他谱聚类算法类似，不在赘述。

4.3.1 亲和矩阵的构造

谱聚类的关键何难点在于亲和矩阵的构造，在属于不同聚类的点之间的关联值很低的情况下，谱聚类方法可以得到很好的结果。但是基于点之间欧式距离的传统亲和矩阵并不适用于大多数混合子空间聚类。因此 SMMC 的思想是根据样本数据的几何信息重新构建一个更加合适的亲和矩阵来正确分类。

虽然这些数据位于或接近平滑的非线性流形，但是局部上每个数据点和其临近点位于流形的一个线性区域上[9,10]。此外，对于非线性流形的局部几何结构，每个点的局部切线空间（Local Tangent Space）提供了一个很好的低维线性逼近[11]。最终，在不同流形的交叉区域上，相同流形上的点有相似的局部切线空间，而不同流形上的点则具有不同的切线空间。

另一方面，对于距离较大的点，很难通过局部几何信息判断这些点是否是在同样的流形上，所以可以关注其局部区域。直观上，对于在同一局部区域上的两个点，如果：（a）彼此间很近（b）他们有相似的局部切线空间，那么可以认为这两个点有很高的可能性在同样的流形上。对于临近的点，如果他们的局部切线空间不同，他们很有可能来自于不同的流形。因此，应该考虑位于x_i和x_j两点之间的关联函数，其中一个定义为对应的局部切线空间的函数（结构相似性函数p_{ij}），另一个通过欧式距离$q_{ij} = q(\| x_i - x_j \|)$来定义（局部相似性函数）。通过这两个定义可以得到下面的关联值：

$$w_{ij} = f(p_{ij}, q_{ij}) \tag{4-13}$$

其中，f是一个混合函数。值得注意的是，为了得到期望的亲和矩阵属性，f应该是一个关于欧式距离的单调减函数，同时是一个关于两个切线空间相似性的单调增函数。

假设在点$x_i(i = 1, \dots, N)$的切线空间是Θ_i，x_i与x_j两点之间的结构相似性可以被定义为：

$$p_{ij} = p(\Theta_i, \Theta_j) = (\prod_{l=1}^{d} \cos(\theta_l))^o \tag{4-14}$$

在上述公式中，其中$o \in N^+$是可调参数。$0 \le \theta_1 \le, \dots, \le \theta_d \le \frac{\pi}{2}$是两个切线空间$\Theta_i$与$\Theta_j$之间一系列的特征角，递归地定义为:

$$\cos(\theta_1) = \max_{\substack{u_1\in\Theta_i, v_1\in\Theta_j \\ \|u_1\|=\|v_1\|=1}} {u_1}^T v_1 \tag{4-15}$$

并且

$$\cos(\theta_l) = \max_{\substack{u_l\in\Theta_i, v_l\in\Theta_j \\ \|u_l\|=\|v_l\|=1}} {u_l}^T v_l, l = 2, \dots, d \tag{4-16}$$

其中，${u_l}^T u_i = 0, {v_l}^T v_i = 0, i = 1, \dots, l-1.$

局部相似性可以定义为：

$$q_{ij} = \begin{cases} 1 & x_i \in Knn(x_j) \text{或} x_j \in Knn(x_i) \\ 0 & \text{其他} \end{cases} \tag{4-17}$$

其中，$Knn(x)$表示最接近x的 K 个点构成的集合。
最后，根据这两个函数得出关联值

$$w_{ij} = p_{ij}q_{ij} = \begin{cases} (\prod_{l=1}^{d} \cos(\theta_l))^o & x_i \in Knn(x_j) \text{或} x_j \in Knn(x_i) \\ 0 & \text{其他} \end{cases} \tag{4-18}$$

容易发现上述公式定义的相似度w_{ij}具有期待的属性，如：属于不同流形或子空间的点，相对的取值较低。这是因为当来自于不同流形的两点之间的距离较远时，它们的相似度w_{ij}为 0。同时，当它们距离不同流形的交叉区域很近时，它们的局部切线空间不相似；而当参数 o 很大的时候也会有一个相对较低的相似度。因此将谱聚类方法用于这类矩阵时，会得到较好的预计结果，具体算法流程见算法（4-2）。注意到过程中有用到概率主成分分析（MPPCA），所以有时候相同的参数会得到不同的结果，一般最多只会出现两个不同的结果。鉴于此，本文用投票的方法降低出现不同结果的概率：多次运行 SMMC 程序，根据产生的结果投票，若某个点的第i个聚类标签(记为Label(i))的票数最多，则该点属于Label(i)。

算法(4-2)：多流形谱聚类
输入：
数据集 X，子空间个数 k，流形的维数 d，混合模型个数 M，K-邻近个数 K，调节参数 o
过程：
1 用 MPPCA 训练 M 个 d 维的局部线性流形来近似潜在的局部流形；
2 计算每个点的切线空间；
3 用公式（4-14）计算任意两点切线空间的相似度；
4 用公式（4-18）计算亲和矩阵 W；
5 计算对角矩阵 E，其中$E_{ii} = \sum_j w_{ij}$
6 对于方程$(\mathrm{E} - \mathrm{W})\mathrm{u} = \lambda \mathrm{E}\mathrm{u}$，提取前 k 个广义特征向量$u_1, \dots u_k$
7 用 k-means 算法聚类 U 的行向量

4.4 局部线性嵌入 LLE[9]

LLE 刻画了数据点之间的局部特性。其基本思想就是：首先将非线性数据局部线性化，即点 x 可以被其邻近的点线性表示，详细证明请见定理（1）；然后在充分保持数据局部集合关系的情况下寻求最佳的低维嵌入表示。

定理（1） 空间中的一点$\mathrm{x}\epsilon R^d$,可以被其邻近的点集$\mathrm{x}_1,\ldots,x_k$线性表示为

$$\mathrm{x}=\sum_{i=1}^{k}w_i x_i \quad s.t.\sum_{i=1}^{k}w_i=1 \tag{4-19}$$

证明：

在欧式距离下的误差为

$$\mathrm{e}=\left\|x-\sum_{i=1}^{k}w_i x_i\right\|_2=\left\|\sum_{i=1}^{k}w_i\,x-\sum_{i=1}^{k}w_i x_i\right\|_2=\left\|\sum_{i=1}^{k}w_i(x-x_i)\right\|_2$$

$$\leq\sum_{i=1}^{k}w_i\|x-x_i\|_2$$

若$\|x-x_i\|_2\ (\mathrm{i}=1,2,\ldots,\mathrm{k})$极小，则误差 e 显然也是极小的

证毕

LLE 算法的流程如下：

1 对每个数据点x_i，寻找其最近的 k 的点$x_j^i,\ x_i\neq x_j^i,\ j=1,\ldots,k$

2 对所有数据点，计算最优权重

$$\min_{w_{ij}}\sum_{i=1}^{N}\left\|x_i-\sum_{i=1}^{k}w_{ij}x_j^i\right\|_2^2 \tag{4-20}$$

3 对于权重w_{ij}，计算低维嵌入表示

$$\min_{y_i}\sum_{i=1}^{N}\left\|y_i-\sum_{i=1}^{k}w_{ij}y_j\right\|_2^2$$

4.5 逐步多流形谱聚类 GSMMC 和增加数据维度

对于混合多子空间分类（子空间数大于 2），我们提出了一种方法就是逐步多流形谱聚类 GSMMC。该方法的核心思想就是：一个一个逐步递进的划分，首先划分成两类，然后根据划分的结果修改亲和矩阵 W，接着根据得出的 W 划分为三类，依此类推，直至划分为要求的 n 类。

GSMMC 的流程如下：

输入：数据集 X，子空间个数 n，k-邻近个数 k，其余参数见 SMMC 算法

初始化 i= 2：

当$\mathrm{i}\leq\mathrm{n}$时，重复如下步骤：

1 用 SMMC 将数据集 X 划分为 i 类

2 对于每一点x_j,若$\mathrm{x}_j\epsilon Label(h),\ h\epsilon[1,i]$,则从 Label(h)中选择 k 各最邻近的点组成集合$Knn(x_j)$

3 根据公式（4-18）计算任意两点之间的相似度w_{ij}，得到调整之后的亲

和矩阵 W

4 i=i+1;

结束

输出：数据集 X 的聚类结果$X_1,\ldots,X_n$

实验可以验证，GSMMC 对于处理第四题的图(a)有很好的效果，但是对于图(b)的分类效果却很难令人满意，所以在 GSMMC 的基础上又增加了一个处理数据的手段：扩张数据集的维数，使不同类别的数据越远越好。基于此可以对给定的数据集$\mathrm{X}=\{x_1,\ldots,x_N\}$做如下处理：

1. 首先计算数据的平均值$\bar{x}=\frac{1}{N}\sum_{i=1}^{N}x_i$
2. 对于每个点x_i，计算x_i与$\bar{x}$之间的欧式距离$d_i=\|x_i-\bar{x}\|_2$
3. 将d_i作为点x_i的新的一维

经过上述 3 个步骤，我们命名为平均值-欧式距离扩维法我们成功将数据集 X 扩张了一维。更进一步，在 GSMMC 的过程中，我们可以根据分类的结果扩张维数，具体方法如下：

1. 在 GSMMC 迭代过程中，已经分割出了 n 个不同的类别$\{X_1,\ldots X_n\}$；
2. 在每个类别X_i中使用上述的平均值-欧式距离扩维法，只是计算的平均值是X_i中点的平均值。

实验结果表明，上述的平均值-欧式距离扩维法可以有效提升分割的效果，但是提升的效果有限，对于第四题的图(b)任然不能有一个令人满意的结果。这是我们不愿意看到的，所以在进行 GSMMC 和平均值-欧式距离扩维法之前，先对数据进行预处理：直线探测，并剔除直线。如此一来，就能简化数据集的模型，然后能用 GSMMC 更好的分割。

4.6 RANSAC[12]

直线探测的目的是探测出混合子空间中的直线或者线性子空间，我们使用的方法就是 RANSAC[12]。RANSAC 方法普遍用于线性拟合和图像配准[14]等领域。RANSAC 是利用随即采样若干数据点来进行直线拟合的稳定的估算过程。其目的是为了消除异常点的影响，基本步骤如下：

设置误差参数 e，内窗层的点的个数 k，循环次数 t；

初始化 j=1；

1. 从数据集中随即采样 n 个点，进行直线拟合，得到直线L_j；
2. 计算每一点x_i到直线L_j的距离d_i；
3. 若$d_i<e$，则认为点x_i是内窗层的点；
4. 计算内窗层点的个数为p_j；
5. 若$\mathrm{p}_j\geq\mathrm{k}$，则保留$\mathrm{p}_j$和$\mathrm{L}_j$
6. j=j+1；
7. 重复过程 1-6 直至 j=t，计算出了 t 个内窗层点的个数$\mathrm{p}=\{p_1,\ldots,p_t\}$，在

p 中选取最大的数为p_m，则我们认为该数据集拟合出的直线为L_m。

RANSAC 确实是一个非常有效的估算方法，一个直观的线性拟合的流程可以见图 4-1。RANSAC 为我们提供了良好的线性拟合手段，如此我们可以剔除一部分线性子空间的影响，改善聚类的结果。然而，实际结果如图 4-2 所示，一共探测出了蓝、红、绿三条直线，而我们所需要探测的是蓝色的那条线。为了解决这个问题，我们利用了这副图的特殊性，那就是蓝线的斜率比红线和绿线的斜率都要大，利用这一特性我们成功分割出了我们需要的那条蓝线，如图 4-3 所示。

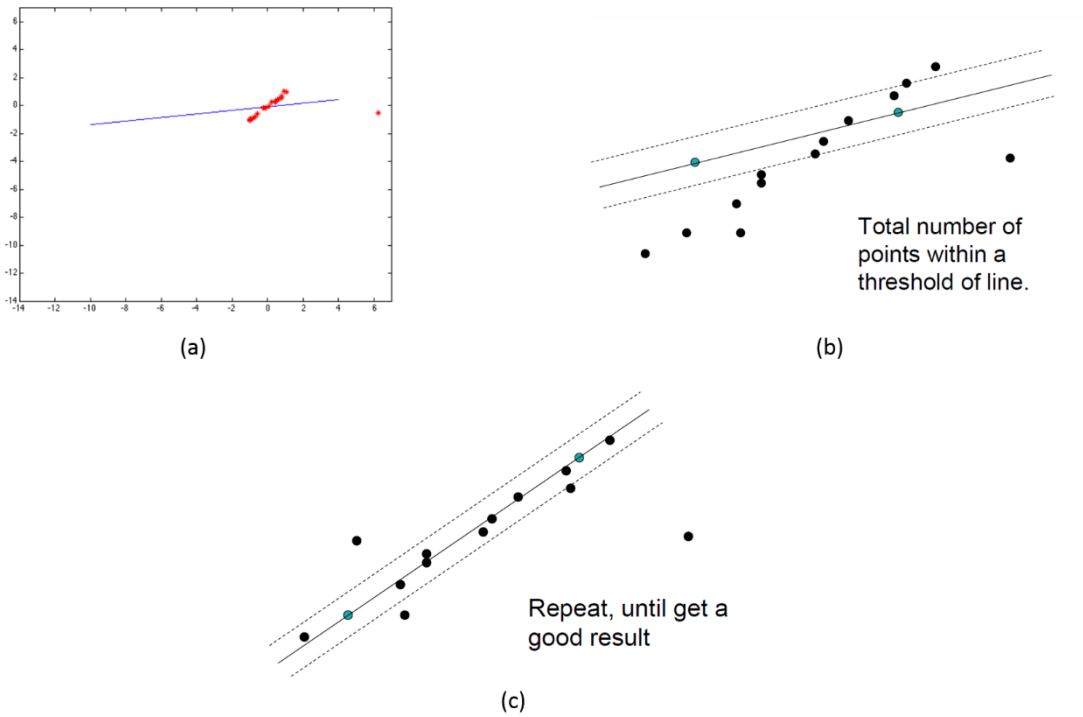


图 4-1 使用 RANSAC 进行线性拟合.(a) 异常点会干扰线性拟合的结果；(b) 随即采样两点进行线性拟合，然而内窗层的个数较少，舍弃此次拟合；(c) 循环数次，选取内窗层个数最多的一次拟合，即为所求得的拟合直线。

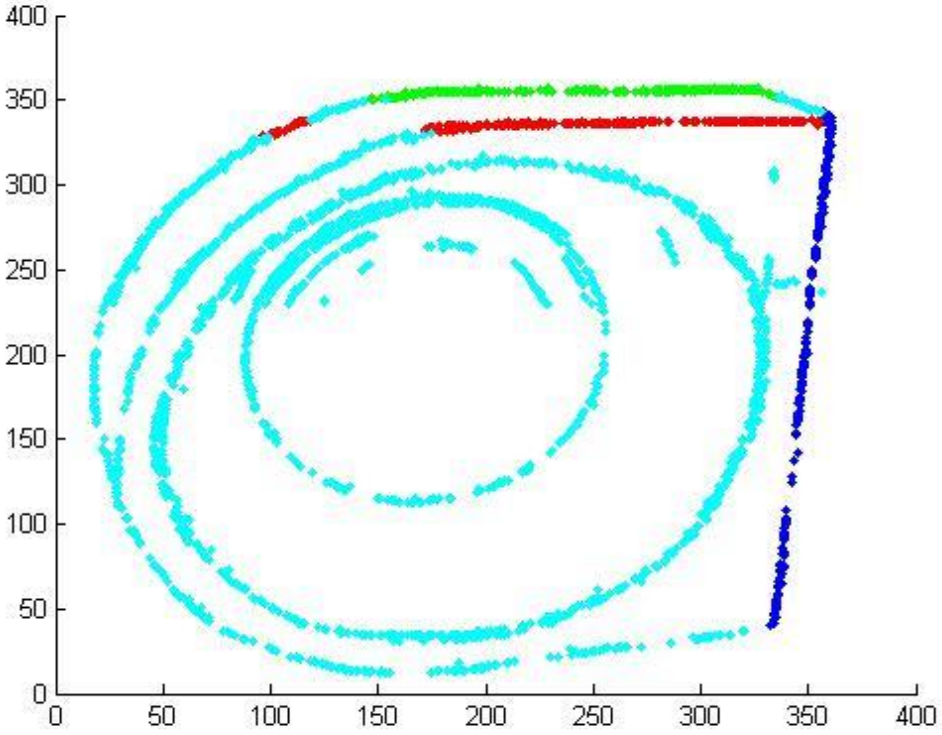


图 4-2 使用 RANSAC 拟合直线结果，一共探测出蓝、红、绿三条直线

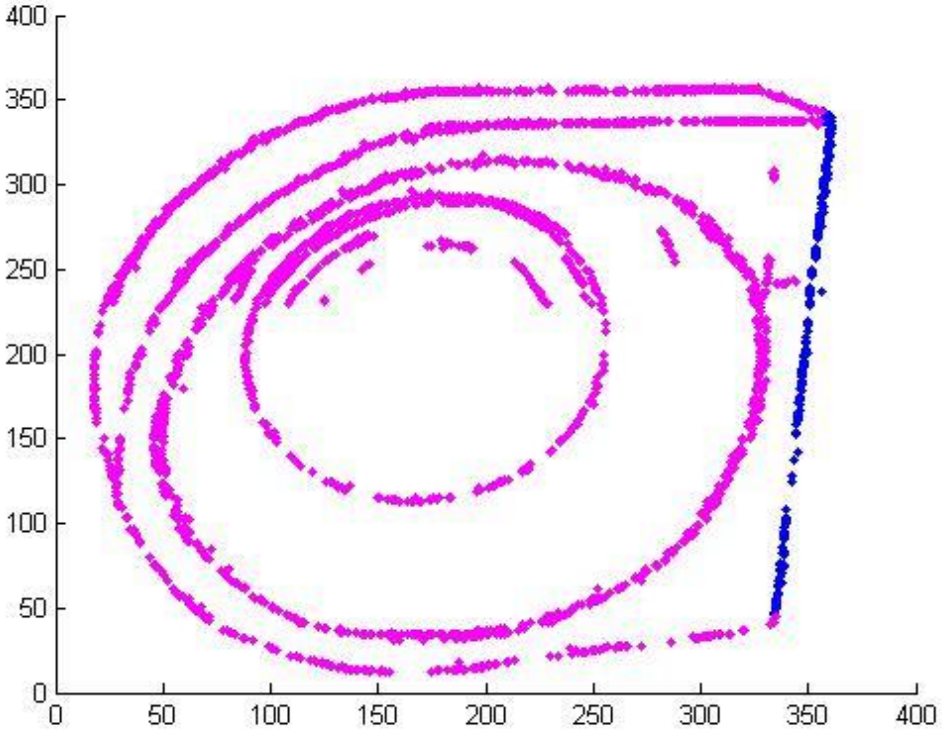


图 4-3 使用 RANSAC 拟合直线，然后利用斜率之间的差异只保留蓝线

五、问题求解

5.1 问题一——子空间独立

由于问题一是 100 维数据，共有 200 点，所以首先用 LLE 降到 3 维，观察一下数据呈现的分布情况，如图 5-1 所示。明显可以观察到 200 个数据点呈现的是两类线性分布，与题目要求的分为两类要求相吻合，所以可以用 SSC 来聚类。SSC 的聚类结果如图 5-2 所示。图 5-2（a）和（b）显示的结果是相同的，说明对于线性的子空间相互独立的两类，首先通过 LLE 降维之后聚类也是一种可行的方法。图 5-3 是用 SMMC 算法聚类的结果，可以得到与 SSC 相同的结果，表明 SMMC 适用于多类相互独立的线性子空间的聚类问题，且对于高维数据也是成立的。

问题一的类别标签见表 5-1.每行 20 个标签，共 10 列，第一行第一列是第 1 个点的标签，第一行第二列是第二个点的标签，第二行第一列是第 21 个点的标签，依此类推，遵循从左往右，从上往下的顺序。第三问的（b）和（c）中的表 5-2 和表 5-3 也是遵循这个顺序。

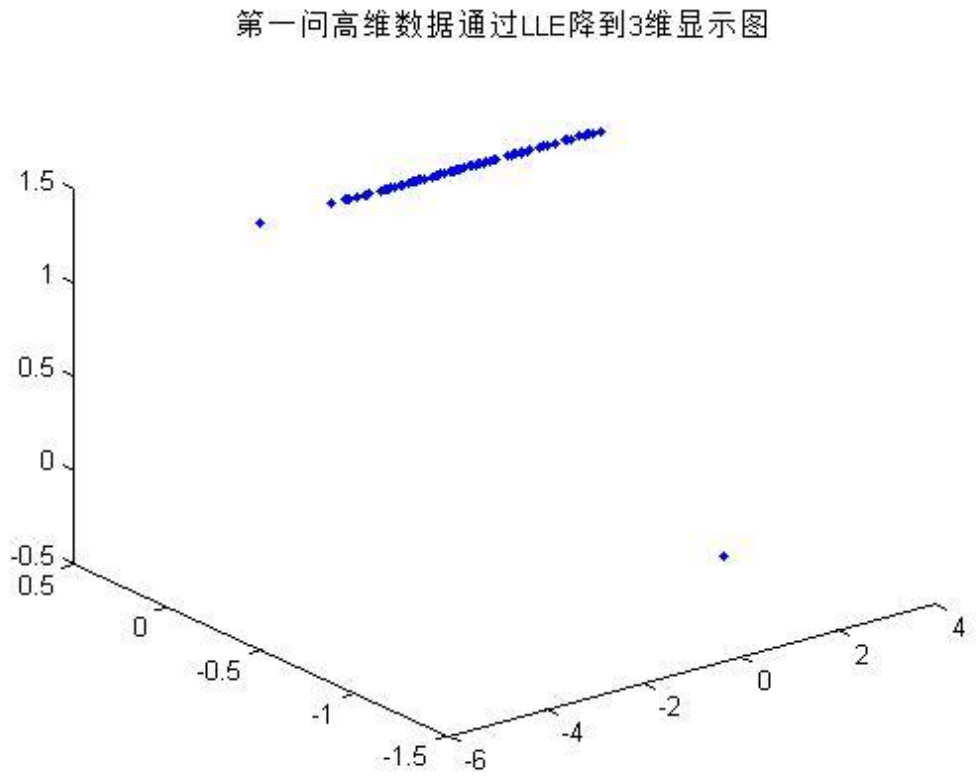
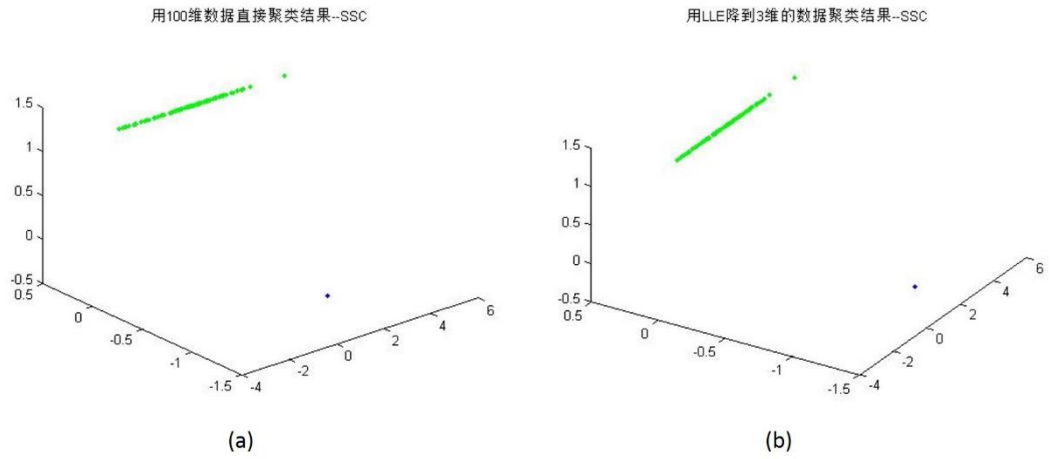
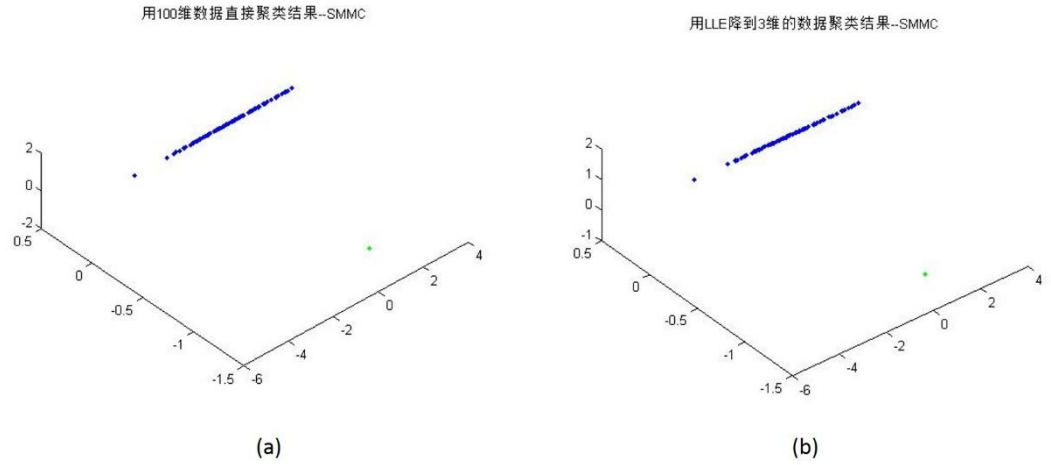


图 5-1 第一问 100 维数据通过 LLE 降到 3 维之后呈现的分布情况，明显观察到数据呈线性分布，可用 SSC 来聚类



(a) (b)

图 5-2 用 SSC 聚类的结果；（a）直接用 100 维数据聚类；（b）用 LLE 降到 3 维的数据聚类结果。



(a) (b)

图 5-3 用 SMMC 聚类的结果；（a）直接用 100 维数据聚类；（b）用 LLE 降到 3 维的数据聚类结果。

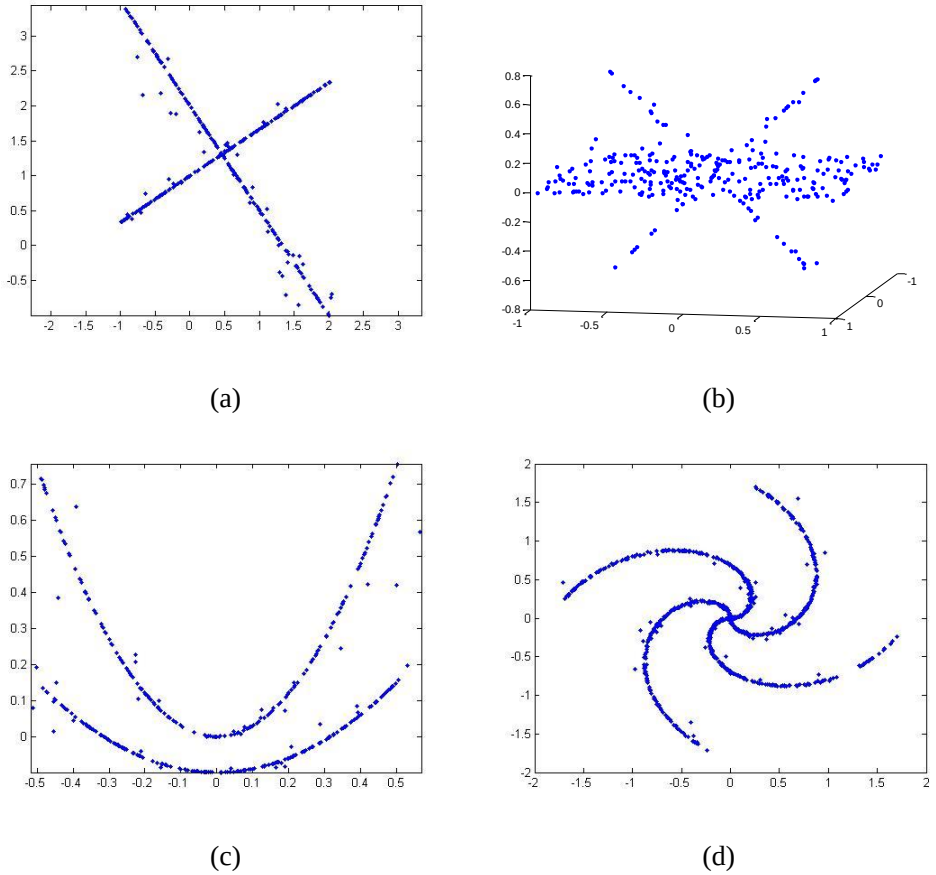
表 5-1 问题一的类别标签

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

5.2 问题二——低维空间中的子空间聚类问题和多流形聚类问题

问题二分为 4 个小问题，问题（a）、（b）、（c）、（d），分别如图 5-4 所示，均为低维的子空间聚类。问题（a）是两个相交的一维线性子空间聚类；问题（b）

是一个二维，两个一维的线性子空间聚类，这三个空间互相相交；问题（c）是两个不相交的非线性子空间聚类；问题（d）是两个相交的非线性子空间聚类。难度依次递增。

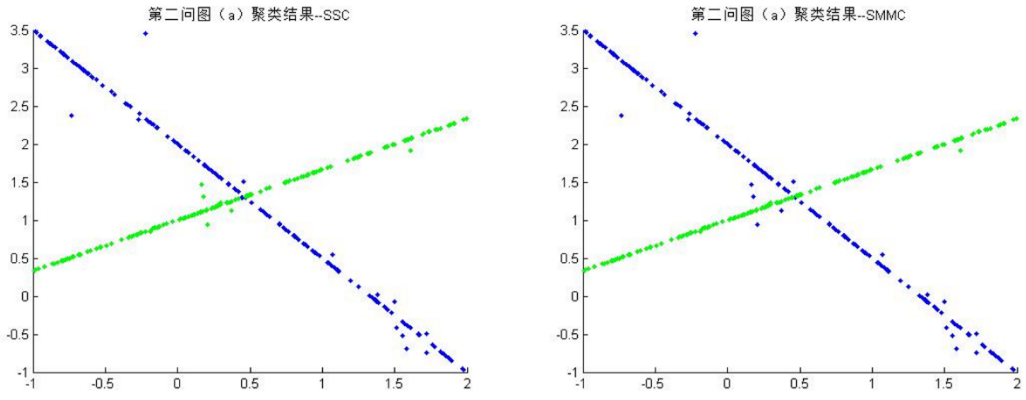


(a) (b)

(c) (d)

图 5-4 问题二的四个小问题（a）、（b）、（c）、（d）

5.2.1 问题二（a）



(a) (b)

图 5-5 问题二（a）的聚类结果；（a）SSC 聚类结果；（b）SMMC 聚类结果

问题二（a）的聚类结果如图 5-5 所示，图 5-5（a）是用 SSC 聚类的结果，图 5-5（b）是用 SMMC 聚类的结果。可以明显看到 SSC 和 SMMC 均适用于两个相交线性子空间的聚类。

5.2.2 问题二（b）

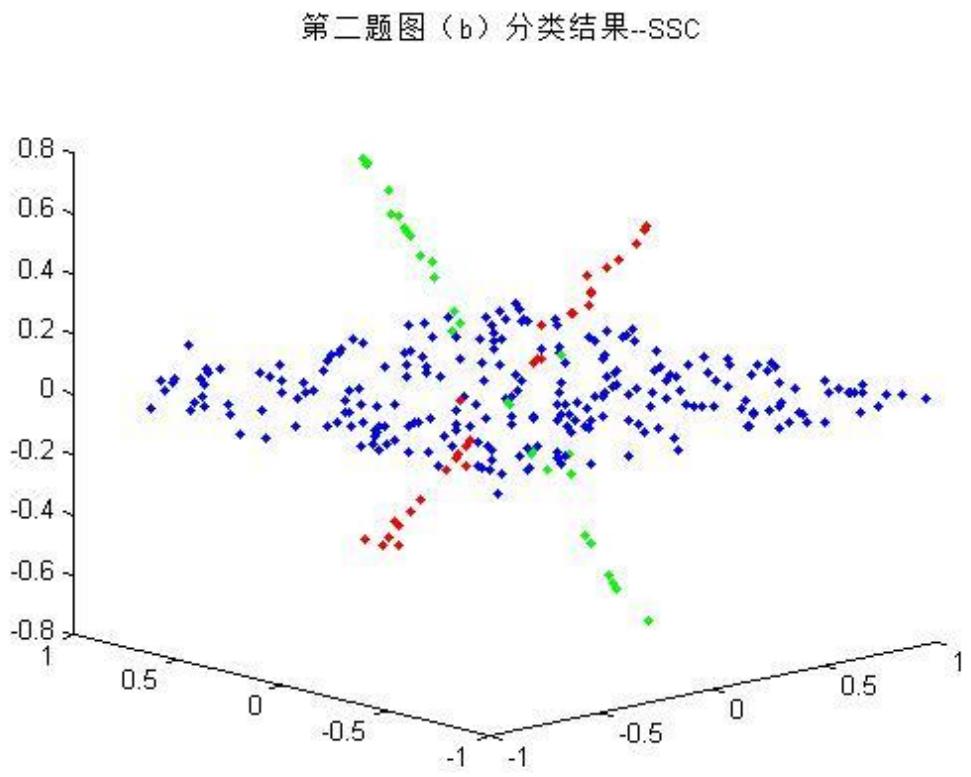


图 5-6 问题二（b）的聚类结果,用 SSC 实现

问题二（b）的聚类 结果图如 5-6 所示，是用 SSC 算法实现的聚类。本问题并没有 SMMC 算法的结果是由于 SMMC 算法虽然可以用于线性分类，适用范围广，然而 SMMC 的缺点是参数较多，并对参数的要求较高。从本题可以看出，SSC 针对性非常强，对于多个相交线性子空间的聚类能起到比 SMMC 更好的效果，在调参上也更有效率。

5.2.3 问题二（c）

问题（c）就比较有趣了，因为这是一个非线性子空间的聚类问题，本不适用于 SSC 算法，然而结果如图 5-7 所示。直接将二维非线性数据用 SSC 聚类，所得到的结果如图 5-7（a）所示，效果不好。然而神奇的是，将二维数据用 LLE 扩充到三维之后，数据分布情况如图 5-7（c）所示，是呈线性分布，所以可以用 SSC 算法进行聚类，聚类的结果如图 5-7（b）所示，效果很好，与 SMMC 算法聚类的结果相同。

通过此问题可以得到两个关于子空间聚类的启发：（1）扩充数据维度有时

会起到意想不到的作用；（2）LLE 算法不止可以用来降低维度，也可以扩充维度。

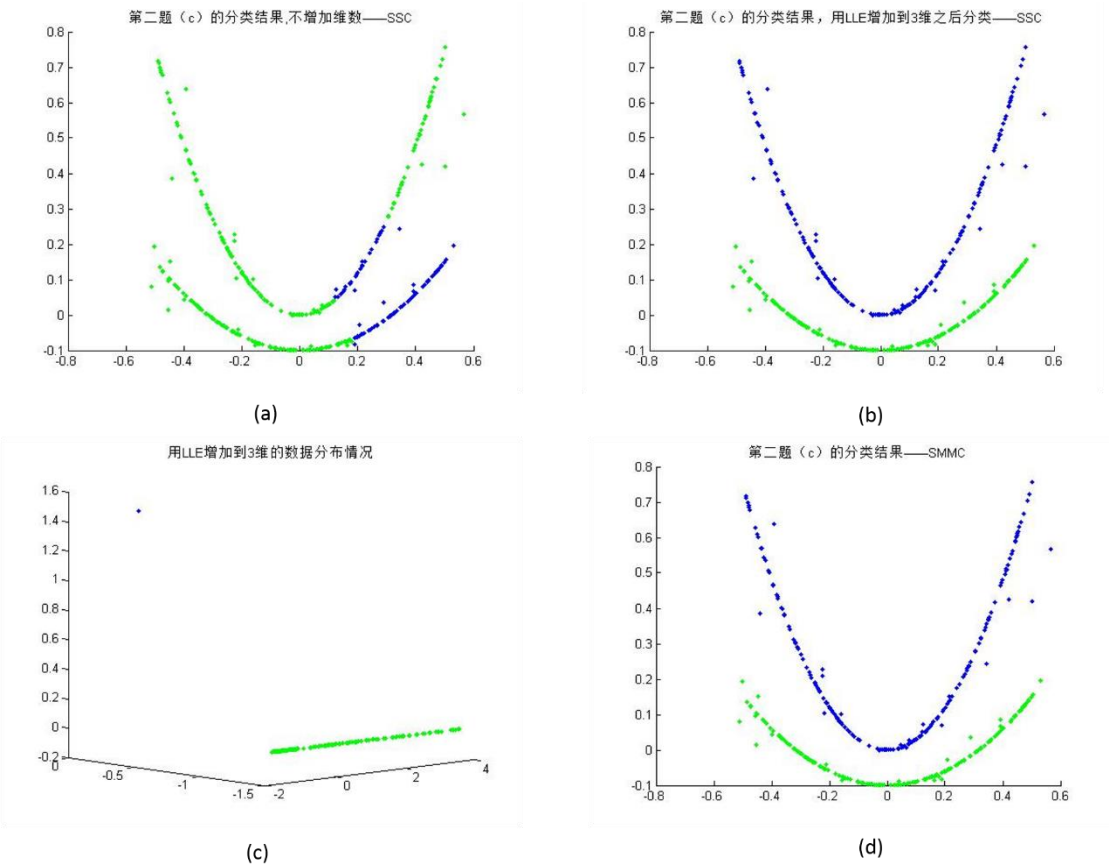


图 5-7 问题二（c）的聚类结果；（a）单纯用 SSC 聚类，效果不好；（b）将数据通过 LLE 算法扩充到三维之后，然后用 SSC 分类，效果良好；（c）扩充到三维之后的分布情况，为两个不相交的线性子空间；（d）单纯用 SMMC 算法聚类的结果

5.2.4 问题二（d）

问题（d）又比上述三个小问题复杂的多，要求分割出相交的两条螺旋线，聚类结果如图 5-8 所示。由于数据点呈现非线性性，并且两个子空间相交，SSC 算法不适用，所以采用了 SMMC 算法实现

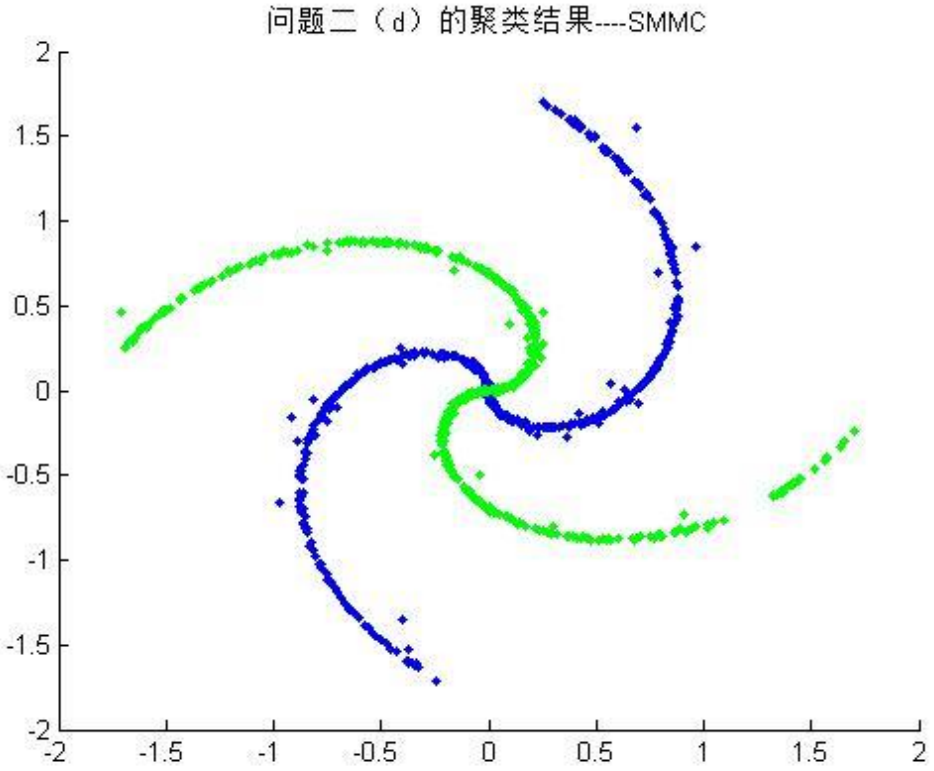
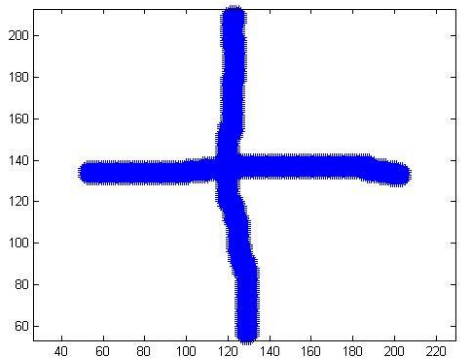


图 5-8 问题二（d）的聚类结果,用 SMMC 算法实现，并且为了降低 不同结果的概率，本文采用了投票的方法

5.3 问题三——实际应用中的子空间聚类问题

问题三分为三个聚类小问题，问题（a）是多个点的十字形分类，往往用于工业测量，分为两类，如图 5-9（a）所示；问题（b）是运动分割，将视频中不同的运动物体分开，分为 3 类，如图 5-9（b）所示；问题（c）是不同光照下的人脸识别，分为两类。



(a) (b)

图 5-9 问题三

5.3.1 问题三（a）

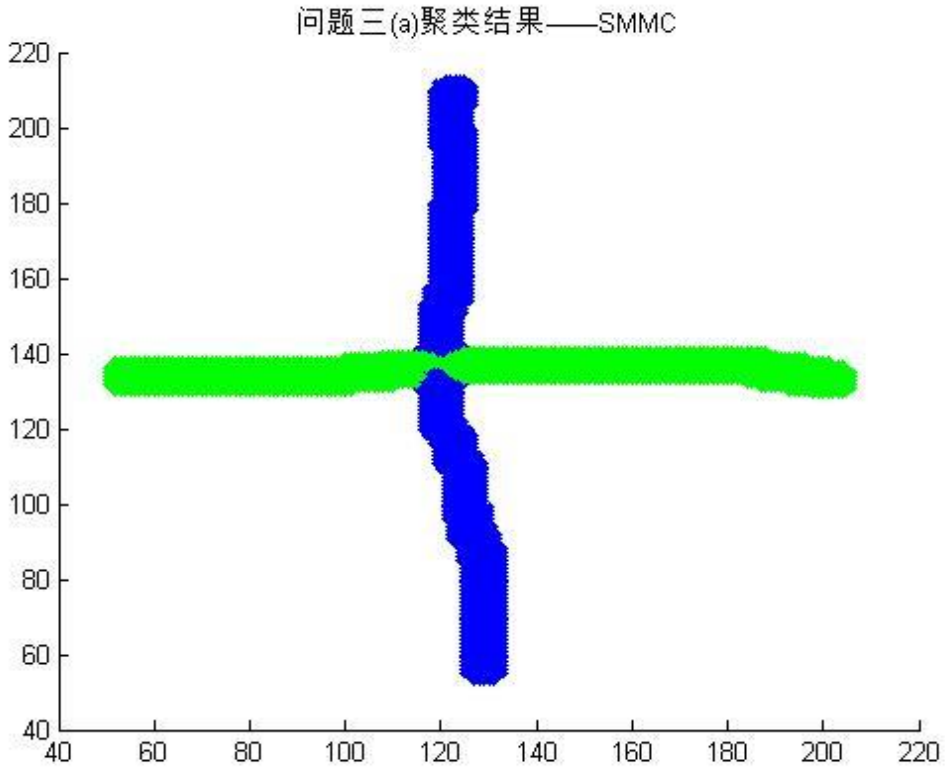


图 5-10 问题三（a）聚类结果，采用的是 SMMC 算法

图 5-9（a）看似两个一维线性子空间，其实两个二维的线性子空间，然而本问题中数据集较多，SSC 所需时间较长，所以采用了 SMMC 实现。这也暗示了随着数据点的增多，SSC 所需时间将大幅增加；而 SMMC 算法所需时间并没有大幅增加。

5.3.2 问题三（b）

问题三（b）又是一个高维聚类问题，数据集是 62 维，类似于问题一，以先使用 LLE 算法降维，观察其分布情况，然后再选择聚类方法：若分布情况为多个线性子空间，则用 SSC 算法；否则用 SMMC 算法。分布情况如图 5-11 所示，

数据点分为 3 个线性子空间，所以是用 SSC 算法。聚类结果如图 5-12 所示，图 5-12（a）与（b）的分类结果不一致。因为降低维度毕竟会损失一些数据，导致聚类错误，所以此处我们以图 5-12（b）为最终的聚类结果。类别标签见表 5-2.

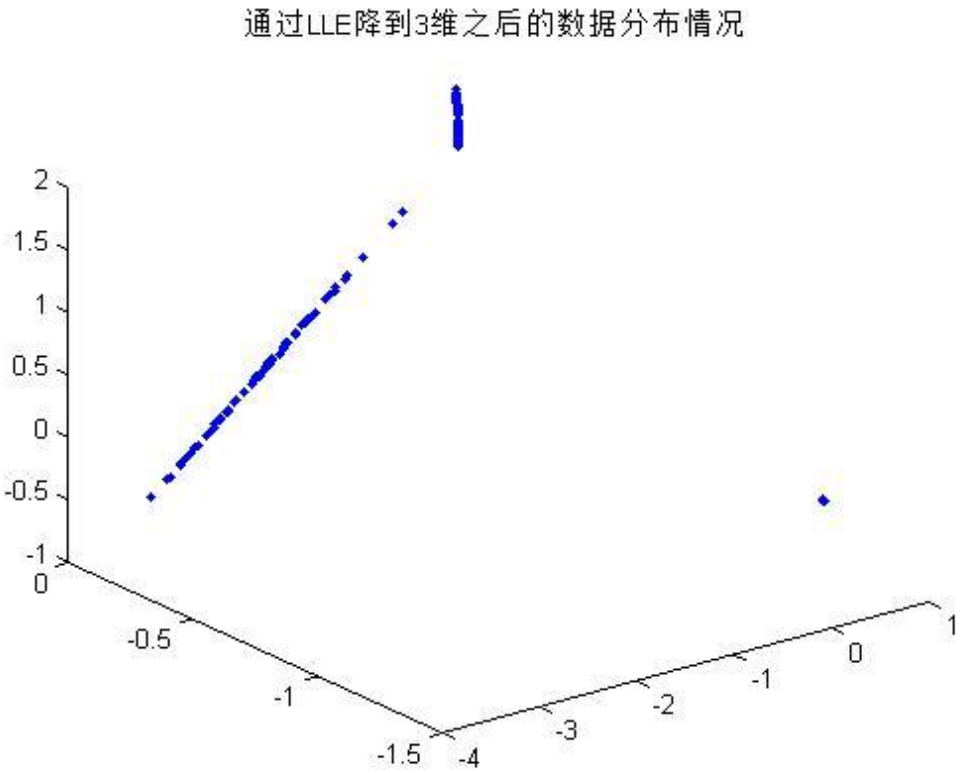


图 5-11 将问题三（b）的 62 维数据用 LLE 降到 3 维显示

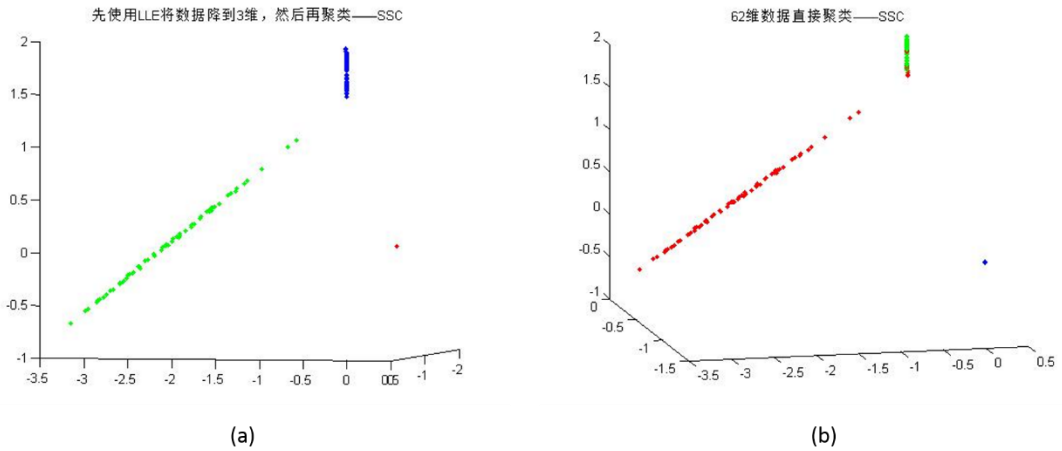


图 5-12 是用 SSC 聚类结果，将三个运动物体的点分离 （a）先使用 LLE 降维，然后聚类；（b）直接用 62 维数据聚类

表 5-2 问题三（b）的分类标签

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			

5.3.3 问题三（c）

本题的数据集仍然是高维度的，而且维度达到了 2016，在所有题目中，此数据维度是最高的。用 LLE 降维的情况如图 5-13 所示，明显呈两类近似线性分布，所以我们采用 SSC 算法来聚类，聚类结果如图 5-14 所示，两种情况的分类结果一致，将 20 个数据点分为两类，每类 10 个。类别标签见表 5-3.

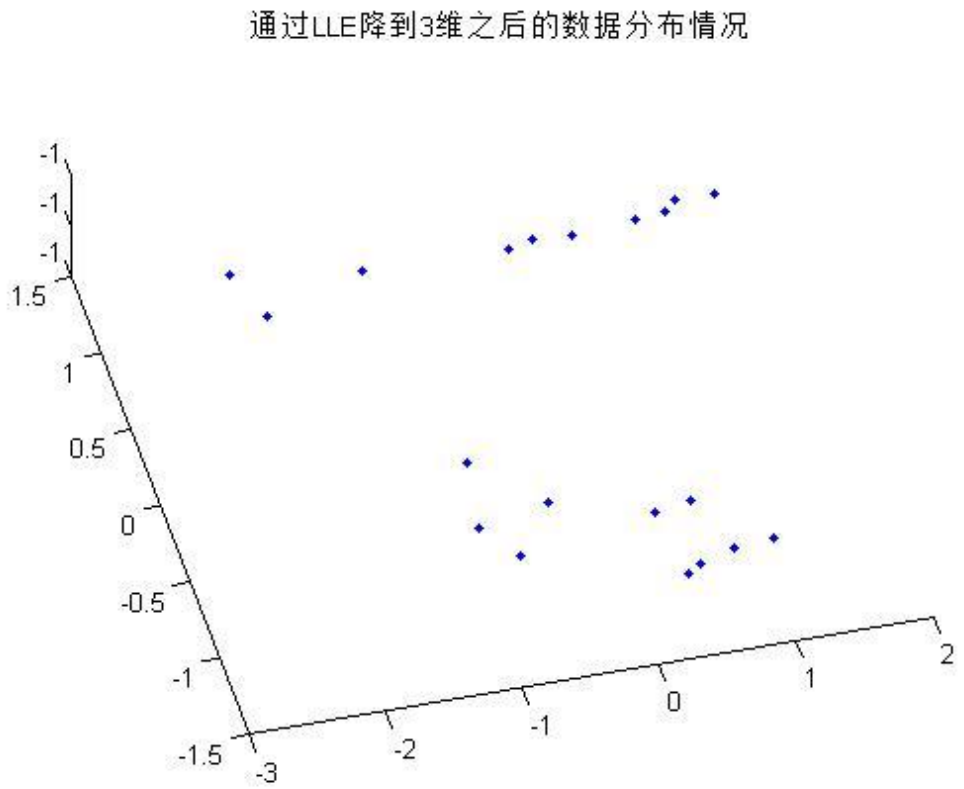
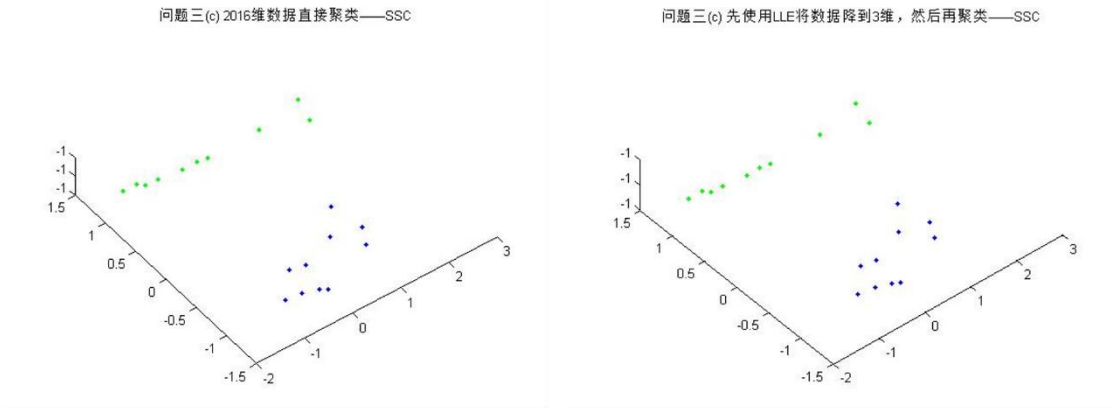


图 5-13 数据经过 LLE 降维之后的分布，呈两类近似线性分布。



(a) (b)

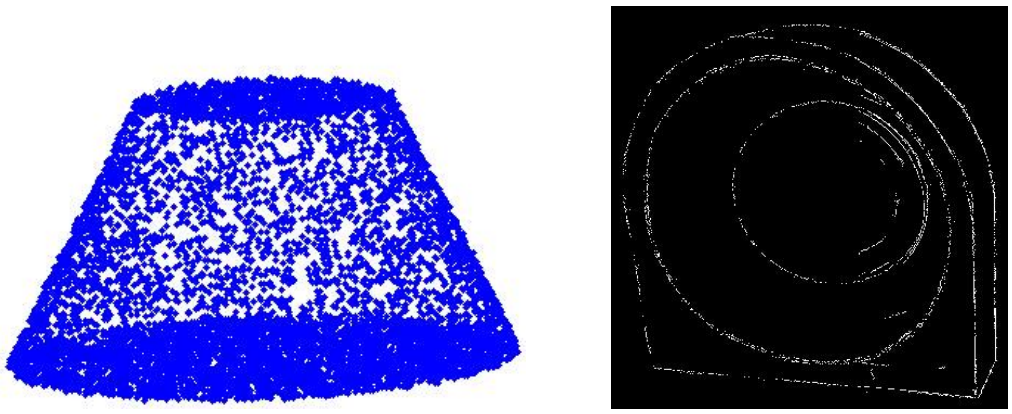
图 5-14 SSC 对问题三（c）聚类结果，将两个不同人脸识别；（a）直接用 62 维数据聚类；（b）先使用 LLE 降维，然后聚类。

表 5-3 问题三（c）的分类标签

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

5.4 问题四——实际应用中的多流形聚类问题

问题四分为两小题，如图 5-15 所示，难度较大。单纯的使用 SMMC 或者 SSC 算法都无法准确的分割问题四（a）和问题四（b），所以我们改进了 SMMC 算法，令其逐步分类，此方法称之为逐步多流形谱聚类。



(a) (b)

图 5-15 问题四

5.4.1 问题四（a）

对于图 5-15（a），需要分为 3 类，即圆台的顶、底和侧面。由于混合子空间类型复杂，SMMC 是无法完成这项聚类的任务。现在我们把这项聚类任务化简，首先将这个圆台分割为两类：圆台的顶和底为一类，圆台的侧面为另一类。由于侧面和顶底之间差别较大，尤其是切线空间更是不同，所以这项两类划分的

任务是相对容易完成的。

随后，根据分割的两类，修改亲和矩阵，使得不同类别之间的相似度为 0，即若点$x_i \in Label$(顶和底)，$x_j \in Label$(侧面)，那么亲和矩阵 W 的元素$w_{ij} = 0$，这时我们基本保证了，在以后的分割任务中，顶和底的点不会划分到$Label$(侧面)；同样，侧面的点也不会划分到$Label$(顶和底)。最后再使用 SMMC 进行 3 种类别的划分，这时就可以在上一次两类划分中，分为一类的顶和底分开——这项任务也是很简单就能完成，因为圆台的顶和底之间的欧式距离较远。分割的结果如图 5-16 所示。

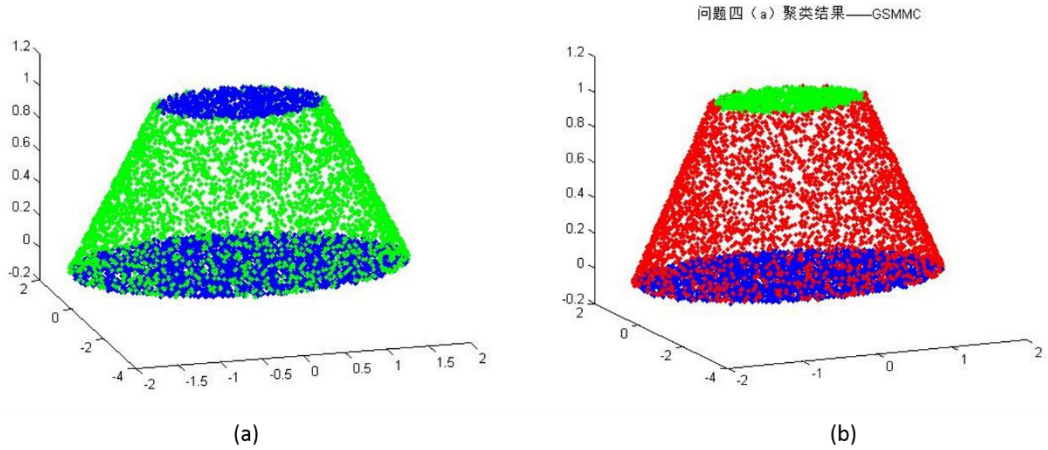


图 5-16 问题四（a）聚类结果；（a）GSMMC 中两分类的结果；（b）GSMMC 中三分类的结果，也是最终的结果

5.4.2 问题四（b）

图 5-15（b）是机器工件外部边缘轮廓图，相当的复杂，而且类数自定，即使 GSMMC 算法也不能完全得到满意的结果。其主要原因在于两点，如图 5-17 所示：

1. 左上角的圆弧距离相当接近，很容易误判成为一个类别中；
2. 红色圆圈内一部分的是噪声数据，这些点很容易影响最右边那条直线的划分。

基于上述两个原因，我们提出了两种方法改进数据集，分别为第四章中提到的直线检测和平均值-欧式距离扩维法。利用 RANSAC 进行直线检测的目的是检测出与噪声点相近的直线，从数据集中剔除，减少了噪声的影响。如图 5-18（a）所示，蓝色的线即为探测到的直线。

为了减少第一点的影响，增大不同流形中点的欧式距离，本文中引入了平均值-欧式距离扩维法。使用此方法的主要原因是基于如下的事实：

采样自圆形流形上的均匀分布的数据集，其平均值近似为圆心，而圆心与每个数据点的距离是近似相同的。所以以此距离为新的一维（在本例中是第三维）是合情合理的。因为剔除直线之后的图 5-18（b）中，剩下的流形几乎都是近似于圆的。这种方法对于存在有较多圆形的聚类问题比较有效。使用扩维法得到的 3 维数据如图 5-19 所示。

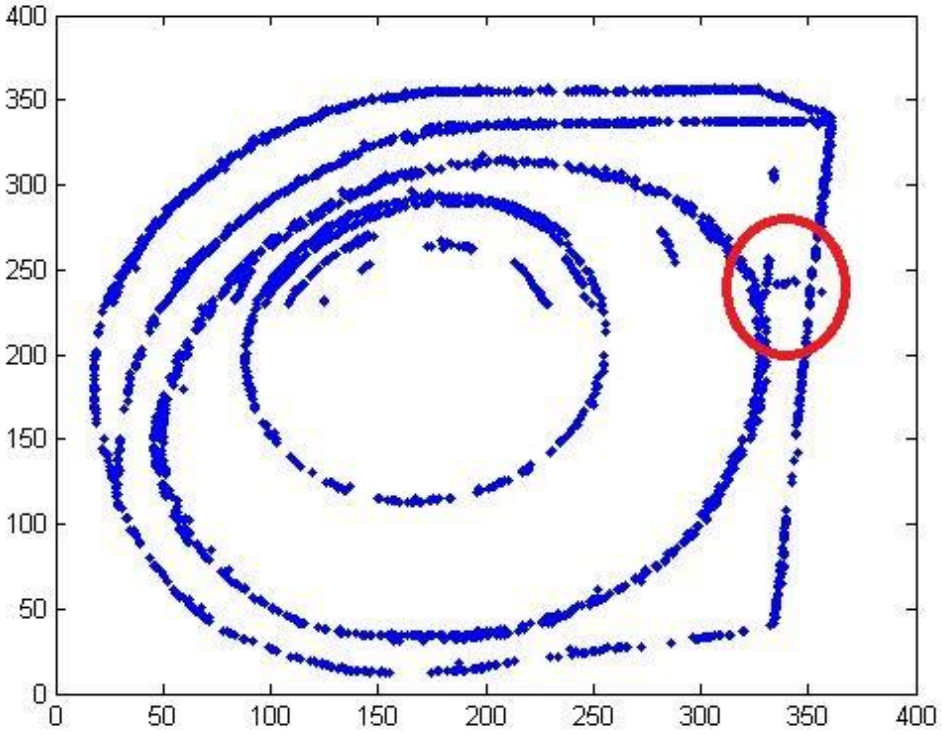


图 5-17 工件轮廓图，红色圆圈内的为噪声。

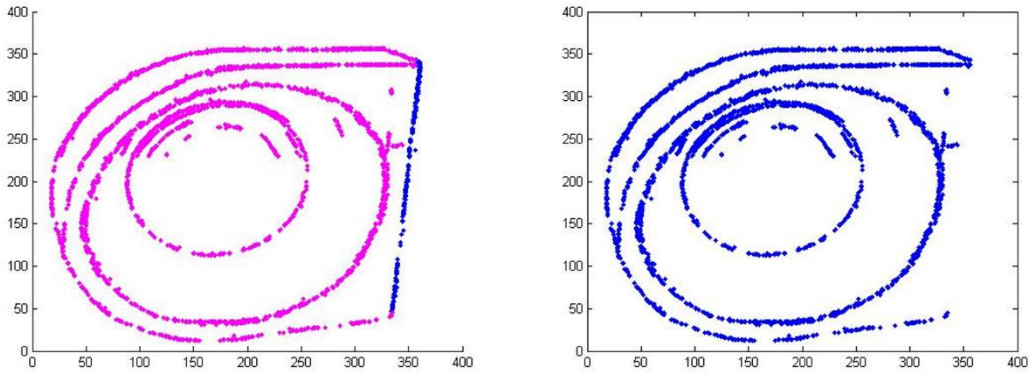


图 5-18 直线检测；（a）蓝线为检测出的直线；（b）为剔除直线后的数据

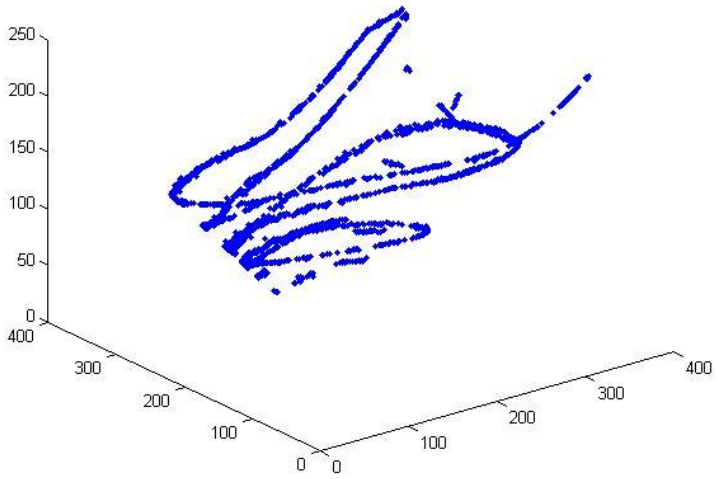


图 5-19 使用平均值-欧式距离扩维法得到的三维数据

我们选择对数据集划分为 5 个类别，最终的分类结果以及 GSMMC 的中间划分图像如图 5-20，5-21，5-22，5-23 所示，分别为两分图，三分图，四分图以及最终的五分图。

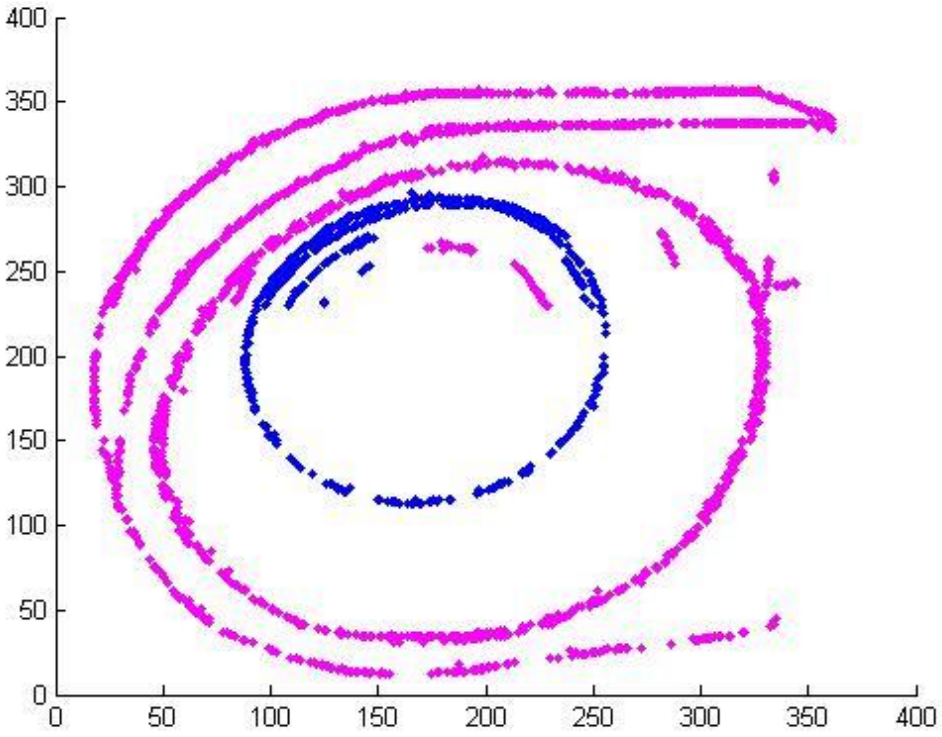


图 5-20 两分图

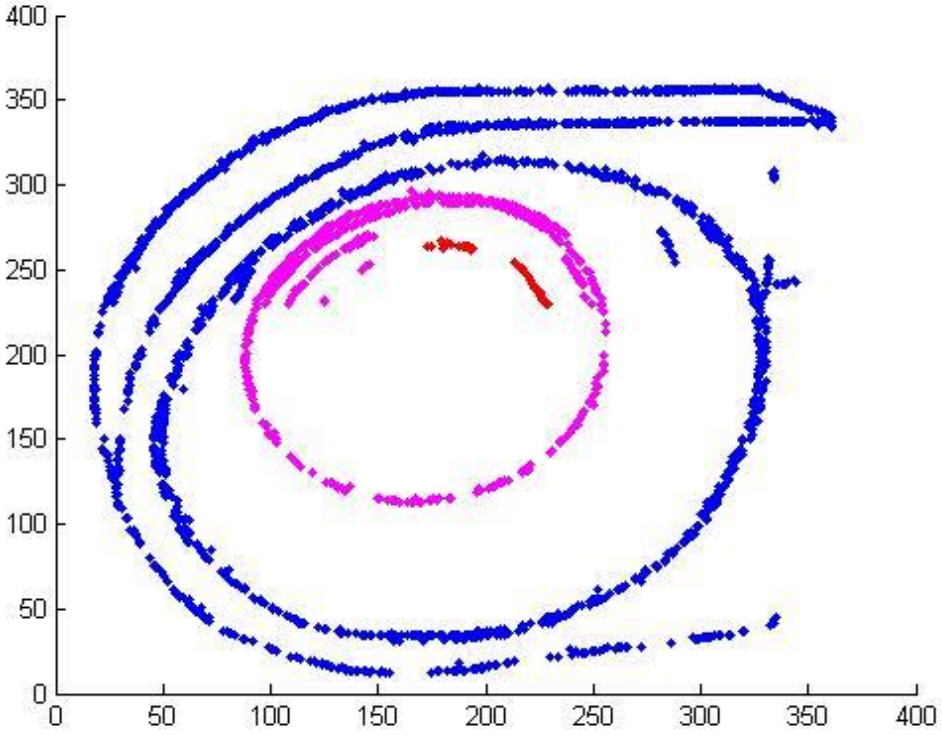


图 5-21 三分图

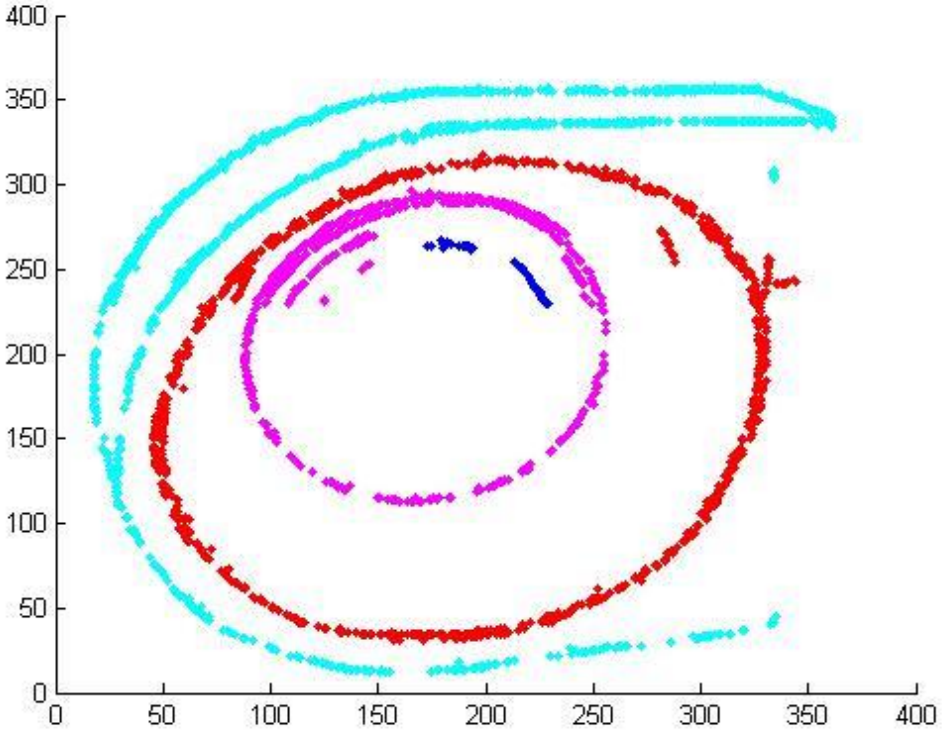


图 5-22 四分图

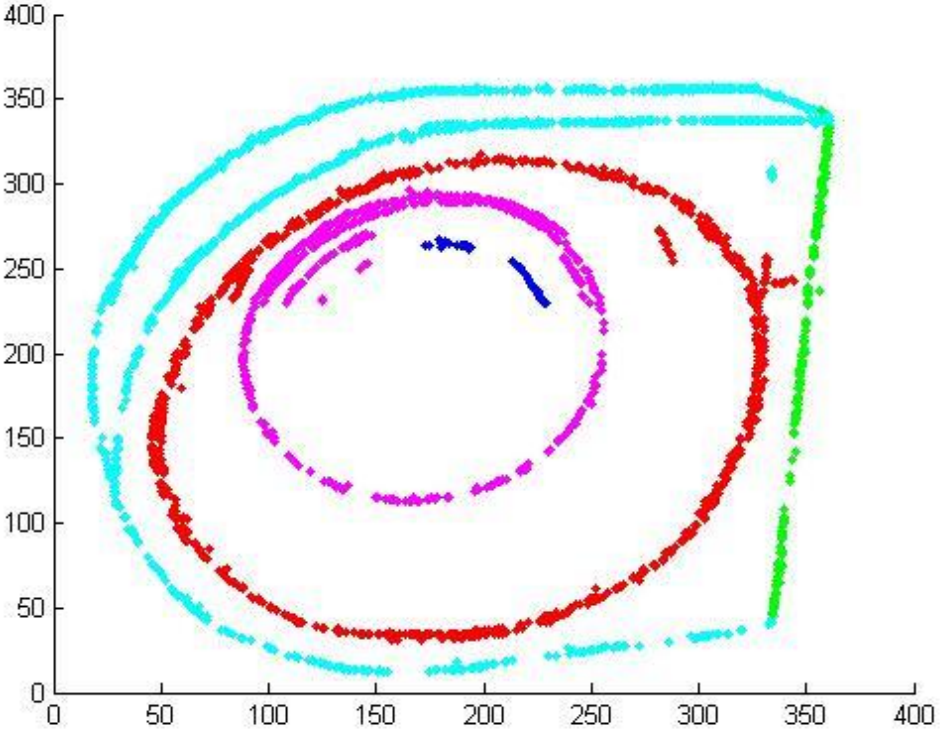


图 5-23 最终结果五分图

六、评价与结论

本次所有的问题都是建立在了子空间聚类和大数据分析的基础之上，已经有许许多多的方法来探讨这个问题。本文选取参考文献[7]和[8]中的算法 SSC 和 SMMC 成功解决了前三个问题。SSC 算法适用于多个线性子空间的聚类;而 SMMC 适用范围更广，不仅能够划分线性子空间，对于非线性子空间的聚类同样有效果。相对的，SSC 的优点同样很明显，那就是参数较少较稳定，面对线性子空间的聚类问题，推荐使用 SSC。

而现有的算法，包括 SSC 和 SMMC，对于第四个问题显得束手无策。于是本文基于 SMMC 和混合多子空间聚类提出了一种创新的算法 GSMMC，提出了化繁为简的思想,通过一次循环多划分一类,逐步递进的原理,成功解决了问题四(a)。

对于最为复杂的问题四（b），在 GSMMC 的基础上，反过来处理数据集，同样的化繁为简：将一条靠近噪声的直线分离并剔除。然后设法扩大不同类别之间的距离，减小相似度，最后成功分理处了第四题图（b）的各个部分。

七、参考文献

[1] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(2):218–233, 2003.
[2]卜德云，张道强．自适应谱聚类算法研究[J]．山东大学学报：工学版，2009, 39(5): 22-26.
[3] J. B. Shi and J. Malik, “Normalized cuts and image segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888–905, 2000
[4] Von Luxburg U. A tutorial on spectral clustering[J]. Statistics and computing, 2007, 17(4): 395-416.
[5] Ng A Y, Jordan M I, Weiss Y. On spectral clustering: Analysis and an algorithm[J]. Advances in neural information processing systems, 2002, 2: 849-856.
[6] G. Chen and G. Lerman, “Spectral curvature clustering (SCC),” International Journal of Computer Vision, vol. 81, no. 3, pp. 317–330, 2009.
[7] 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.
[8] Y. Wang, Y. Jiang, Y. Wu, and Z. Zhou. Spectral clustering on multiple manifolds. IEEE Transactions on Neural Networks, 22(7):1149–1161, 2011.
[9] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” Science, vol. 290, no. 5500, pp. 2323–2326, 2000.
[10] L. K. Saul and S. T. Roweis, “Think globally, fit locally: Unsupervised learning of low dimensional manifolds,” Journal of Machine Learning Research, vol. 4, no. 2, pp. 119–155, 2004.
[11] Z. Y. Zhang and H. Y. Zha, “Principal manifolds and nonlinear dimension reduction via local tangent space alignment,” SIAM J. Scientific Computing, vol. 26, no. 1, pp. 313–338, 2005.
[12] Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." Communications of the ACM 24.6 (1981): 381-395.
[13] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(2):218–233, 2003.
[14] Brown, Matthew, and David G. Lowe. "Automatic panoramic image stitching using invariant features." International journal of computer vision 74.1 (2007): 59-73.

八、附录

以上分割相关的图像均由 Matlab 生成，只要运行 run_smmc_*.m 和 run_SSC_*.m 文件即可运行程序，其中*代表题号，如要运行第一题的 SMMC 算法实现，则只需运行 run_smmc_1.m，或者想运行第四题的（b），只需要运行 run_smmc_4b.m 文件。