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	西安理工大学
参赛队号	10700027
队员姓名	1.张岩
	2.倪超
	3.毛雯

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：

在海量数据不断产生的时代，对于数据的处理有了更高的要求，而流形学习已经成为信息科学邻域研究的热点。流形学习就是从高维采样的数据中找到高维空间中的低维流形，以实现维数简约或者数据可视化。

针对问题一，由于该组数据的子空间相对独立且维数较高，选用基于低秩表示（LRR）的鲁棒性子空间重构算法建立模型，将这组 200 个数据分成了两类，具体分类结果见表 5.1。同时采用传统 PCA+K-means 算法对数据分类，与 LRR 分类结果进行比较，LRR 分类结果明显优于传统 PCA+K-means 方法。

针对问题二，四个低维空间中的子空间聚类问题和多流形聚类问题，（a）两条交点不在原点且互相垂直的两条直线，采用谱多流形聚类算法（SMMC）建立数学模型将其分为两类，效果见图 5.2(a)。（b）为一个平面和两条直线不满足独立子空间的例子，采用稀疏子空间聚类算法模型（SSC）建立数学模型将其分为三类，效果见图 5.3。与谱多流形聚类算法（SMMC）建立模型的分类效果相比更好，说明该算法对非独立子空间的分类能力更强。（c）为两条不相交的曲线，采用谱多流形聚类算法（SMMC）将其分为两类，效果见图 5.2(c)，结果分析说明该算法对噪声的鲁棒性也很好。(d)为两条相交的螺旋线，采用谱多流形聚类算法（SMMC）将其分为两类，效果见图 5.2(d)。

针对问题三，实际应用中的子空间聚类问题，（a）工业测量中的非接触测量方式所用到的特征提取，需要将十字上的点分成两类，采用一维映射中心方法建立数学模型将其分为两类，效果见图 5.6，同时采用谱多流形聚类算法建立数学模型（SMMC）将其分为两类，效果见图 5.7，与一维映射中心模型结果比

较，后者分类效果较好，但是鲁棒性差，前者效果稍差，但是适用性强，可处理所有同类型问题。（b）视频中的运动目标分割，在基于低秩表示（LRR）的鲁棒性子空间重构算法基础上，提出基于低秩表示与轨迹方向特征的鲁棒性子空间重构算法（LRRTrack）的改进算法，建立数学模型并求解，将运动目标的特征点轨迹分为了 3 类，效果见图 5.8。与 LRR 算法比较，改进的算法，对背景点与运动目标点误分类问题处理效果更好，效果见图 5.11。（c)根据人脸数据的高维特性，采用了稀疏子空间聚类的方法（SSC）将 20 张人脸图像数据分为 2 类，见表 5-6，同时采用传统 PCA+K-means 算法分类，并进行比较，结果与 SSC 分类结果相同，见表 5-7。

针对问题四，两个实际应用中的多流形聚类问题，（a）需要将圆台上的点云按照圆台的顶、底、侧面分类，采用谱多流形聚类算法（SMMC）建立数学模型并求解，将其点云分为了三类，效果见图 5.13。（b）需要将机器工件外部边缘轮廓按照轮廓线中不同的直线和圆弧组合分类，采用谱多流形聚类算法（SMMC）建立模型将其分为 3 类，结果见图 5.14。

关键词：流形学习；LRR；SMMC；SSC；LRRTrack；一维映射模型；PCA；

一、问题的重述

1.1 问题的背景

随着信息爆炸时代的到来，数据集增长的更新更快、数据维度更高、非结构化更突出，而技术的落后造成了信息资源的巨大浪费，所以，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，以及图像分割、运动分割等计算机视觉问题（人脸识别、图像分类、运动分割等实例见下文）中。更一般地，对于高维数据的聚类分析等基本问题，结构分析也格外重要。

利用有限高维样本数据进行学习通常会遭遇维数灾难问题，而避免这种问题最常见的方法就是降维。传统的线性子空间方法对非线性结构的数据不能进行有效地处理。高维数据的几何特性使得在低维嵌入流形上建立模型成为可能，予流形方法则是发现高维数据非线性几何结构的有效方法。

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。

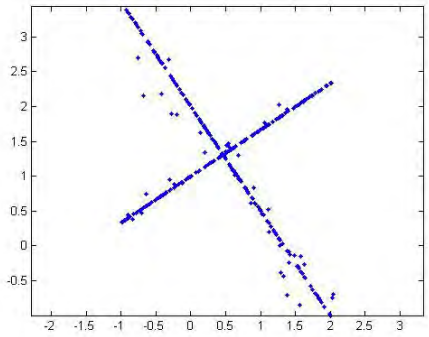
1.当子空间独立时，子空间聚类问题相对容易。附件一中 1.mat 中有一组高维数据，它采样于两个独立的子空间。请将该组数据分成两类。

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。

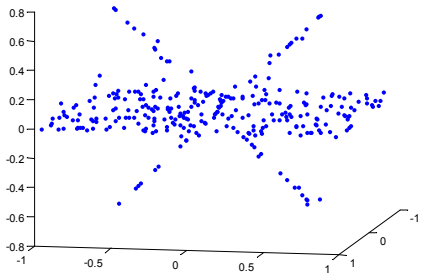
3. 请解决以下三个实际应用中的子空间聚类问题，数据见附件三。

（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 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。

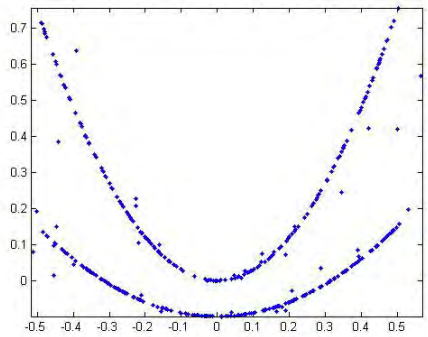
（b）运动分割是将视频中有着不同运动的物体分开，是动态场景的理解和重构中是不可缺少的一步。基于特征点轨迹的方法是重要的一类运动分割方法，该方法首先利用标准的追踪方法提取视频中不同运动物体的特征点轨迹，之后把场景中不同运动对应的不同特征点轨迹分割出来。已经有文献指出同一运动的特征点轨迹在同一个线性流形上。图 1.2（b）显示了视频中的一帧，有三个不同运动的特征点轨迹被提取出来保存在了 3b.mat 文件中，请使用适当方法将这些特征点轨迹分成三类。



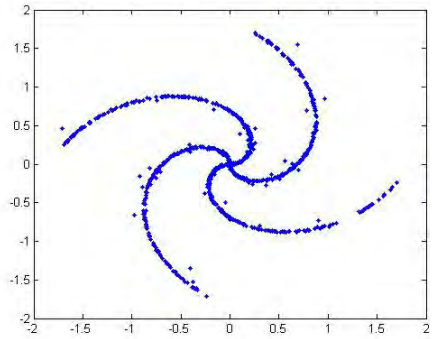
(a)



(b)

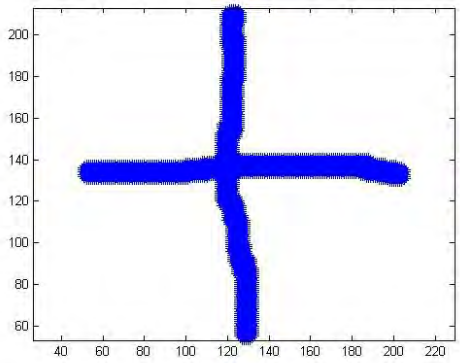


(c)



(d)

图 1.1



(a)

(b)

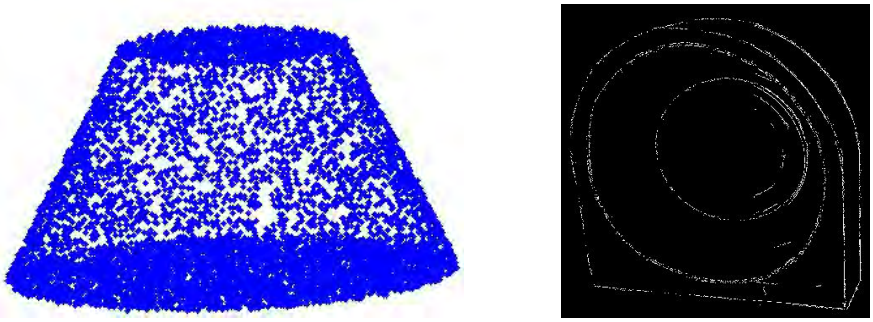
图 1.2

（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、问题的分析

2.1 问题一的分析

已知 1.mat 所提供的数据是采样于两个独立子空间的一组高维数据，要想直接通过传统的分类方法进行子空间的分类是比较难实现的。因此，根据文献[1]中提出的人脸特征在不同光照下都可以被一个低维子空间近似的原理，考虑对 1.mat 中的大量数据进行降维，即构造挖掘数据集的低维线性子空间结构，再依据文献[2]提出的运动分割中的特征点数据具有多个混合子空间的结构，判断哪些特征点属于同一子空间，然后通过子空间聚类，将来自同一子空间中的数据归为一类。

为此，我们需要从给定的多维子空间集合中抽取其中若干维作为数据样本（向量），以使每个样本聚类到到各自的子空间，并剔除可能的异常值。首先，我们考虑使用传统[3]PCA+K-means 算法进行聚类，再根据提出的 LRR[4]新的目标函数算法，寻找能够代表数据样本的低秩，在给定样本下作为基的线性组合进行聚类，最后进行比较验证。

2.2 问题二的分析

针对问题二，要求处理第 2 题数据中四个低微空间中的子空间聚类问题，分析图 1.1 中给出的四幅图，其中包含了流形良分离和明显交错的情况，文献[5]中表明，传统的谱聚类算法对两性分离的流形结构会给出完全正确的聚类结果，而传统的谱聚类不能很好地分割出交叠的流形结构，从而提出了非对称型规范化谱聚类方法，但是它还是会出现错误信息并严重影响分类结果，因此考虑到需要构造新的相似性矩阵以使得它具有所期望的性质：来自不同流形结构的数据点之间有相对低的相似性权值。

基于以上理论，[6]提出了谱多流形聚类算法（SMMC）来实现混合流形聚类，它的基本思想是：从相似性矩阵的角度出发，充分利用流形采样点所内含的自然的局部几何结构信息来辅助构造更合适的相似性矩阵并进而发现正确的流形聚类。

2.3 问题三的分析

1）针对（a）问，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，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。根据题目所提供的数据 3a.mat，为一个高维的二维点集，横竖两条线可以理解为他们分属于不同的流形，那么问题就转化为混合流形的分类问题。因此，我们采用谱多流形聚类算法（SMMC），将两个混合流形的数据点分为两类，尤其是交叉部分的点能够合理的分属所属的流形空间。

2）针对（b）问，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，根据场景中不同运动对用不同的特征点轨迹分割出来。由于同一运动的特征点轨迹在同一个线性流形上，那么就可以将问题转化为多线性流形分类的问题。如图 1.2（b）所示，是视频中的一帧，有三个不同的运动物体，题目所提供的数据 3b.mat 中保存了三个不同运动的特征点轨迹的数据，我们采用基于低秩表示（LRR）的鲁棒性子空间重构算法，就可以将三个在分别属于不同的线性子空间的运动物体分开，从而问题得以解决。

3）针对（c）问所提供的数据 3c.mat, 数据矩阵的每一个列向量表示一副人脸图像，它是将一副人脸图像的灰度矩阵拉成了一个高维列向量，其维度达到了 2016。在如此高维的数据上进行分类处理难度很大，所以这里想到了稀疏表示[7]（SR）的思想。已经证明了在不同的光照或表情变换条件下的人脸图像可以用一个低维子空间来近似，取自多个人的一组人脸图像可以看作是 9 维线性子空间的并[1]，从而人脸识别问题等价于子空间聚类问题. 。根据稀疏表示可以将人脸图像的高维列向量进行降维，但并不会丢失特征信息，再对降维后的数据进行分类，这样降大大减少计算的复杂度。针对降维后数据的分类，由于只有 20 个人脸数据，需要将这 20 个人脸数据分为两类，那么谱聚类是一种很好的解决方法。通过以上的分析，那么[8]理论上可以将问题很好的解决。

2.4 问题四的分析

1）针对（a）题，如图 1.3（a）所示，分别显示了圆台的点云，要求将数据点所在的面分开（即按照圆台的顶、底、侧面分成三类）。可以将圆台的面理解为流形，那么数据分别采样于圆台的顶，底和侧面不同流形，问题就转化为将数据点所在的流形进行分类，即数据的混合多流形问题。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。因此，可以采用谱多流形聚类算法（SMMC）。

2）针对（b）题，如图 1.3（b）所示，是机器工件外部边缘轮廓的图像，题目要求将机器工件外部边缘轮廓的图像根据轮廓线中不同的直线和圆弧的分类，类数自定，那么此问题属于多流形聚类的问题。针对此类问题，采用谱多流形聚类算法（SMMC）是能够解决的。然后根据直线的长短，以及圆弧的长短和圆的半径分类，可以将轮廓线分为 5 类。

三、模型的假设

（1）假设数据的结构为混合多流形更具有一般性；
（2）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；
（3）假设数据均匀采样于一个高维欧式空间中的低维流形；
（4）假设在谱多流形聚类算法中，每个数据点的局部切空间是已知的。
（5）假设在稀疏子空间聚类算法中，稀疏奇异值矩阵也是稀疏的，噪声的误差可用 $F-$ 范数进行衡量；

四、符号说明

符号	符号说明
$\lvert Z\rvert$	对矩阵 Z 的元素取绝对值得到的矩阵
S_α	第 α 个子空间
x_i， y_i	输入数据矩阵
c_j	第 j 个类的类中心
W	相似矩阵
C	稀疏系数矩阵
E	稀疏奇异值矩阵
Z	系数矩阵
M	流形的局部线性子模型的个数
p_{ij}	局部切空间之间的结构相似性
W_{ij}	相似性权值
$u_1,u_2,\cdots,u_k$	k 个特征值所对应的特征向量
D	对角矩阵
C_m	模型的协方差

五、模型的建立与求解

5.1 针对问题一

5.1.1 模型的建立

模型一：LRR 算法建立模型

低秩表示是一种低秩子空间分割的方法。低秩子空间分割的基本思想是利用数据集自身作为字典对数据集进行低秩表示，得到低秩表示系数矩阵，然后把低秩表示系数矩阵看做是相似性矩阵，用谱聚类方法进行子空间分割。低秩表示算法的目标是寻找到一种字典里所有基的线性组合的最低秩的表示。

假设给定的数据集为 $X=[X_1,X_2,\cdots,X_n]\in R^{d\times n}$，数据维度是 D，每一列向量是一个样本点，n 表示样本个数，则每个样本 $X_i\in R^d$ 都可以表示为“字典” $A=[a_1,a_2,\cdots,a_m]$ 中的基的线性组合：

$$X=AZ \tag{5-1}$$

其中，$Z=[Z_1,Z_2,\cdots,Z_m]$ 是一个系数矩阵，它的向量 Z_i 代表样本 X_i。

通过求解凸优化问题来获取数据集 X 的系数矩阵 Z：

$$\begin{aligned}&\min_{z}\|Z\|_*\\&\text{s.t.}\quad X=AZ\end{aligned} \tag{5-2}$$

其中，$\|Z\|_*$ 表示矩阵 Z 的核范数，即矩阵的奇异值之和。

低秩表示的目标函数为：

$$\begin{aligned}&\min_{Z,E}\|Z\|_*+\lambda\|E\|_{2,1}\\&\text{s.t.}\quad X=AZ+E\end{aligned} \tag{5-3}$$

其中，用数据 X 自身做字典，$\|\bullet\|_*$ 表示矩阵的核范数，用核范数可以避免计算矩阵秩的时候出现的 NP 难的问题。$\|E\|_{2,1}=\sum_{j=1}^{n}\sqrt{\sum_{i=1}^{n}([E_{ij}])^2}$ 称为 $l_{2,1}$ 范数，E_{ij} 为矩阵 E 的 (i,j) 位置的元素。

假定 E^* 为式（5-3）对应于 E 的最优解，为得到每一个特征点的显著性指标 $S(x_i)$，需做如下处理

$$S(x_i) = \| E(:,i) \|_2 = \sqrt{\sum_j E_{ij}^2} \tag{5-4}$$

$\| E(:,i) \|_2$ 为矩阵 E 的第 i 列的 l_2 范数。$S(x_i)$ 值越大，说明特征点的显著性越强。由此，显著性检测就转换为式（5-4）的一个低秩逼近问题。

而参数 $\lambda > 0$ 用来平衡两个范数部分的效应，它可以根据两个范数的性质来选择，也可以凭经验确定。

算法步骤：

（1）将给定的数据集分割成 N 个块，提取每一个块的 D 维特征值 x^i，生成一个 $D \times N$ 大小的特征特征矩阵 X;

（2）优化式（5-4）解得稀疏矩阵 E^*；

式（5-4）的优化过程可简单地采用文献[6]提出的方法。通过增广拉格朗日（ALM）将约束问题转化为无约束问题。式（5-4）等价于

$$\begin{aligned} &\min_{Z,E,J} \| J \|_* + \lambda \| E \|_{2,1} \\ &\text{s.t.} \quad X = XZ + E, \quad Z = J \end{aligned} \tag{5-5}$$

用增广拉格朗日方法求解式（5-5）

$$\begin{aligned} L = &\| J \|_* + \lambda \| E \|_{2,1} + \langle Y, X - AZ - E \rangle + \langle W, Z - J \rangle \\ &+ \frac{\mu}{2} \| X - AZ - E \|_F^2 + \frac{\mu}{2} \| Z - J \|_F^2 \end{aligned} \tag{5-6}$$

其中，Y，W 是拉格朗日乘子；u 是罚参数。

式（5-6）可由交替方向（ADM）求解，交替方向方法如下：

步骤 1：固定其他变量更新 J:

$$J = \arg\min \frac{1}{\mu} \|J\|_* + \frac{1}{2} \|J - (Z + W/\mu)\|_F^2 \tag{5-7}$$

步骤 2：固定其他变量更新：

$$Z = (I + A^T A)^{-1} (A^T (X - E) + J + (A^T Y - W)/\mu) \tag{5-8}$$

步骤 3：固定其他变量更新 E:

$$E = \arg\min \frac{\lambda}{\mu} \|E\|_{21} + \frac{1}{2} \|E - (X - AZ + Y/\mu)\|_F^2 \tag{5-9}$$

步骤 4：更新拉格朗日乘子：

$$Y = Y + \mu(X - AZ - E) \tag{5-10}$$

$$Y1 = Y1 + \mu(X - AZ - E)$$

步骤 5 ：更新罚因子：

$$\mu = \min(\rho\mu, 10^{10}) \tag{5-11}$$

ρ 用于控制收敛速度，试验选取 $\rho = 1.1$。

步骤 6 ：检查收敛条件： $X - XZ - E \to 0, Z - J \to 0$。

结束，并输出最优解 E^*。

模型二：PCA 算法建立模型

主成分分析（PCA）是一种常用的基于变量协方差矩阵对信息处理、压缩和抽提的有效方法。PCA 方法的基本原理是利用 K-L 变换抽取高维数组的主要成分，构成特征空间，通过投影变换实现对高维数据的降维处理。通过该方法可以得到一个降维后的矩阵，求解原始矩阵每一列与特征间的欧氏距离，找出相差最大的两个值，将其作为聚类时的两个中心点。

设 $X_1, X_2, \cdots, X_p$ 表示以 $x_1, x_2, \cdots, x_p$ 为样本观测值的随机变量，如果能找到 $c_1, c_2, \cdots, c_p$，使得

$$Var\left(c_1 X_1 + c_2 X_2 + \cdots + c_p X_p\right) \tag{5-12}$$

的值达到最大，则由于方差反应了数据变异的程度，也就表面我们抓住了这 p 个变量的最大变异。当然式（5-12）必须加上某种限制，否则权重可选择无穷大而没有意义，通常规定

$$c_1^2 + c_2^2 + \cdots + c_p^2 = 1 \tag{5-13}$$

在此约束下，求式（5-13）的最优解。这个解是 $p-$维空间的一个单位向量，它代表一个“方向”，就是常说的主成分方向。

一个主成分不足以代表原来的 p 个变量，因此需要寻找第二个乃至第三个，第四个主成分，第二个主成分不应该再包含第一个主成分的信息，统计上的描述就是让这两个主成分的协方差为 0，几何上就是这两个主成分的方向正交。具体确定各个主成分的方法如下。

设 Z_i 表示第 i 个主成分， $i = 1,2,\cdots, p,$ 可设

$$\begin{cases} Z_1 = c_{11} X_1 + c_{12} X_2 + \cdots + c_{1p} X_p, \\ Z_2 = c_{21} X_1 + c_{22} X_2 + \cdots + c_{2p} X_p, \\ \vdots \\ Z_p = c_{p1} X_1 + c_{p2} X_2 + \cdots + c_{pp} X_p, \end{cases} \tag{5-14}$$

其中：对每一个 i，均有 $c_{i1}^2 + c_{i2}^2 + \cdots + c_{ip}^2 = 1$，且 $\left[c_{11}, c_{12}, \cdots, c_{1p}\right]$ 使得 $Var\left(Z_1\right)$ 的值达到最大； $\left[c_{21}, c_{22}, \cdots, c_{2p}\right]$ 不仅垂直于 $\left[c_{11}, c_{12}, \cdots, c_{1p}\right]$，而且使 $Var\left(Z_2\right)$ 的值达到

最大；$[c_{31},c_{32},\cdots,c_{3p}]$同时垂直于$[c_{11},c_{12},\cdots,c_{1p}]$和$[c_{21},c_{22},\cdots,c_{2p}]$，并使$Var(Z_3)$的值达到最大；以此类推可得全部 p 个主成分。

在得到降维的矩阵后，模型一与模型二都采用 K-means 聚类算法，将目标矩阵分为两类。

K-means 算法是典型的基于原型的目标函数聚类方法，它是数据点到原型的某种距离作为优化的目标函数，利用函数求极值的方法得到迭代运算的调整规则。K-means 算法以欧式距离作为相似度测度，它是求对应某一初始聚类中心向量最优分类，使得评价指标最小。算法采用误差平方和准则函数作为聚类准则函数。

5.1.2 模型的求解

模型一求解：

在编程过程中我们涉及到了 LRR 算法中的参数：平衡因子$\lambda=0.18$，对原始数据运用 LRR 算法求解后得到 1 类与 2 类分别各有 100 个数据点，具体分类结果见表 5-1。

表 5-1　LRR 算法分类结果

标号	类别																			
1-20	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
21-40	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
41-60	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
61-80	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
81-100	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
101-120	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
121-140	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
141-160	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
161-180	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
181-200	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

模型二求解：

在对原始数据降维时，根据要求分别选取了贡献率 p 分别为 85%，95%和 99%的 3 种情况，得到降维后矩阵尺寸及两类分类结果见表 5-1：

表 5-1　PCA+K-means 分类结果

p	降维后尺寸	1 类数目	2 类数目
85%	10×200	192	8
95%	19×200	192	8
99%	26×200	192	8

通过模型求解得到，无论贡献率 P 为所示三种情况的任一种情况，分类结果都相同，具体分类结果见附录一中表 1。在不影响结果的前提下，要求复杂度越小越好，计算量越少越好，因此最终选取 p=85%为指标。

5.1.3 模型的分析

本节算法的源程序采用 C++与 Matlab 语言编写，实验平台为：Intel（R）Core(TM)，2.53GHz i3CPU 和 6G 内存。

从分类的结果上分析，传统的 PCA 算法分类效果非常差，一类 192，另一类 8 个，而提出的 LRR 算法结果有明显的改善，两类均为 100 个，可见 LRR 算法的低秩表示方法对噪声的鲁棒性更强。

5.2 针对问题二

5.2.1 模型的建立

模型一：SMMC 算法模型

规定 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。首先训练 M 个混合概率主成分分析器（MPPCA）来估计局部切空间，其中每个分析器由模型参数 $\theta_m=\{\mu_m,V_m,\sigma_m^2\}$（$m=1,2,\cdots,M$）刻画，其中 $\mu_m\in R^D,V_m\in R^{D\times d}$，$\sigma_m^2$ 是一个标量。在第 m 个分析器模型下，一个 D 维的观测数据向量 X 通过下式对应一个相应的 d 维潜在向量 y：

$$X=V_m y+u_m+\varepsilon_m \tag{5-15}$$

其中，μ_m 是数据的均值向量，潜在变量 y 和噪声 ε_m 分别是高斯分布 $y\sim \mathrm{N}(0,1)$ 和 $\varepsilon_m\sim \mathrm{N}(0,\delta_m^2 I)$。在此模型下，$X$ 的边际分布为：

$$p(X\mid m)=(2\pi)^{-D/2}\exp\{-\frac{1}{2}(X-\mu_m)^T C_m^{-1}(X-\mu_m)\} \tag{5-16}$$

其中模型协方差为：

$$C_m=\sigma_m^2 I+V_m V_m^{\mathrm{T}} \tag{5-17}$$

模型参数 μ_m，V_m 和 σ_m^2 可以通过利用 EM 算法最大化观测数据

$\chi=\{X_i, i=1,2,\cdots,N\}$ 的对数似然来得到：

$$\iota=\sum_{i=1}^{n}\ln\{\sum_{m=1}^{M}\pi_m p(X_i\mid m)\} \tag{5-18}$$

其中 π_m 是混合比例，满足条件 $\pi_m\geq 0$ 和 $\sum_{m=1}^{M}\pi_m=1$。

该模型采用 K-means 聚类算法来初始化 EM 学习过程。最后，样本点 X_i 根据下述关系分组到第 j 个局部分析器：

$$p(X_i\mid j)=\max_m p(X_i\mid m) \tag{5-19}$$

同时其局部切空间由下式给出：

$$\Theta_i=span(V_j) \tag{5-20}$$

则两个数据点 X_i 和 X_j 的局部切空间之间的结构相似性可以定义为：

$$p_{ij}=p(\Theta_i,\Theta_j)=\left(\prod_{l=1}^{d}\cos(\theta_l)\right)^{o} \tag{5-21}$$

在式（5-21）中，$o\in N^{+}$ 是可调节的参数。$0\leq\theta_1\leq,\cdots,\leq\theta_d\leq\pi/2$ 是两个切空间 Θ_i 和 Θ_j 之间的主角度，递归地定义为：

$$\cos(\theta_1)=\max_{\substack{u_1\in\Theta_i,v_1\in\Theta_j\\ \|u_1\|=\|v_1\|=1}} u_1^{\mathrm{T}}v_1 \tag{5-22}$$

$$\cos(\theta_1)=\max_{\substack{u_l\in\Theta_i,v_l\in\Theta_j\\ \|u_l\|=\|v_l\|=1}} u_l^{\mathrm{T}}v_l,\quad l=1,2,\cdots,d \tag{5-23}$$

其中 $u_l^{\mathrm{T}}u_i=0$，$v_l^{\mathrm{T}}v_i=0$，$i=1,2,\cdots,l-1$。

根据点 X_i 和 X_j 之间的局部相似性简单地定义为：

$$q_{ij}=\begin{cases}1 & X_i\in Knn(X_j)\ or\quad X_j\in Knn(X_i)\\ 0 & otherwise\end{cases} \tag{5-24}$$

其中 $Knn(X)$ 代表 X 的 K 个近邻数据点。

最后函数 f 将这两个函数 p 和 q 简单的乘在一起得到相似性权值：

$$w_{ij}=p_{ij}q_{ij}=\begin{cases}(\prod_{l=1}^{d}\cos(\theta_l))^{o} & X_i\in Knn(X_j)\ or\quad X_j\in Knn(X_i)\\ 0 & otherwise\end{cases} \tag{5-25}$$

式（5-25）中定义的相似性权值具有来自不同聚类或流形的数据点之间有

相对较低的权值。其原因是：（a）.当来自不同流形的两个数据点相互远离时，根据式（5-25）其相似性权值为 0；（b）.当来自不同流形的两个数据点靠近流形的相交区域时，它们有不同的局部切空间结构，因此当调节参数 o 足够大时，也可以使得相似性权值相对低。

由此构造出相似性矩阵 $W \in R^{N\times N}$，并计算出对角矩阵 D，其中 $d_{ii}=\sum_j w_{ij}$。

构造广义特征矩阵

$$(D-W)u=\lambda Du \tag{5-26}$$

求解式（5-26），找出最小 k 个特征值对应的特征向量 $(u_1,u_2,\cdots,u_k)$，最后利用 K-means 算法将 $U=[u_1,u_1,\cdots,u_k]\in R^{N\times K}$ 的行向量分组为 k 个聚类。

模型二：SSC 算法模型

稀疏子空间聚类的基本思想是：将数据 $x_i \in S_\alpha$ 表示为所有其他数据的线性组合

$$x_i=\sum_{j\neq i} Z_{ij}x_j \tag{5-27}$$

并对表示系数施加一定的约束使得在一定条件下对所有的 $x_j \notin S_\alpha$，对应的 $Z_{ij}=0$。将所有数据及其表示系数按一定方式排成矩阵，则式（5-27）等价于

$$X=XZ \tag{5-28}$$

且系数矩阵 $Z\in R^{N\times N}$ 满足：当 x_i 和 x_j 属于不同的子空间时，有 $Z_{ij}=0$。不同于用一组基或字典表示数据，式（5-28）用数据集本身表示数据，成为数据的自表示。若已知数据的子空间结构，并将数据按类别逐列排放，则在一定条件下可使系数矩阵 Z 具有块对角结构，即

$$Z=\begin{bmatrix} z_1 & 0 & \cdots & 0 \\ 0 & z_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & z_k \end{bmatrix} \tag{5-29}$$

这里 $Z_\alpha(\alpha=1,\cdots,k)$ 表示子空间 S_α 中数据的表示系数矩阵；反之，若 Z 具有块对角结构，这种结构揭示了数据的子空间结构。稀疏子空间聚类就是通过对系数矩阵 Z 采用不同的稀疏约束，使其尽可能具有理想结构，从而实现子空间聚类。

通常，稀疏子空间聚类基于系数向量的一维稀疏性或系数矩阵的二维稀疏性来建立高维数据在低维子空间的表示，利用表示系数矩阵 Z 构造数据的相似度矩阵 $W=\left(|Z|+|Z^T|\right)/2$，最后利用谱聚类算法如 Ncut[9]得到最终的聚类结果。

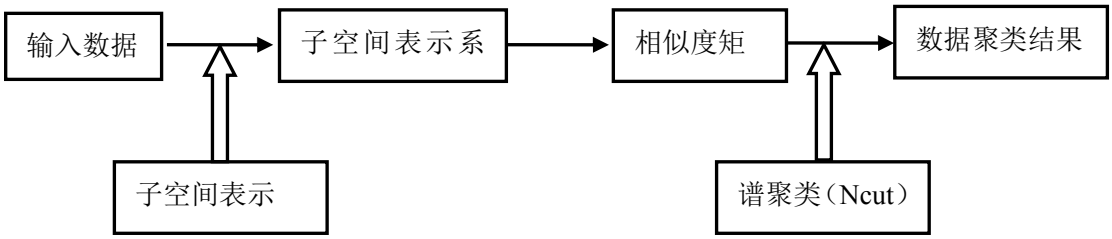


图 5.1　稀疏子空间聚类的基本框架

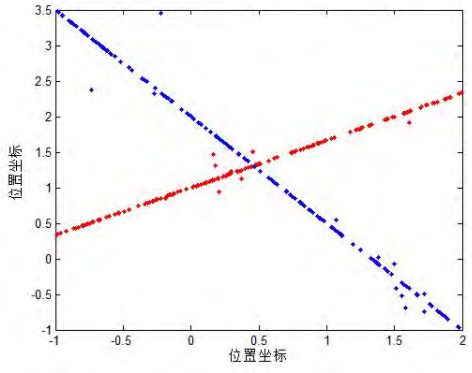
5.2.2 模型的求解

编程过程中我们涉及到了谱多流形聚类（SMMC）模型中的几个参数，包括：局部化模型个数 M，近邻点数 K，以及调节参数 o（其中聚类数 k 和流形维数 d 根据问题为定值），见表 5-3，求解 SMMC 模型，得到结果，效果见图 5.2 中（a）、（b）、（c），具体分类结果见附录一中表 2，表 3，表 5，表 6。

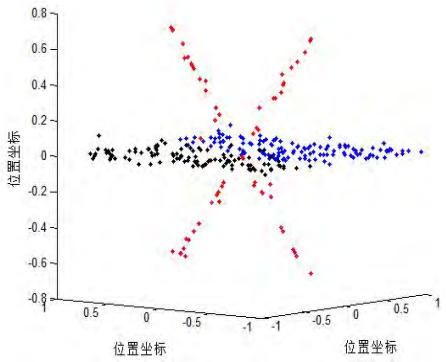
表 5-3　SMMC 模型参数

标号	调节参数 o	近邻点数 n	局部化模型 M	聚类数 k	流形维数 d
（a）	12	$4*\log(N)$	6	2	1
（b）	12	$4*\log(N)$	9	3	2
（c）	5	$4*\log(N)$	20	2	1
（d）	2	$4*\log(N)$	32	2	1

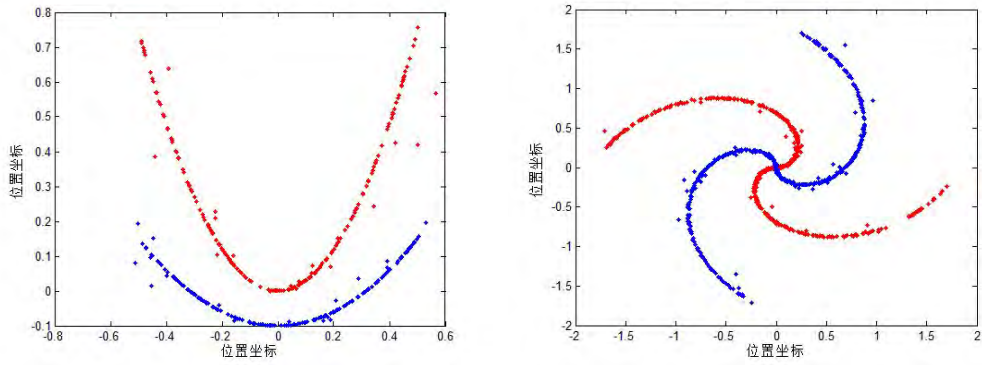
由图 5.2 可以明显看到，对于其中的（b）的分类结果明显有误，通过重新建模后，采用 SSC 算法建立模型进行分类，在编程过程中我们涉及到了 SSC 算法中的参数设定为：$\rho=1,\alpha=20,\gamma=0$，得到的结果效果如图 5.3 所示，具体分类结果见附录一中表 4。



（a）



（b）



（c）　　（d）

图 5.2　SMMC 分类结果

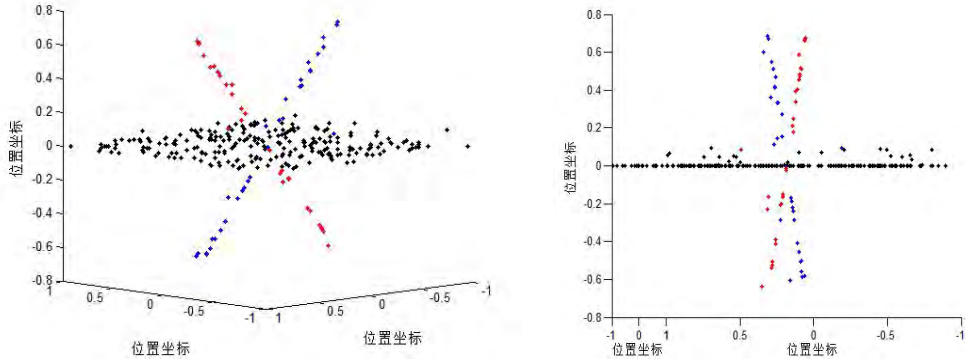


图 5.3　SSC 对（b）分类结果

5.2.3 模型的分析

本节算法的源程序采用 Matlab 语言编写，实验平台为：Intel（R）Core(TM)，2.53GHz i3CPU 和 6G 内存。

(a)，(c)，(d)，分别为线性流形子空间或者良性可分非线性子空间，(b)属于交叠流形子空间例子。从分类结果可以看到，SMMC 算法对前两种子空间具有很好的分类能力，但是对重叠子空间效果较差。针对该类问题问题，SSC 算法模型具有很好的解决能力。

5.3 针对问题三

5.3.1 模型的建立

（1）针对十字的分类问题：

模型一：一维映射中心算法建立模型

将二维的数据分别投影到水平和垂直方向上，在水平方向上，找到投影中最大的前 5 个值，求其均值，即为贯穿十字中心的垂线的纵坐标，同理，在垂直方向上，找到投影中最大的前 5 个值，求其均值，即为贯穿十字中心的垂线

的横坐标。最后运用 1-范数进行相似度测量，得出分类结果。

模型二：SMMC 算法模型

即 5.2.1 中 SMMC 算法建立的模型。

（2）针对运动分割问题：

模型三：LRRTrack 运动分割算法建立模型

利用 5.1.1 中的 LRR 算法模型进行聚类，将背景点、公交车和出租车三类分割开。由于背景点与运动车辆之间特征点位置坐标有重叠部分，所以会有误分现象产生。因为背景点与运动车辆之间运动轨迹 $Tr(n),(n=1,2,...N)$ 不同，所以其运动方向具有很好的可分性，特征点运动方向即为轨迹的斜率：

$$k_m(n)=(Tr(n+1).y-Tr(n).y)/(Tr(n+1).x-Tr(n).x) \quad (5\text{-}30)$$

$$n=1,2,...,(N-1)$$

其中，$Tr(n+1).y$ 与 $Tr(n+1).x$ 分别表示第 $n+1$ 帧特征点位置坐标，$Tr(n).y$ 与 $Tr(n).x$ 分别表示第 n 帧特征点位置坐标，N 表示总共的帧序列数目，本实验数据中 $N=31$。

判断同一帧内任意两特征点之间的方向距离 D_{Tr}：

$$D_{Tr(a,b)}(n)=(k_a(n)-k_b(n))^2,\ \ a,b\in m \quad (5\text{-}31)$$

如果 $D_{Tr}\le TH$，则认为该两个特征点运动方向一致，本实验数据中 $TH=4$。当在 N 总帧中，两特征点运动方向有 $M\ge 80\%N$ 帧运动方向一致，则就认为该两个特征点属于同一运动目标。

（3）针对人脸识别问题：

模型四：SSC 算法建立模型

即 5.2.1 中 SSC 算法建立的模型。

模型五：PCA 算法建立模型

即 5.1.1 中 PCA 算法建立的模型。

5.3.2 模型的求解

1）模型一求解：

针对十字的分类问题，根据一维映射中心模型，最终得到水平和垂直方向的投影如图 5.4 所示。

如图 5.4 所示可以得到投影后的 5 个最大值，求其的均值，得到了贯穿十字中心的垂线的纵坐标为 122，同理，横坐标为 135，即找到十字的中心点为（135,122），同样可以得到其画出的直线，见图 5.5，最后运用 1-范数将聚为两类，第一类有 1428 个，第二类有 1407 个，结果见图 5.6。

（a）水平投影　　（b）水平投影

图 5.4　十字水平与垂直投影

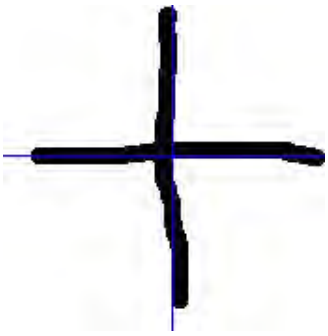


图 5.5 贯穿十字中心的垂线与水平线

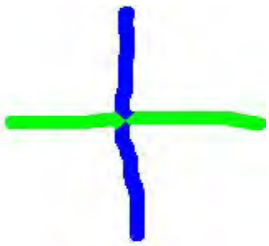


图 5.6 一维映射中心算法模型分类结果

2）模型二求解：

针对十字的分类问题，也可以采用 SMMC 算法模型，聚类效果见图 5.7，具体分类结果见附录一中表 7。

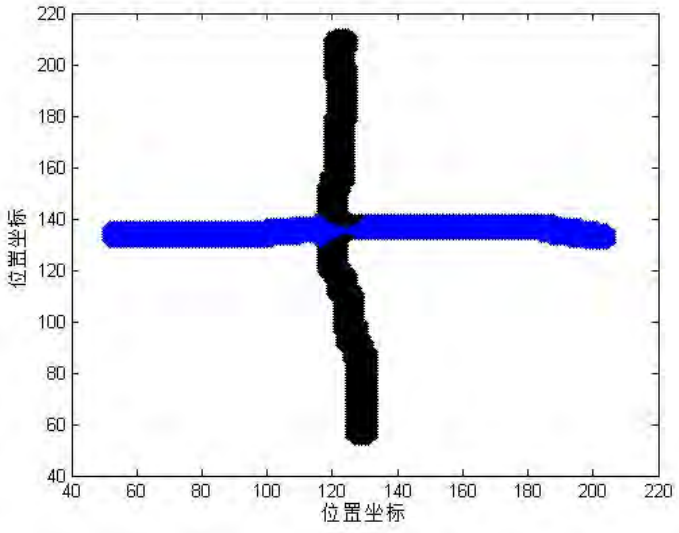


图 5.7 SMMC 分类结果

3）模型三求解：

在编程过程中我们涉及到了 LRR 算法中的参数：平衡因子 $\lambda = 4$，在对运动分割的问题求解时，首先利用 LRR 算法进行分类，分类结果效果见图 5.8。具体分类结果见表 5-4。

表 5-4 LRR 算法模型分类结果

标号	类别																			
1-20	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
21-40	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2
41-60	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
61-80	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
81-100	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
101-120	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
121-140	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
141-160	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
161-180	1	2	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
181-200	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
201-220	1	1	1	1	1	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3
221-240	3	3	3	3	3	3	3	3	3	1	3	1	3	3	3	3	1	3	1	3
241-260	3	3	3	3	3	3	3	3	3	3	1	1	3	3	3	3	3	3	3	3
161-280	3	3	3	3	3	3	3	1	3	1	3	1	3	3	3	3	3	3	3	3
281-200	3	1	3	3	3	3	1	3	3	3	3	3	3	3	3	3	3			

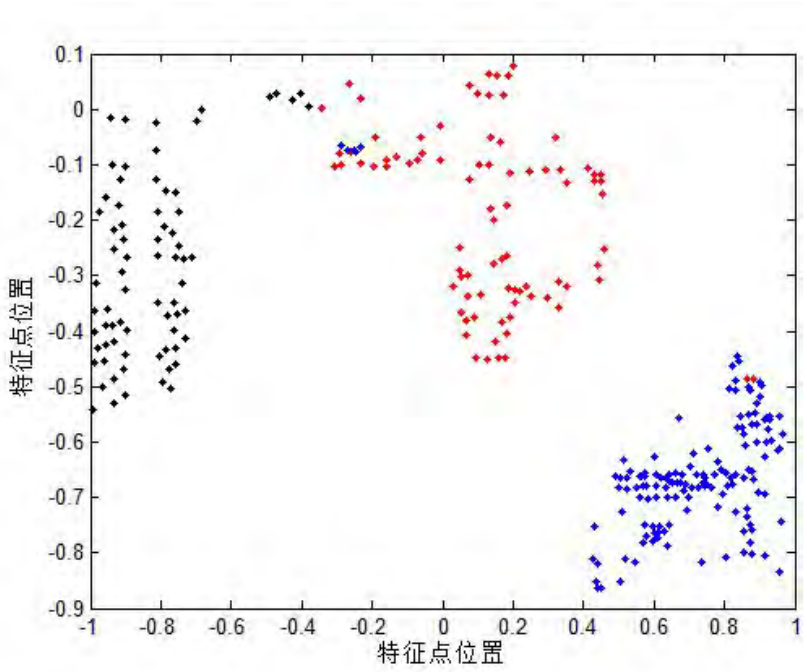
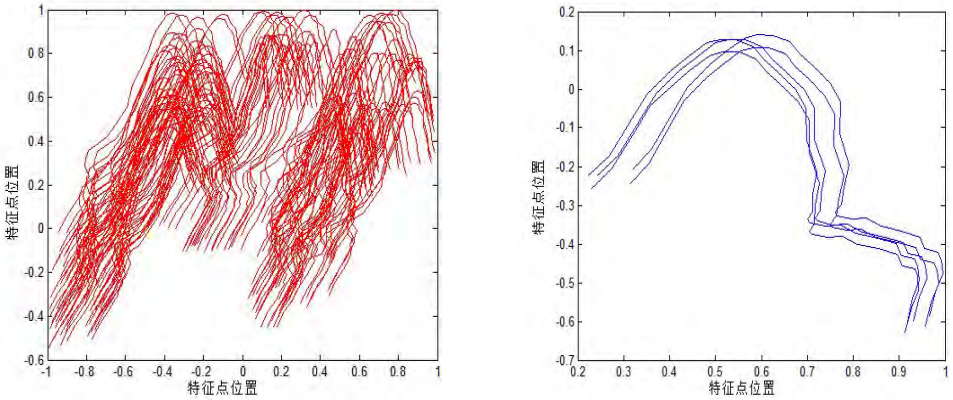


图 5.8　LRR 分类结果

黑色点团表示第 1 类，红色点团表示第 2 类，蓝色点团表示第 3 类，由结果图可看出，2 类中有 3 个 3 类数据误分，3 类中有 2 个 2 类数据误分，且 2 类中还有若干 1 类背景点误分。

画出 3 类目标轨迹，如图 5.9 所示，（a）图表示第 2 类与第 3 类车的运动轨迹，而（b）图表示第 1 类背景点的轨迹，因为数据的来源是在运动场景中得

到的，所以背景点的轨迹变化可能是由于摄像机的晃动造成的。从中可以看出1类与2、3类的运动轨迹有很好的可分性。



（a）第2类与第3类运动轨迹图　　（b）第1类轨迹图

图5.9　三类目标轨迹

每条运动轨迹位置坐标后一帧与前一帧做差，可以得到运动轨迹的斜率，即运动方向，见图5.10，3类中运动轨迹坐标能够相互交叉，但是同一类运动目标特征点的运动方向不会发生突变。从图5.10中也可以验证该理论，从图中明显的可以看出，第2类和第3类的轨迹斜率基本相同，都与第1类有明显的差别，所以，根据判定原理，在同一帧的任意两特征点轨迹斜的欧氏距离 E 小于经验值4时，就认为当前运动方向一致。如果该两个特征点80%以上的帧中满足该条件，就认为这两个特征点属于同一运动目标。

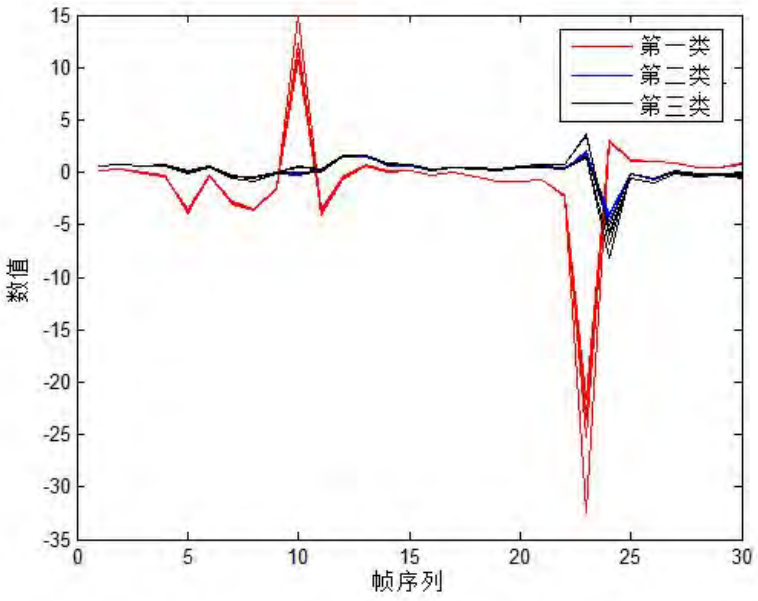


图5.10　第一类、第二类与第三类轨迹斜率

运用以上阐述算法就可以将背景点与运动目标区分清楚。另外，同一运动目标的特征点会集中到一块区域，彼此间不会特别分散太开，利用该原理接着再对2类与3类初始帧位置坐标进行简单K-means聚类，即可解决2、3类中的

误分问题。最终聚类结果效果见图 5.11，具体分类结果见表 5-5。

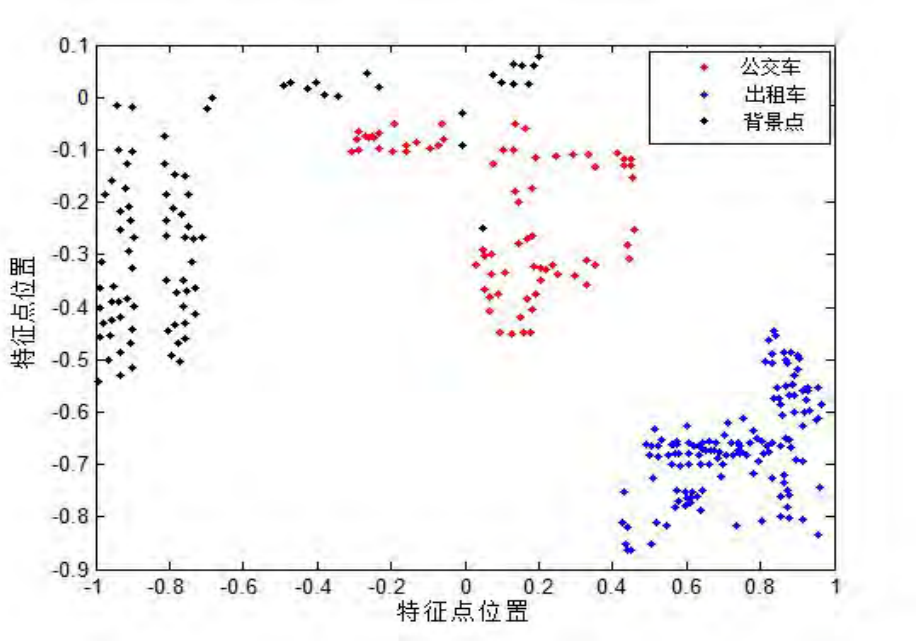


图 5.11　运动分割结果

表 5-5 LRRTrack 算法模型分类结果

标号	类别																			
1-20	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
21-40	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2
41-60	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
61-80	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
81-100	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
101-120	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
121-140	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
141-160	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
161-180	1	2	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
181-200	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
201-220	1	1	1	1	1	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3
221-240	3	3	3	3	3	3	3	3	3	1	3	1	3	3	3	3	1	3	1	3
241-260	3	3	3	3	3	3	3	3	3	3	1	1	3	3	3	3	3	3	3	3
161-280	3	3	3	3	3	3	3	1	3	1	3	1	3	3	3	3	3	3	3	3
281-200	3	1	3	3	3	3	1	3	3	3	3	3	3	3	3	3	3			

4）模型四求解：

在人脸识别时，求解 SSC 算法建立的模型，采用 SSC 算法进行分类，在编程过程中我们涉及到了其中三个的参数分别设为：$\rho = 1, \alpha = 20, \gamma = 0$，得到的

结果见表 5-6。

表 5-6　SSC 算法人脸分类效果

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

5）模型五求解：

在对原始数据降维时，根据要求分别选取了 3 个贡献率 p 分别为 85%，95%，99%，得到降维后矩阵尺寸及两类聚类结果如下表 5-7：

表 5-7　PCA+K-means 模型参数

p	降维后尺寸	1 类数目	2 类数目
85%	3×20	10	10
95%	4×20	10	10
99%	7×20	10	10

通过模型求解得到，无论贡献率 *P* 为所示三种情况的任一种情况，分类结果都相同，具体分类结果见表 5-8。在不影响结果的前提下，要求复杂度越小越好，计算量越少越好，因此最终选取 *p*=85%为指标。

表 5-8　PCA+K-means 人脸分类效果

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

5.3.3 模型的分析

本节算法的源程序采用 C++与 Matlab 语言编写，实验平台为：Intel（R）Core(TM)，2.53GHz i3CPU 和 6G 内存。

a.针对十字的分类问题，模型一：一维映射中心模型与模型二：SMMC 算法模型聚类结果相比，针对本实验数据，前者占优，但是一维映射中心模型使用具有很强的限制性，当十字发生旋转时，该方法则不再适用，虽然 SMMC 针对本实验数据效果稍差，但是却普遍适用性更强，可以用来解决所有同类问题。

b.针对运动分割问题，在 LRR 的基础上提出模型三：LRRTrack 模型，对误分的背景特征点与运动目标特征点有更好的处理效果。

c.针对人脸识别问题，PCA 与 SSC 识别完全效果相同，分析数据，每个人脸图像的数据为 2016 维的列向量，根据图像的先验知识可以反推回原始人脸图像的灰度图，结合参考文件[6]经过实验测试我们可以得出在图像的大小为 48×42 像素时，可以重构回原始的 20 张人脸图像见图 5.12。

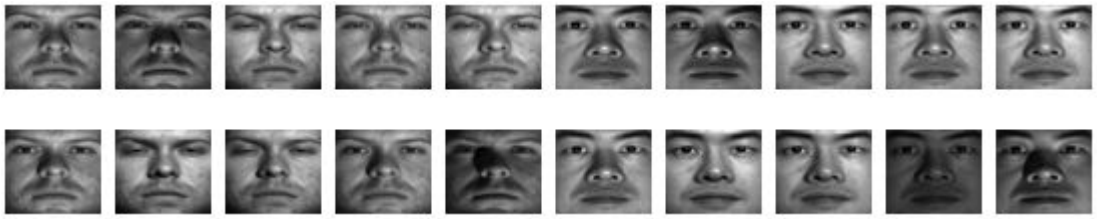


图 5.12　重构回的原始人脸图像从上到下从左到右为 1-20

由图 5.12 可以明显的看出 1-5 号，11-15 号为同一个人在不同光照下的人脸图像，6-10 号，16-20 号为另外一个人在不同光照下的人脸图像。通过与表 5-6 和表 5-4 中的实验结果相比较，我们建立的模型四与模型五的分类结果完全相同，错误率都为 0。

5.4 问题四模型的建立与求解

5.4.1 模型的建立

（1）针对圆台分类问题：

模型一：SMMC 算法建立模型

经过问题的分析，如图 3（a）中一个圆台的点云，圆台的顶，底和侧面分别采样于不同流形，那么基于谱聚类的方法是解决该问题的流行方法，因此我们采用 SMMC 解决该问题。

即 5.2.1 中 SMMC 算法建立的模型，同样给出了算法步骤，据此利用 Matlab 编程求解问题。

（2）针对机器工件外部轮廓分类问题：

模型二：SMMC 算法建立模型

题目要求将机器工件外部边缘轮廓的图像根据轮廓线中不同的直线和圆弧的分类，类数自定，此问题属于多流形聚类的问题。因此，考虑仍然采用 SMMC 解决该问题。

即 5.2.1 中 SMMC 算法建立的模型，同样给出了算法步骤，据此利用 Matlab 编程求解问题。

5.4.2 模型的求解

模型一求解：

在编程过程中我们涉及到了 SMMC 算法中的几个参数，包括：局部化模型个数 M，近邻点数 K，以及调节参数 o（其中聚类数 k 和流形维数 d 根据问题为

表 5-7　SMMC 算法圆台分类参数

参数名	调节参数 o	近邻点数 n	局部化模型 M	聚类数 k	流形维数 d
参数值	12	$4*\log(N)$	30	3	2

定值）。根据本文中图 3（a）圆台的点云，我们设置的参数见表 5-7。

通过表 5-7 中参数设计，运行程序所获得的结果，效果见图 5.13，具体分类结果见附录一中表 8。

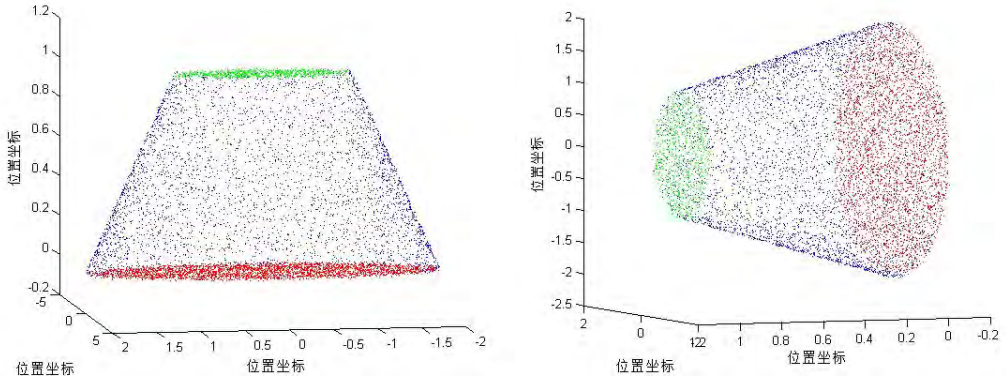


图 5.13 SMMC 算法求得的分类结果，其中绿色代表圆台顶部流形的点，蓝色代代表圆台侧面流形的点，红色代表圆台流形底部的点。

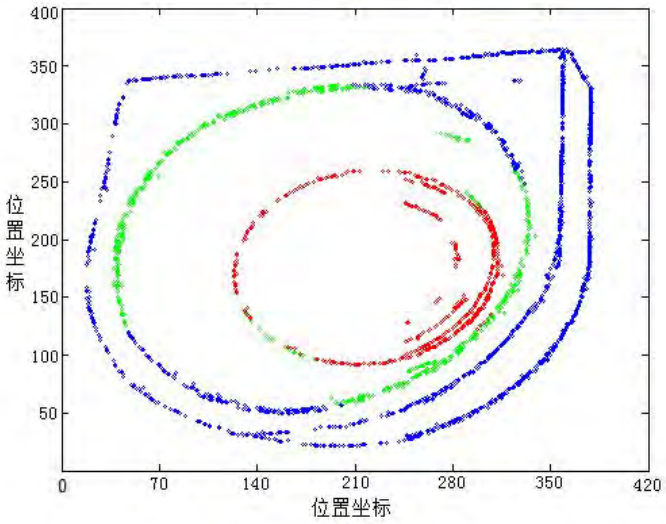


图 5.14　SMMC 算法对机器工件外部边缘的分类结果

模型二求解：

在编程过程中我们同样涉及到了 SMMC 模型中的几个参数，包括：局部化模型个数 M，近邻点数 K，以及调节参数 o（其中聚类数 k 和流形维数 d 根据问题为定值）。根据本文中图 3（b）机器工件外部边缘的轮廓图像，根据不同的直线和圆弧进行分类，那么直线的长短以及圆弧的弧长都作为了分类的条件，大致可以分为 3 类（即 $k=3$），设置的参数如表 5-8。

通过表 5-8 参数的设计，运行程序所获得的结果见图 5.14。

表 5-8　SMMC 算法工件轮廓分类参数

参数名	调节参数 o	近邻点数 n	局部化模型 M	聚类数 k	流形维数 d
参数值	14	$4*\log(N)$	26	3	1

5.4.3 模型的分析

本节算法的源程序采用 Matlab 语言编写，实验平台为：Intel 多核系统、4*2.0GHz CPU 和 16G 内存。

从上面的实验结果可以看出，谱多流形聚类方法（SMMC）很好的解决了云台点云数据的分类问题。它是从相似性矩阵的角度出发，充分利用流形采样点所内含的自然的局部集合结构信息来辅助构造更合适的相似性矩阵并进而发现正确的流形聚类。但是该方法存在参数调节困难的问题，三个调节参数（M ,K ,o ）对实验结果有很大的影响，其中一个参数稍有改动就会对实验结果造成很大的影响，造成实验结果很不稳定。所以，要想获得理想的实验数据需要充分理解该方法，并不断的测试。

六、模型评价与改进

6.1 LRR 模型的评价与改进

LRR 是基于图正则低秩表示的维数约减算法[10]，根据低秩表示对噪声的鲁棒性以及全局性表示的特性，结合图的方法中数据的流形几何结构信息，提出的图正则低秩表示维数约减算法，并在 3（c）题中人脸分类数据上进行了实验验证，与传统的维数约减方法进行比较，提高了后续分类的精确度，且算法性能稳定。

在维数约减方面，低秩表示在子空间分割中能有非常好的效果，低秩表示和低秩分解都可以作为进一步的研究方向，可以结合半监督方法和流形学习的其他办法进行拓展。

6.2 SMMC 模型的评价与改进

针对 SMMC 模型，从相似性矩阵构造的角度出发，提出了基于谱方法的流形聚类方法来分组数据中的低维混合流形结构，从而有效地解决了混合流形聚类问题。SMMC 方法从相似性矩阵的角度出发，充分利用流形采样点所内含的局部几何结构信息来辅助构造更合适的相似性矩阵来分组混合结构数据，即来自不同流形结构的数据点之间有相对低的权值。

SMMC 是基于谱图理论的谱聚类方法，用于混合流形聚类的潜力并据此提出了用于分组混合结构数据，也可以深入研究基于其它数学学科和理论（例如线性和多线性代数、计算拓扑与几何、概率建模等）的方法用于混合流形聚类的潜力并设计相应的方法来处理混合流形聚类问题，也可以建立一些全新的理论和方法来解决这个问题。另一方面，可以进一步考察和拓展流形聚类在其它领域的应用，如文本数据、生物数据、网络数据分析等，建立更适用于具体问题背景的流形聚类理论和方法。

6.3 SSC 模型的评价与改进

稀疏子空间聚类[11]是一种基于谱聚类的数据聚类框架。 SSC 利用高维数据的稀疏表示系数构造相似度矩阵，然后利用谱聚类方法得到数据的子空间聚类结果. 其核心是设计能够揭示高维数据真实子空间结构的表示模型，使得到的表示系数及由此构造的相似度矩阵有助于精确的子空间聚类。稀疏子空间聚类的性能取决于两个方面：一方面, 模型对各种噪声、数据缺损、奇异数据的鲁棒性取决于数据项；另一方面, 子空间表示系数矩阵的结构取决于正则项. 现有模型选择不同的度量来设计数据项与正则项, 存在一定的局限性。

1）针对数据项$F(E)$，根据实际数据的噪声统计分布情况进行误差建模，从而更好的拟合数据，提高鲁棒性。

2）设计适当的正则项$R(Z)$, 使得相应的子空间聚类模型得到的表示系数矩阵Z满足类间系数，类内一致的性质，从而提高聚类的性能。

3）针对所提的子空间聚类模型，设计快速有效的算法，降低算法的时间复杂度，同时考虑算法的可扩展性。

4）进一步完善其理论，寻求更广泛的应用领域。

6.4 总结

针对题目的要求以上三种模型可以相应的解决文中的问题，但是每种算法都有其局限性，模型选取不当就会得不到理想的效果，因此应该对模型按照前面提高的改进方向进行改进，追求一般性使其适用性得到提高。

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] R. Vidal. Subspace clustering. IEEE Signal Processing Magazine, 28(2):52–68, 2011.

[3]WOLD S,ESBENSEN K,GELADI P.Principal Component Analysis［J］.Chemometrics and Intelligent Laboratory Systems,1987,2: 37.

[4] G. Liu, Z. Lin, S. Yan, J. Sun,Y. Yu, and Y. Ma.Robust recovery of subspace structures by low-rank representation. IEEE Transactions on Pattern Analysis and Machine Intelligence,35(1):171–184, 2013.

[5] 王勇，基于流形学习的分类与聚类方法及其应用研究. 国防科学技术大学研究生院博士学位论文.

[6] Y. Wang, Y. Jiang, Y. Wu, and Z. Zhou.Spectral clustering on multiple manifolds.IEEE Transactions on Neural Networks,22(7):1149-1161,2011

[7] D. L. Donoho. Compressed sensing. IEEE Transactions on Information Theory,vol.52, no.4, pp.1289-1306, Apr.2006.

[8] 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
[9] J. Shi and J. Malik.Normalized cuts and image segmentation.IEEE Transactions Pattern Analysis Machine Intelligence, 22(8):888–905, 2000.
[10]贺予迪. 基于流形学习和低秩表示的维数约减算法研究[D]. 西安电子科技大学, 2014.
[11]王卫卫, 李小平, 冯象初, 王斯琪. 稀疏子空间聚类综述[J]. 自动化学报, 2015, 08:1373-1384.

附录

附录一：

表 1　PCA+K-means 模型求解结果

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

类别	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
标号	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 2　图 2.2（a）SMMC 模型求解结果

1~20	1	2	2	1	2	2	2	2	2	1	2	1	1	1	2	1	2	2	1	1
21~40	2	1	1	2	2	2	2	1	2	1	1	1	1	2	2	2	2	1	1	1
41~60	2	2	2	1	2	1	2	2	1	2	2	2	2	1	2	2	2	1	1	2
61~80	2	1	1	1	2	2	2	2	1	1	1	1	1	2	1	1	2	1	2	2
81~100	1	2	2	2	1	2	2	2	2	1	1	1	1	2	2	2	2	2	2	1
101~120	1	1	2	2	1	2	1	1	1	1	1	1	2	1	1	2	2	2	2	1
121~140	2	1	1	1	1	2	2	1	2	1	2	1	2	2	1	2	1	1	1	1
141~160	2	2	1	1	2	1	2	1	2	2	2	2	2	1	2	2	2	1	2	1
161~180	1	2	1	1	2	2	1	2	2	2	2	2	2	1	2	1	2	1	2	1
181~200	1	2	1	2	1	2	1	2	1	1	1	2	1	1	2	1	1	1	2	1
201~220	1	1	1	1	1	2	1	1	1	2	1	2	1	1	1	1	2	2	1	1
221~240	1	2	2	1	2	1	2	2	1	2	1	1	1	2	2	2	1	1	2	1
241~260	1	1	2	2	1	2	1	2	2	2	2	1	2	2	1	2	1	1	2	2
261~280	1	1	2	1	1	1	1	2	1	1	1	2	2	2	1	2	2	1	2	2
281~300	1	1	1	1	2	1	2	1	2	2	2	2	1	2	1	1	2	1	1	2
301~320	1	2	1	1	1	1	2	1	2	2	2	1	1	2	1	2	1	1	1	1
321~340	2	2	1	1	1	2	1	1	2	2	1	2	2	2	1	1	2	2	1	2

表 3　图 2.2（b）SMMC 模型求解结果

1~20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1
21~40	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
41~60	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
61~80	3	2	2	2	2	2	2	2	2	2	3	2	3	2	3	3	3	2	2	2
81~100	2	2	2	3	3	2	3	2	2	3	3	3	3	2	3	2	2	2	2	3
101~120	2	2	2	3	3	2	3	2	2	3	2	2	2	3	2	2	2	2	3	3
121~140	2	2	2	3	2	3	3	2	3	3	3	2	3	3	2	2	3	3	2	3
141~160	2	3	3	2	2	2	3	2	3	3	3	2	2	3	2	2	2	2	3	2
161~180	2	3	3	2	2	2	2	2	2	3	3	2	3	3	2	3	2	3	2	3
181~200	2	3	3	2	2	3	2	2	2	2	3	2	3	2	3	3	2	2	2	3
201~220	3	2	3	3	3	3	3	3	3	2	2	2	3	3	2	2	2	2	2	3
221~240	3	2	2	3	2	2	3	2	2	2	3	2	3	2	2	2	2	3	2	3
241~260	2	3	3	3	3	2	2	3	3	2	2	2	3	3	2	2	2	3	3	2
261~280	3	3	2	2	3	2	3	3	3	3	3	3	2	3	2	2	2	3	3	3
281~300	1	3	3	3	2	2	3	2	3	2	2	3	2	2	3	3	2	2	2	2

表 4　图 2.2（b）SSC 模型求解结果

1~20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1
21~40	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
41~60	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
61~80	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
81~100	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
101~120	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
121~140	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
141~160	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
161~180	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
181~200	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
201~220	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
221~240	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
241~260	3	3	3	3	3	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3
261~280	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
281~300	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

表 5　图 2.2（c）SMMC 模型求解结果

1~20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21~40	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
41~60	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
61~80	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
81~100	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
101~120	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
121~140	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
141~160	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
161~180	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
181~200	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
201~220	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
221~240	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
241~260	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
261~280	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
281~300	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
301~320	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
321~340	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
341~360	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
361~380	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
381~400	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

表 6　图 2.2（d）SMMC 模型求解结果

1~20	2	2	2	2	2	2	1	2	1	1	1	2	1	2	1	2	1	2	2	1
21~40	2	2	1	2	2	1	1	1	2	2	2	2	2	1	2	1	2	2	1	1
41~60	2	2	2	2	1	2	1	1	2	2	1	1	2	1	1	2	2	1	1	1
61~80	2	1	2	2	1	1	2	2	1	2	1	2	2	2	1	1	2	1	2	2
81~100	1	1	2	2	2	2	2	1	2	2	1	1	1	2	2	2	1	1	2	1
101~120	1	2	1	1	2	2	1	2	2	2	2	2	2	2	1	2	1	2	2	2
121~140	2	2	1	1	2	1	2	1	1	1	1	1	2	1	2	1	1	1	1	1
141~160	2	1	1	1	1	2	1	2	1	2	2	1	1	2	1	2	2	2	1	2
161~180	2	1	2	1	1	2	2	2	2	1	2	1	1	2	1	2	1	1	2	1
181~200	2	2	1	2	1	2	1	2	2	1	2	2	2	1	1	2	1	1	1	1
201~220	1	1	1	2	2	2	2	1	1	2	2	2	1	1	2	1	1	2	1	2
221~240	2	2	2	2	1	1	2	2	1	2	2	2	2	2	2	1	1	1	2	1
241~260	2	1	2	1	1	2	2	2	2	1	1	2	2	2	1	2	1	2	2	1
261~280	1	2	2	1	1	2	1	2	2	1	1	1	1	1	2	1	1	1	2	2
281~300	1	2	2	1	2	1	2	1	1	2	1	2	2	1	1	1	1	2	1	1
301~320	1	1	2	1	1	2	2	1	1	1	1	1	2	1	1	2	2	1	2	2
321~340	2	2	2	1	2	2	1	1	1	1	2	2	2	1	2	1	2	2	1	2
341~360	2	2	1	2	1	2	1	2	1	1	2	2	1	2	1	1	1	2	1	1
361~380	1	1	1	1	2	1	2	1	2	1	2	2	1	1	2	1	2	2	2	1
381~400	1	1	1	1	1	1	2	1	2	2	2	1	2	1	1	2	1	1	2	2
401~420	2	2	2	1	1	2	1	1	2	1	2	2	1	1	1	1	1	1	1	1
421~440	1	2	2	1	2	1	1	2	1	2	1	2	1	2	2	2	2	2	1	1
441~460	2	2	2	2	1	2	1	1	2	1	2	1	2	1	1	1	2	1	2	1
461~480	2	2	1	1	2	1	1	1	1	1	2	2	1	2	1	1	1	1	2	2
481~500	2	2	2	1	1	2	2	2	2	1	2	1	2	1	1	1	1	1	2	1
501~520	2	2	1	1	1	1	2	1	2	2	2	2	2	2	1	2	2	2	1	1
521~540	1	2	2	2	1	1	2	2	2	2	1	1	1	1	1	1	2	1	2	2
541~560	1	1	1	2	2	1	2	1	1	2	2	1	1	2	2	2	1	2	2	2
561~580	1	2	2	2	1	2	2	1	2	2	1	2	2	1	2	1	1	2	1	1
581~600	1	1	1	2	1	2	1	1	1	2	2	2	1	1	1	1	2	1	2	2
601~620	1	2	2	2	1	1	1	2	2	1	2	2	2	1	1	2	2	1	2	2
621~640	2	2	2	2	1	1	1	1	2	1	1	2	1	2	1	2	1	1	2	2
641~660	1	2	1	2	1	1	1	2	1	2	1	2	2	2	2	2	2	1	2	2
661~680	1	2	2	1	2	2	2	2	2	2	2	2	1	2	1	1	1	2	2	2
681~700	2	1	1	2	2	1	2	1	2	1	1	2	1	2	1	2	2	1	1	2
701~720	1	1	1	2	2	2	1	1	2	2	2	1	1	2	1	2	1	1	2	2
721~740	2	1	1	1	2	2	1	2	2	1	1	2	1	2	2	1	1	2	1	2
741~760	2	1	2	1	1	1	2	2	2	1	1	1	2	2	2	1	1	1	1	2

761~780	1	2	2	1	1	2	1	2	1	2	2	1	2	1	2	1	1	2	2	2
781~800	1	1	2	2	1	1	2	2	1	1	1	2	2	1	2	2	2	2	1	2
801~820	2	1	1	1	1	1	1	2	1	1	2	2	1	2	1	2	1	1	2	1
821~840	1	2	2	2	2	2	2	2	2	2	1	2	1	2	1	2	2	1	2	2
841~860	1	2	1	1	1	1	1	2	2	1	2	2	2	2	1	1	1	1	1	1
861~880	2	2	2	1	2	2	1	2	1	1	1	1	1	1	1	1	1	2	2	2
881~900	1	2	2	2	1	1	2	2	2	1	2	1	2	1	2	1	2	2	1	1
901~920	1	1	1	2	2	2	1	1	1	2	1	2	1	1	2	2	1	2	2	1
921~940	1	1	1	2	1	2	1	1	2	1	2	1	1	1	2	2	1	1	1	2
941~960	1	1	1	2	2	1	1	2	1	2	1	1	1	1	2	2	2	1	2	2
961~980	1	2	1	1	2	2	1	1	2	1	2	2	2	1	1	2	2	1	1	1
981~988	2	2	1	2	1	2	2	2												

表 7　图 3.1（a）SMMC 模型求解结果

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

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

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

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

注：从左到右，从上到下，每行 17 个，依次排序

表 8　图 4.1（a）SMMC 模型求解结果

1~20	2	2	2	1	3	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
21~40	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
41~60	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
61~80	2	2	2	1	1	1	1	3	1	1	1	3	3	3	3	3	3	3	3	3
81~100	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
101~120	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
121~140	2	2	2	3	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
141~160	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
161~180	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
181~200	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
201~220	2	2	2	1	1	1	1	1	3	1	1	3	3	3	3	3	2	3	3	3
221~240	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
241~260	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
261~280	2	2	2	1	3	1	1	1	1	1	1	3	3	3	3	3	3	1	3	3
281~300	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
301~320	3	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
321~340	2	3	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
341~360	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
361~380	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
381~400	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
401~420	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
421~440	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
441~460	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
461~480	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
481~500	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
501~520	3	2	2	1	1	1	1	1	1	1	1	3	3	3	3	1	3	3	3	3
521~540	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
541~560	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
561~580	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
581~600	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
601~620	2	2	2	1	1	1	1	3	1	3	1	3	3	3	3	3	3	3	3	3
621~640	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
641~660	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
661~680	2	2	2	1	1	1	1	1	1	1	1	3	3	3	1	3	3	3	3	3
681~700	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
701~720	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3

721~740	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
741~760	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
761~780	2	2	2	1	1	1	1	1	1	1	1	3	3	2	3	3	3	3	3	3
781~800	2	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3
801~820	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
821~840	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
841~860	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
861~880	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
881~900	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
901~920	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
921~940	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
941~960	2	3	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
961~980	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
981~1000	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1001~1020	2	2	2	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
1021~1040	2	2	3	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1041~1060	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1061~1080	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1081~1100	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1101~1120	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1121~1140	2	2	2	1	1	3	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1141~1160	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1161~1180	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1181~1200	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1201~1220	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1221~1240	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1241~1260	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1261~1280	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1281~1300	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1301~1320	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1321~1340	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1341~1360	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1361~1380	2	2	2	1	3	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1381~1400	2	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1401~1420	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1421~1440	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1441~1460	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1461~1480	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1481~1500	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3

1501~1520	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1521~1540	2	2	1	1	1	1	1	1	1	1	1	3	3	1	3	3	3	3	3	3
1541~1560	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1561~1580	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1581~1600	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1601~1620	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1621~1640	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1641~1660	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1661~1680	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1681~1700	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1701~1720	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1721~1740	2	2	3	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1741~1760	2	2	1	1	1	3	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1761~1780	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1781~1800	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1801~1820	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1821~1840	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1841~1860	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1861~1880	2	2	1	1	1	1	1	3	1	1	1	3	3	3	1	3	3	3	3	3
1881~1900	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1901~1920	2	2	1	1	1	1	1	1	1	3	1	3	3	3	3	3	3	3	3	3
1921~1940	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1941~1960	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1961~1980	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
1981~2000	2	2	3	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2001~2020	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2021~2040	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2041~2060	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2061~2080	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2081~2100	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2101~2120	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2121~2140	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2141~2160	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2161~2180	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2181~2200	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2201~2220	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2221~2240	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2241~2260	2	2	1	1	1	1	1	1	1	1	1	3	3	3	1	3	3	3	3	3
2261~2280	2	2	1	3	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3

2281~2300	2	2	1	1	1	1	1	3	1	1	1	3	3	3	3	3	3	3	3	3
2301~2320	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2321~2340	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2341~2360	3	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2361~2380	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2381~2400	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2401~2420	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2421~2440	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2441~2460	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2461~2480	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2481~2500	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2501~2520	2	2	1	1	1	1	1	3	1	1	1	3	3	3	3	3	3	3	3	3
2521~2540	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2541~2560	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2561~2580	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2581~2600	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
2601~2620	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2621~2640	2	2	1	1	1	1	1	1	1	3	1	3	3	3	3	3	3	3	3	3
2641~2660	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2661~2680	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2681~2700	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2701~2720	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2721~2740	2	2	1	1	1	1	3	1	1	1	1	3	3	3	3	3	3	3	3	3
2741~2760	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2761~2780	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2781~2800	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2801~2820	3	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2821~2840	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2841~2860	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2861~2880	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2881~2900	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2901~2920	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2921~2940	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2941~2960	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
2961~2980	2	2	1	1	1	1	1	3	1	1	1	3	3	3	3	3	3	3	3	3
2981~3000	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3001~3020	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3021~3040	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3041~3060	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3

3061~3080	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3081~3100	2	2	1	1	1	1	1	1	1	3	1	3	3	3	3	3	3	3	3	3
3101~3120	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	2	3
3121~3140	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3141~3160	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3161~3180	2	2	1	3	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3181~3200	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3201~3220	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3221~3240	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3241~3260	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3261~3280	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3281~3300	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	1	3	3
3301~3320	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3321~3340	2	2	1	1	3	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3341~3360	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3361~3380	2	2	1	1	1	3	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3381~3400	2	2	3	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3401~3420	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3421~3440	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3441~3460	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3461~3480	2	2	1	1	1	1	3	1	1	1	1	3	3	3	3	3	3	1	3	3
3481~3500	2	2	3	1	1	1	1	1	1	1	1	3	3	3	3	3	1	3	3	3
3501~3520	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3521~3540	2	2	3	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3541~3560	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3561~3580	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3581~3600	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3601~3620	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3621~3640	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3641~3660	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3661~3680	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3681~3700	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3701~3720	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3721~3740	2	2	3	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3741~3760	3	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3761~3780	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3781~3800	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3801~3820	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3821~3840	2	2	1	1	1	1	1	1	3	1	1	3	3	3	3	3	3	3	3	3

3841~3860	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3861~3880	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3881~3900	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3901~3920	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3921~3940	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3941~3960	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
3961~3980	2	2	1	1	1	1	1	1	3	1	1	3	3	3	3	3	3	3	3	3
3981~4000	2	2	1	1	1	1	1	1	1	1	1	3	3	1	3	3	3	3	3	3
4001~4020	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4021~4040	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4041~4060	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4061~4080	2	2	1	1	1	1	3	1	1	1	1	1	3	3	3	3	3	3	3	3
4081~4100	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4101~4120	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4121~4140	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4141~4160	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4161~4180	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4181~4200	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4201~4220	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4221~4240	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4241~4260	2	2	1	1	1	3	1	1	1	1	1	3	3	1	3	3	3	3	3	3
4261~4280	2	2	1	1	1	1	3	1	1	1	1	3	3	3	3	3	3	3	3	3
4281~4300	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4301~4320	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4321~4340	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4341~4360	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4361~4380	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4381~4400	2	2	1	1	1	1	1	1	1	1	1	3	3	1	3	1	3	3	3	3
4401~4420	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4421~4440	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4441~4460	2	2	1	1	1	1	3	1	1	1	1	3	3	3	3	3	3	3	3	3
4461~4480	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4481~4500	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4501~4520	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4521~4540	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4541~4560	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4561~4580	2	2	1	1	1	1	1	1	3	1	1	3	3	3	3	3	3	3	3	3
4581~4600	2	2	1	1	1	3	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4601~4620	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3

4621~4640	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4641~4660	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4661~4680	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4681~4700	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4701~4720	2	2	1	3	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4721~4740	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4741~4760	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4761~4780	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4781~4800	2	2	3	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4801~4820	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4821~4840	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4841~4860	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4861~4880	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4881~4900	2	3	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4901~4920	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4921~4940	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4941~4960	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4961~4980	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
4981~5000	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5001~5020	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5021~5040	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5041~5060	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5061~5080	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5081~5100	2	2	1	1	1	1	3	1	1	1	1	3	3	3	3	3	3	3	3	3
5101~5120	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5121~5140	2	2	1	1	1	1	1	1	1	3	1	3	3	3	3	3	3	3	3	3
5141~5160	2	2	3	1	1	1	1	3	1	1	1	3	3	3	3	3	3	3	3	3
5161~5180	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5181~5200	3	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5201~5220	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5221~5240	2	2	1	1	1	1	1	1	1	3	1	3	3	3	3	3	3	2	3	3
5241~5260	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5261~5280	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5281~5300	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5301~5320	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5321~5340	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5341~5360	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5361~5380	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5381~5400	2	2	1	1	1	1	1	3	1	1	1	3	3	3	3	3	3	3	3	3

5401~5420	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5421~5440	2	2	1	1	3	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5441~5460	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5461~5480	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5481~5500	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5501~5520	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5521~5540	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5541~5560	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5561~5580	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5581~5600	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5601~5620	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5621~5640	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5641~5660	3	2	1	1	1	1	1	1	1	3	1	3	3	3	3	3	1	3	3	3
5661~5680	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5681~5700	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5701~5720	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5721~5740	2	2	1	1	1	1	3	1	1	1	1	3	3	3	3	3	3	3	3	3
5741~5760	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5761~5780	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5781~5800	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5801~5820	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5821~5840	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5841~5860	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5861~5880	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5881~5900	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5901~5920	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5921~5940	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5941~5960	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5961~5980	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
5981~6000	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6001~6020	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6021~6040	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6041~6060	3	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6061~6080	2	3	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6081~6100	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6101~6120	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6121~6140	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6141~6160	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6161~6180	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3

6181~6200	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6201~6220	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
6221~6240	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6241~6260	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6261~6280	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6281~6300	2	2	1	1	1	1	1	3	1	1	1	3	3	3	3	3	3	3	3	3
6301~6320	2	2	1	1	1	1	1	1	3	1	1	3	3	3	3	3	3	3	3	3
6321~6340	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6341~6360	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6361~6380	2	2	1	1	1	1	1	1	3	1	1	3	3	3	3	3	3	3	3	3
6381~6400	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6401~6420	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6421~6440	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6441~6460	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6461~6480	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6481~6500	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6501~6520	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6521~6540	2	2	1	1	1	3	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6541~6560	2	2	1	1	1	1	3	1	1	1	1	3	3	3	3	3	3	3	3	3
6561~6580	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6581~6600	3	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6601~6620	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6621~6640	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6641~6660	2	3	1	1	1	1	3	1	1	1	1	3	3	3	3	3	3	3	3	3
6661~6680	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6681~6700	2	2	1	1	1	1	1	3	1	1	1	3	3	3	3	3	3	3	3	3
6701~6720	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6721~6740	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6741~6760	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6761~6780	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6781~6800	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6801~6820	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6821~6840	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6841~6860	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6861~6880	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6881~6900	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6901~6920	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6921~6940	2	2	1	1	1	3	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6941~6960	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3

6961~6980	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
6981~7000	2	2	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
7001~7020	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7021~7040	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7041~7060	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7061~7080	3	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7081~7100	2	2	1	1	1	1	1	3	1	1	3	3	3	3	3	3	3	3	3	3
7101~7120	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7121~7140	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	1	3	3
7141~7160	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7161~7180	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7181~7200	3	2	3	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7201~7220	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7221~7240	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7241~7260	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7261~7280	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7281~7300	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7301~7320	3	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7321~7340	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7341~7360	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7361~7380	3	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7381~7400	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7401~7420	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7421~7440	2	2	3	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7441~7460	2	2	1	1	3	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7461~7480	2	2	1	1	1	1	1	3	1	1	3	3	3	3	3	3	3	3	3	3
7481~7500	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7501~7520	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7521~7540	3	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7541~7560	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7561~7580	2	2	1	1	1	1	1	3	1	1	3	3	3	3	3	3	3	3	3	3
7581~7600	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7601~7620	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7621~7640	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7641~7660	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7661~7680	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7681~7700	2	2	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3
7701~7720	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7721~7740	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3

7741~7760	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7761~7780	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7781~7800	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7801~7820	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7821~7840	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7841~7860	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7861~7880	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7881~7900	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7901~7920	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7921~7940	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7941~7960	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7961~7980	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
7981~8000	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
8001~8020	2	2	1	3	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
8021~8040	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	1	3	3	3	3
8041~8060	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
8061~8080	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
8081~8100	2	2	1	1	1	3	1	1	1	1	3	3	3	3	3	3	3	3	3	3
8101~8120	3	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
8121~8140	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
8141~8160	3	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
8161~8180	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
8181~8200	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
8201~8220	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
8221~8240	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
8241~8260	3	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	2	3	3	3
8261~8280	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3
8281~8300	2	2	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3