稀疏凸聚类学习报告

目录

摘要

凸聚类是 k 均值聚类和层次聚类的一种凸松弛方法，它很好地解决了传统非凸聚类方法的不稳定性问题，近年来备受关注。虽然凸聚类的计算和统计特性最近得到了研究，但是在高维聚类场景中，凸聚类的性能还没有得到研究，在高维聚类场景中，数据包含大量的特征，其中许多特征没有关于聚类结构的信息。在这篇文章中，作者证明了如果在聚类中包含非信息特征，凸聚类的性能可能会被扭曲。为了克服这个问题，作者引入了一种新的聚类方法，称为稀疏凸聚类，它可以同时对观测数据进行聚类并进行特征选择。其关键思想是用正则化的形式表示凸聚类，并在聚类中心上加上一个自适应的组 LASSO 惩罚项。为了最优地平衡聚类拟合和稀疏性之间的平衡，提出了一种基于聚类稳定性的优化准则。

1 介绍

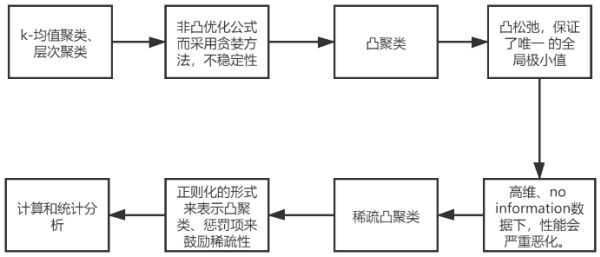

图 1-1 本节结构

1.1 凸聚类

聚类分析是一种无监督学习方法，其目的是将观察结果分配到若干个聚类中，使同一组中的观察结果彼此相似。传统的聚类方法，如 k-均值聚类、层次聚类和高斯混合模型，由于其非凸优化公式而采用贪婪方法，因而存在不稳定性。为了解决原始聚类方法的不稳定性问题，凸聚类被提出。对于 n 个观测值，每个观测值有 p 个特征的数据矩阵 X，这 n 个观测值的聚类等价于解决如下最小化问题：

$$\min_{\mathbf{A}\in\mathbb{R}^{n\times p}} \frac{1}{2}\sum_{i=1}^{n}||X_{i\cdot}-A_{i\cdot}||_2^2+\gamma\sum_{i_1<i_2}||A_{i_1\cdot}-A_{i_2\cdot}||_q, \tag{1}$$

K-means，层次聚类在第二项中考虑到了 L0 范数，这导致了非凸优化问题。因此，凸聚类可以视为其凸松弛，并且由于该凸松弛保证了它实现唯一的全局极小值。

由于 fused-lasso 的惩罚项在上式的第二项，上式鼓励特征矩阵 A 的行是相同的。如果不同行相等，说明其属于同一个类。参数 gamma 控制了特征矩阵 A 行的唯一性，也就是估计的类的数量。随着 gamma 的增大，一些行变成相同的，这展示了融化过程。对比传统非凸聚类算法，由于目标函数严格为凸，该估计对于每一个 gamma 是唯一的。

1.2 稀疏凸聚类

近年来，人们对凸聚类的计算和统计性质进行了研究。特别是 Zhu 等人(2014)为凸聚类提供了恢复真实聚类的条件，Chi 和 Lange(2015)提出了高效、可伸缩的凸聚类实现，Tan 和 Witten(2015)研究了凸聚类的几种统计特性。

凸聚类具有良好的理论性质和计算效率，但在聚类高维数据时，当特征数量较大且很多特征可能不包含聚类结构信息时，其性能会严重恶化。

本文介绍了一种新的聚类方法——稀疏凸聚类，将稀疏性引入到高维数据的凸聚类中。其核心思想是用**正则化的形式来表示凸聚类**，并在聚类中心上附加一个**主动的组 LASSO 惩罚项来鼓励稀疏性**。这种正则化算法虽然简单，但与传统的凸聚类算法相比，需要更具有**挑战性的计算和统计分析**。特别地，在计算上，我们需要将稀疏凸包重新规划成几个子优化问题，然后通过一个伪回归公式来解决每个子优化问题。为了证明所提出的稀疏凸聚类方法是一个无偏估计，我们需要仔细量化由于组 LASSO 惩罚而导致的变量选择的影响。此外，我们还对稀疏凸聚类估计的预测误差进行了非渐近分析。在高维的情况下，维数与样本量是发散的，我们的估计在变量的选择上是一致的。值得注意的是，我们的方法不仅在理论上是合理的，而且在实践中也是有前途的。通过大量的仿真实例和手部运动聚类的实际应用，证明了该方法的优越性。

我们使用第 5 节中第 4 个模拟设置生成的数据集来演示所提方法的优越性能。在这个数据集中，有来自 4 个集群的 60 个观测值，每个观测值有 500 个特征，其中只有前 20 个特征是有信息的。热力图表明稀疏凸聚类能够过滤掉这些非信息性的特性，从而提高聚类性能。

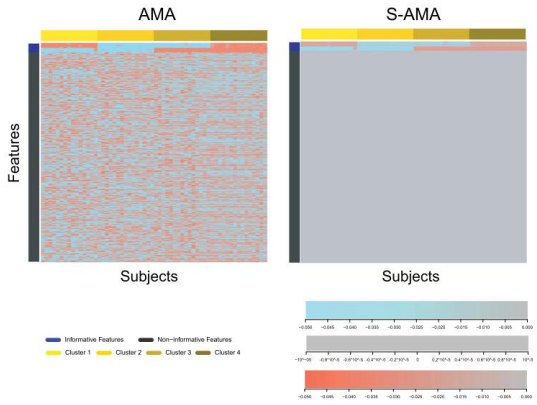

图 1-2 两种方法聚类性能展示

1.3 相关工作

与凸聚类相关的一篇文章是 Chi 和 Lange(2015)提出的凸聚类的高效实现，Chi、Allen 和 Baraniuk(2016)开发了凸聚类的扩展。介绍了两种高效的数据聚类算法 ADMM 和 AMA，它们主要用于低维数据的聚类。为了解决高维性问题，我们的稀疏凸聚类方法的一个关键组成部分是在 ADMM 和 AMA 算法的基础上建立一个新的**正则化惩罚**，以鼓励**聚类中心的稀疏结构。**实验研究表明，这种正则化步骤能够显著提高高维聚类问题的聚类精度。另一个研究方向是同时**聚类和特征选择**。一些方法是基于模型的聚类方法。这些稀疏聚类方法的一个常见构建块是用于特征选择的 lasso 形式的惩罚。我们建议读者查阅 Alelyani、Tang 和 Liu(2013)的全面概述。这些稀疏聚类算法虽然具有良好的数值性能，但由于**非凸优化公式的存在**，仍然存在**不稳定性**。为了克服这个问题，我们的**稀疏凸聚类算法**解决了一个**凸优化问题**，并确

保了一个**唯一的全局解**。

2 模型

凸聚类的一个变体便是为第二项加入自适应的惩罚项，考虑到一个成对系数。

$$\min_{\mathbf{A}\in\mathbb{R}^{n\times p}} \frac{1}{2}\sum_{i=1}^{n}||X_{i\cdot}-A_{i\cdot}||_2^2+\gamma\sum_{i_1<i_2} w_{i_1,i_2}||A_{i_1\cdot}-A_{i_2\cdot}||_q, \tag{2}$$

对于（2），我们将 X 和 A 从行向量转化为特征列向量。并且不失一般性，假设特征向量是中心化的（均值为 0），则可以将其改写为:

$$\min_{\mathbf{A}\in\mathbb{R}^{n\times p}} \frac{1}{2}\sum_{j=1}^{p}||\mathbf{x}_j-\mathbf{a}_j||_2^2+\gamma\sum_{l\in\mathcal{E}} w_l||A_{i_1\cdot}-A_{i_2\cdot}||_q, \tag{3}$$

特征矩阵 A 的行向量和列向量为解，分别为观测估计和特征估计。聚类结构由行向量得到。特征重要程度由列向量得到。由于特征向量是中心化的，所以特征 j 是非信息特征的充要条件是特征向量全 0。以下图为例，其列为特征，行为观测值，F2 和 F3 即为非信息特征，由于 D1 和 D2 一样，D3 和 D4 一样，所以 D1 和 D2 为一类，D3 和 D4 为一类。

A	F1	F2	F3	F4	F5	F6
D1	12	0	0	2	65	4
D2	12	0	0	2	65	4
D3	-8	0	0	1	25	5
D4	-8	0	0	1	25	5

图 2-1 特征矩阵 A 示例

在高维聚类时，希望能获得一个稀疏的 A，即一些列向量为 0 向量。由于去除非信息特征的重要性，作者为原先的损失函数加入了一个自适应的 group lasso 损失。其中 Gamma_2 控制信息特征的数量。Mu 为每个特征的自适应损失。

$$\begin{aligned}\min_{\mathbf{A}\in\mathbb{R}^{n\times p}} &\frac{1}{2}\sum_{j=1}^{p}||\mathbf{x}_j-\mathbf{a}_j||_2^2\\ &+\gamma_1\sum_{l\in\mathcal{E}} w_l||A_{i_1\cdot}-A_{i_2\cdot}||_q+\gamma_2\sum_{j=1}^{p}u_j||\mathbf{a}_j||_2,\end{aligned} \tag{3}$$

Group lasso 形式的损失保持了全局稀疏条件。即使得每一个列向量的元素要么是全 0，要么全是非 0。整体稀疏条件可以被放缩到两个方向。第一，加入 lasso 形式的损失。第二，可以再加入一个 lasso 损失。形成一个稀疏 group lasso 损失。

3 算法

我们对交替方向乘法器（ADMM）与交替最小算法（AMA）两种算法进行改进，得到稀疏交

替方向乘法器（S-ADMM）与稀疏交替最小算法（S-AMA）。为了利用到 ADMM 和 AMA，我们对（3）进行改写：

$$\begin{aligned} \min_{\mathbf{A}\in\mathbb{R}^{n\times p}} \ & \frac{1}{2}\sum_{j=1}^{p}||\mathbf{x}_j-\mathbf{a}_j||_2^2+\gamma_1\sum_{l\in\mathcal{E}}w_l||\mathbf{v}_l||_q+\gamma_2\sum_{j=1}^{p}u_j||\mathbf{a}_j||_2, \\ \text{s.t.} \ & A_{i_1\cdot}-A_{i_2\cdot}-\mathbf{v}_l=\mathbf{0}. \end{aligned} \tag{4}$$

增加了等式约束，将其转化为一个带等式约束的最小化问题。这个优化问题等价于最小化下面的增广拉格朗日函数。

$$\begin{aligned} \mathcal{L}_\nu(\mathbf{A},\mathbf{V},\mathbf{\Lambda}) = & \frac{1}{2}\sum_{j=1}^{p}\|\mathbf{x}_j-\mathbf{a}_j\|_2^2+\gamma_1\sum_{l\in\mathcal{E}}w_l\|\mathbf{v}_l\|_q \\ & +\gamma_2\sum_{j=1}^{p}u_i\|\mathbf{a}_j\|_2+\sum_{l\in\mathcal{E}}\langle\lambda_l,\mathbf{v}_l-A_{i_1\cdot}+A_{i_2\cdot}\rangle \\ & +\frac{\nu}{2}\sum_{l\in\mathcal{E}}\|\mathbf{v}_l-A_{i_1\cdot}+A_{i_2\cdot}\|_2^2, \end{aligned} \tag{5}$$

3.1 S-ADMM

S-ADMM 通过一次交替求解一组变量来最小化增广拉格朗日问题。每次固定剩余变量，优化一组变量。

$$\begin{aligned} \mathbf{A}^{m+1} &= \operatorname*{argmin}_{\mathbf{A}}\mathcal{L}_\nu(\mathbf{A},\mathbf{V}^m,\mathbf{\Lambda}^m), \\ \mathbf{V}^{m+1} &= \operatorname*{argmin}_{\mathbf{V}}\mathcal{L}_\nu(\mathbf{A}^{m+1},\mathbf{V},\mathbf{\Lambda}^m), \\ \boldsymbol{\lambda}_l^{m+1} &= \boldsymbol{\lambda}_l^m+\nu(\mathbf{v}_l^{m+1}-A_{i_1\cdot}^{m+1}+A_{i_2\cdot}^{m+1}),\ l\in\mathcal{E}. \end{aligned} \tag{6}$$

首先，优化 A 等价于最小化以下问题：

$$\begin{aligned} f(\mathbf{A}) = & \frac{1}{2}\sum_{j=1}^{p}\|\mathbf{x}_j-\mathbf{a}_j\|_2^2+\frac{\nu}{2}\sum_{l\in\mathcal{E}}\|\widetilde{\mathbf{v}}_l-A_{i_1\cdot}+A_{i_2\cdot}\|_2^2 \\ & +\gamma_2\sum_{j=1}^{p}u_j\|\mathbf{a}_j\|_2. \end{aligned} \tag{7}$$

优化困难在损失函数包含矩阵 A 的行和列。利用以下引理将其与一个 group-lasso 回归问题联系起来。根据原文的引理 1，可以发现（7）等价于下面的（8）。

$$\min_{\mathbf{a}_j}\frac{1}{2}\|\mathbf{y}_j-\mathbf{N}\mathbf{a}_j\|_2^2+\gamma_2u_j\|\mathbf{a}_j\|_2, \quad \text{for each } j=1,\ldots,p. \tag{8}$$

即将其转化为一个 group-lasso 问题。证明的关键在于置换矩阵的性质。根据这个性质，我们可以将最小化问题转化为 p 个单独的次最优问题。并且与 group-lasso 惩罚的性质一起导致了目标解。引理 1 的证明请参阅原文附录。

第二步 ，更新 V， 我们定义一个近端映射。

$$\text{prox}_{\sigma\Omega}(\mathbf{u})=\operatorname*{argmin}_{\mathbf{v}}\left[\sigma\Omega(\mathbf{v})+\frac{1}{2}\|\mathbf{u}-\mathbf{v}\|_2^2\right]. \tag{9}$$

Omega 函数是 q-范数，具体关系查阅参考文献 Chi and Lange (2015)的表 1。由于 v 是独立的，所以能通过近端映射解决。

$$\begin{aligned}\mathbf{v}_l &= \underset{\mathbf{v}_l}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{v}_l - (A_{i_1\cdot} - A_{i_2\cdot} - \nu^{-1}\boldsymbol{\lambda}_l)\|_2^2 + \frac{\gamma_1 w_l}{\nu}\|\mathbf{v}_l\|_q \\ &= \operatorname{prox}_{\sigma_l\|\cdot\|_q}(A_{i_1\cdot} - A_{i_2\cdot} - \nu^{-1}\boldsymbol{\lambda}_l).\end{aligned} \tag{10}$$

第三步，按照（11）更新 lambda。

$$\boldsymbol{\lambda}_l^m = \boldsymbol{\lambda}_l^{m-1} + \nu(\mathbf{v}_l^m - A_{i_1\cdot}^m + A_{i_2\cdot}^m). \tag{11}$$

整个算法流程图如下：

Algorithm 1 S-ADMM

1. Initialize $\mathbf{V}^0$ and $\boldsymbol{\Lambda}^0$. For $m = 1, 2, \ldots$
2. For $j = 1, \ldots, p$, do　P个次最优问题-A

$$\widetilde{\mathbf{v}}_l^{m-1} = \mathbf{v}_l^{m-1} + \frac{1}{\nu}\boldsymbol{\lambda}_l^{m-1}, l \in \mathcal{E}$$

$$\mathbf{y}_j^{m-1} = \mathbf{N}^{-1}\left(\mathbf{x}_j + \nu\sum_{l\in\mathcal{E}}\widetilde{v}_{lj}^{m-1}(\mathbf{e}_{i_1} - \mathbf{e}_{i_2})\right),$$

化为group-lasso问题

$$\mathbf{a}_j^m = \underset{\mathbf{a}_j}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y}_j^{m-1} - \mathbf{N}\mathbf{a}_j\|_2^2 + \gamma_2 u_j\|\mathbf{a}_j\|_2,$$

$$\mathbf{a}_j^m = \mathbf{a}_j^m - \bar{\mathbf{a}}_j^m\mathbf{1}_n, \text{ where } \bar{\mathbf{a}}_j^m = \mathbf{1}_n^T\mathbf{a}_j^m/n.$$

得到A0

3. For $l \in \mathcal{E}$, do

$$\mathbf{v}_l^m = \operatorname{prox}_{\sigma_l\|\cdot\|_q}(A_{i_1\cdot}^m - A_{i_2\cdot}^m - \nu^{-1}\boldsymbol{\lambda}_l^{m-1}).$$

4. For $l \in \mathcal{E}$, do

$$\boldsymbol{\lambda}_l^m = \boldsymbol{\lambda}_l^{m-1} + \nu(\mathbf{v}_l^m - A_{i_1\cdot}^m + A_{i_2\cdot}^m).$$

5. Repeat Steps 2–4 until convergence.

图 3-1 S-ADMM 算法流程图

3.2 S-AMA

S-AMA 可以显著提升计算效率。S-AMA 主要在优化 A 时有变化。v=0，从而 N=I，从而原损失函数（5）被简化。

$$\min_{\mathbf{a}_j} \frac{1}{2}\|\mathbf{x}_j - \mathbf{a}_j\|_2^2 + \gamma_2 u_j\|\mathbf{a}_j\|_2, \quad j = 1, \ldots, p. \tag{12}$$

根据 KKT 条件得到 a 的估计，证明在原文 supplementary。

$$\widehat{\mathbf{a}}_j = \left(1 - \frac{\gamma_2 u_j}{\|\mathbf{z}_j\|_2}\right)_+ \mathbf{z}_j, \tag{13}$$

对于这样的估计，显著减低了计算负担。如上估计 A 是与 V 独立的，所以不需要计算 V 的更新。从而能提升计算效率。整个算法流程图如下：

Algorithm 2 S-AMA

1. Initialize $\mathbf{\Lambda}^0$. For $m = 1, 2, \ldots$
2. For $j = 1, \ldots, p$, do 不需要更新V，与V独立

$$\mathbf{z}_j^m = \mathbf{x}_j + \sum_{l \in \mathcal{E}} \lambda_{lj}^{m-1} (\mathbf{e}_{i_1} - \mathbf{e}_{i_2}),$$

$$\mathbf{a}_j^m = \left(1 - \frac{\gamma_2 u_i}{\|\mathbf{z}_i^m\|_2}\right)_+ \mathbf{z}_j^m,$$

$$\mathbf{a}_j^m = \mathbf{a}_j^m - \bar{\mathbf{a}}_j^m \mathbf{1}_n, \text{ where } \bar{\mathbf{a}}_j^m = \mathbf{1}_n^T \mathbf{a}_j^m / n.$$

3. For $l \in \mathcal{E}$, do

$$\boldsymbol{\lambda}_l^m = \mathcal{P}_{C_l}[\boldsymbol{\lambda}_l^{m-1} - \nu(A_{i_1\cdot}^m - A_{i_2\cdot}^m)],$$

where $C_l = \{\boldsymbol{\lambda}_l : \|\boldsymbol{\lambda}_l\|_\dagger \le \gamma_1 w_l\}$.

4. Repeat Steps 2–3 until convergence.

图 3-2 S-AMA 算法流程图

3.3 算法收敛

参考文献中提供了一般优化问题收敛的充分条件。两种算法可以看做（8）的特例。其证明保证了算法收敛。

$$\min_{\xi,\zeta} \ f(\xi) + g(\zeta), \ \text{ s.t. } A\xi + B\zeta = c. \tag{14}$$

4 数值实验

4.1 度量

在对于观测值的分类准确率方面我们采取兰德系数（RAND Index），兰德系数的计算过程如图 4-1 所示，具体可参阅[2]。兰德系数越大越好。

TP：同一类的文章被分到同一个簇

TN：不同类的文章被分到不同簇

FP：不同类的文章被分到同一个簇

FN：同一类的文章被分到不同簇

Rand Index度量的正确的百分比

RI = （TP+TN）/（TP+FP+FN+TN）

图 4-1 兰德系数的计算过程

在变量筛选方面，我们采取假阴性率和假阳性率作为评价指标。即下图中 FP 和 FN 所占比例。这两个指标都是越小越好。

		Test Result	
		0	1
Ground Truth	0	TN (True Negative)	FP (False Positive)
	1	FN (False Negative)	TP (True Positive)

图 4-2 假阴性率和假阳性率

4.2 生成数据集

主要实现本文算法 S-ADMM,S-AMA 与 k-means 算法，ADMM,AMA 算法的对比。第一组数据集设置：考虑了四个球面设置。有 60 个观测值，聚类数量为 2 或 4，特征的数量为 150 或 500。只有前 20 个特征是含信息的。（含信息）数据产生于多元正态分布，（非信息）数据产生于标准正态分布。

第二组数据集是一个双月牙数据集，只有前两维特征有信息，其余 38 个特征都是非信息特征。分类相对比较困难，因为无信息特征是信息特征的 19 倍。

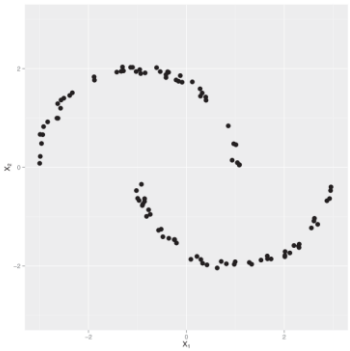

图 4-3 双月牙数据集

具体实验结果如下:

		RAND		FNR		FPR	
	Algorithm	mean	SD	mean	SD	mean	SD
Setting 1	*k*-means	0.95	0.06	0.00	0.00	1.00	0.00
	ADMM	0.53	0.39	0.00	0.00	1.00	0.00
	AMA	0.66	0.40	0.00	0.00	1.00	0.00
	S-ADMM	0.82	0.24	0.04	0.05	0.25	0.16
	S-AMA	**0.96**	0.06	0.03	0.07	0.30	0.21
Setting 2	*k*-means	0.95	0.11	0.00	0.00	1.00	0.00
	ADMM	0.14	0.20	0.00	0.00	1.00	0.00
	AMA	0.08	0.21	0.00	0.00	1.00	0.00
	S-AMA	**0.97**	0.07	0.07	0.09	0.11	0.10
Setting 3	*k*-means	0.83	0.15	0.00	0.00	1.00	0.00
	ADMM	0.56	0.22	0.00	0.00	1.00	0.00
	AMA	0.47	0.21	0.00	0.00	1.00	0.00
	S-ADMM	0.82	0.14	0.04	0.06	0.25	0.24
	S-AMA	**0.84**	0.13	0.02	0.04	0.11	0.18
Setting 4	*k*-means	0.89	0.14	0.00	0.00	1.00	0.00
	ADMM	0.31	0.23	0.00	0.00	1.00	0.00
	AMA	0.31	0.20	0.00	0.00	1.00	0.00
	S-AMA	**0.94**	0.09	0.01	0.02	0.01	0.03
Setting 5	*k*-means	0.51	0.07	0.00	0.00	1.00	0.00
	ADMM	0.54	0.08	0.00	0.00	1.00	0.00
	AMA	0.53	0.09	0.00	0.00	1.00	0.00
	S-AMA	**0.57**	0.07	0.00	0.00	0.34	0.27
	SPECC	0.52	0.08	0.00	0.00	1.00	0.00

图 4-4 生成数据集实验结果

实验结果表明，1 凸聚类在高维情形性能糟糕 2 稀疏凸聚类性能很好还能同时筛选特征，并且很鲁棒。

4.3 真实数据集

接下来，我们考虑在手部运动数据集上的应用。数据集包含 15 个类（每一类是一种手部运动），每类包含了 24 个样本，有 90 个特征。由于一些 cluster 重叠，所以筛选了六个区分度更大的类。用 PCA 的前两个主成分画图。

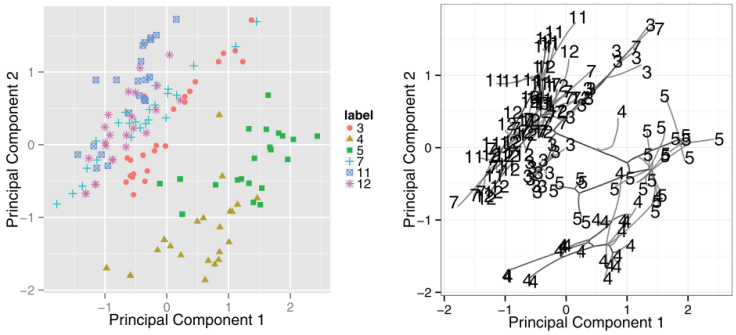

图 4-5 手部运动数据集 PCA 主成分画图/AMA 算法分类结果

AMA 算法展示的结果说明了高维特征的诅咒，即只能将其近似分为两个类。而我们采用 S-AMA 算法进行处理后，S-AMA 算法能将多个类分开，并且只用了 13 个含信息特征。

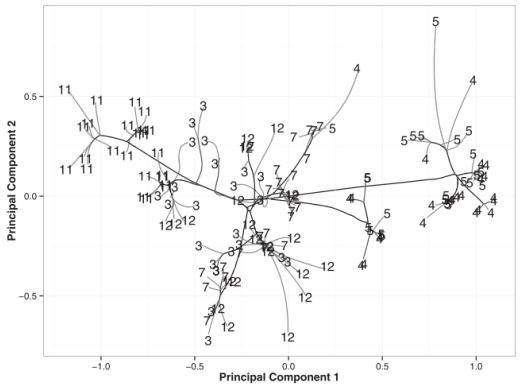

图 4-6 S-AMA 算法分类结果

S-AMA 算法最后将其分为三类，和真实标签对比如下图:

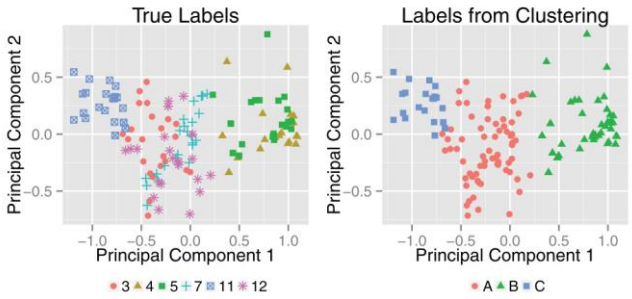

图 4-7 真实标签/S-AMA 算法分类结果

评价指标展示了 S-AMA 取得了最好的聚类效果，并且只用了 13 个含信息特征。

Algorithm	No. of clusters	No. of features	RAND index
k-means	2	90	0.06
AMA	3	90	0.31
S-AMA	3	13	**0.45**

图 4-8 评价指标对比

5 复现代码

稀疏凸聚类的原作者将稀疏凸聚类算法制作为 R 语言包 scvxclustr[6]，本文的复现也是基于其提供的的 R 语言包 scvxclustr。本文的复现代码可以在我的 Github[3]上获取：
https://github.com/stxupengyu/sparse-convex-clustering-demonstration

5.1 生成数据集

首先，我们生成一个数据集。含有 80 个特征和 60 个样本，只有前 20 个特征是有信息的，服从于多元正态分布，后面的特征来源于标准正态分布。共有四个聚类中心。训练集的热力图如下：

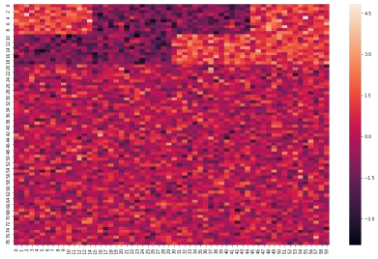

图 5-1 生成数据集热力图

可以看到其只有前 20 行含信息，并且明显可以看出有四个类。
首先，我们使用凸聚类方法，得到的特征矩阵如下：

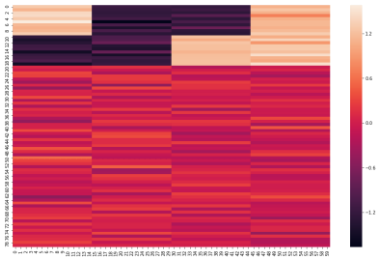

图 5-2 AMA 特征矩阵 A 热力图

可以看到其列向量共有四种模式，说明其将观测值聚为四类。
稀疏凸聚类的特征矩阵 A 如下，可以看出其非信息特征全部压缩为 0。并且其列向量也是共有四种模式，说明其将观测值聚为四类。

图 5-3 S-AMA 特征矩阵 A 热力图

计算其评价指标，发现两种算法在生成数据集上都获得了最好的 RAND Index，即等于 1。

5.2 真实数据集

接下来，我们在真实数据集上进行实现。我们所选取的数据集[4]与原文一样，即手部运动数据集。其包含 15 个类（每一类是一种手部运动），每类包含了 24 个样本，有 90 个特征。与原文一致，筛选了六个区分度更大的类。

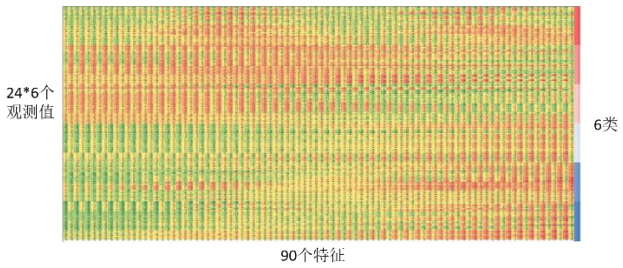

图 5-4 真实数据集热力图

训练集如下所示，其中每一行是一个特征，每一列是一个观测值：

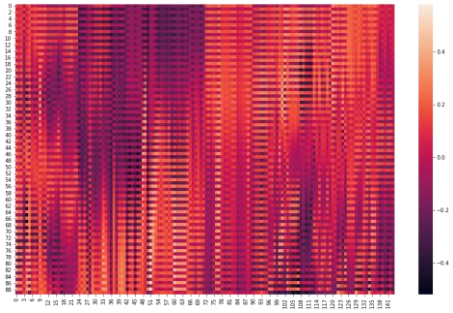

图 5-5 训练集热力图

可以看到其每一个类还是有一定的规律，但是无法区分出有多少个类。

首先，我们使用凸聚类方法，得到的特征矩阵如下：

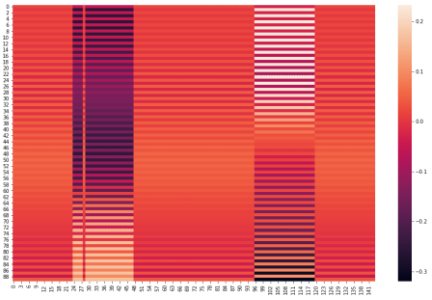

图 5-6 AMA 特征矩阵 A 热力图

可以看到其列向量共有三种模式，说明其将观测值聚为三类。

稀疏凸聚类的特征矩阵 A 如下，可以看出其非信息特征全部压缩为 0。并且其列向量共有四种模式（即白色、橘色、黑色、紫色），说明其将观测值聚为四类。

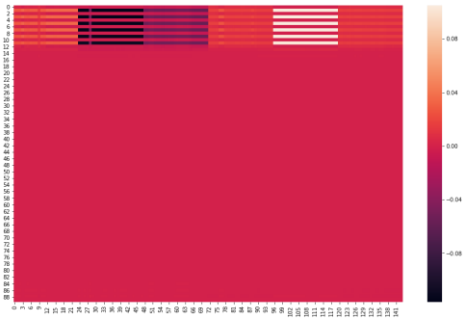

图 5-7 S-AMA 特征矩阵 A 热力图

计算其评价指标。发现 S-AMA 方法在聚类数量和兰德系数上都要好于 AMA 方法。两种方法效果都要好于原文的实验结果，可能的原因是我们通过调获得了更好的参数。

方法	聚类数量	RAND-Index
AMA	3	0.653
S-AMA	4	0.779
真实数据	6	1

图 5-8 评价指标对比

6 引证文献

6.1 Localized Lasso for High-Dimensional Regression[5]

6.2 摘要

这篇文章提出的 Localized Lasso 模型是在高维度，低样本数下，可解释，高预测精度的模型。其损失函数考虑到了本地稀疏模型，并且加入变量选择。本地化模型是可以对稀疏形式进行解释的。损失函数是凸的，因此具有全局最优解。其次提出了一种不需要调参的优化方法。

6.3 和稀疏凸聚类的关系

基于稀疏性的全局特征选择方法如 Lasso 对基因选择是有用的。然而，在个性化医疗设置中，我们最终希望个性化每个病人(或药物)的模型，而不是为每个人假设相同的特征集(例如，基因)。该任务需要筛选特征，类似稀疏凸聚类。

6.4 与稀疏凸聚类对比

下图展示了多种算法特征矩阵 A 热力图对比，如我们看到的，network lasso 不是稀疏的。但是加上文章提出的正则项，其就可以捕获到正确的稀疏模式。稀疏凸聚类方法可以筛选到正确的全局含信息特征，但是在局部特征集合上精确度不如本文提出的模型。

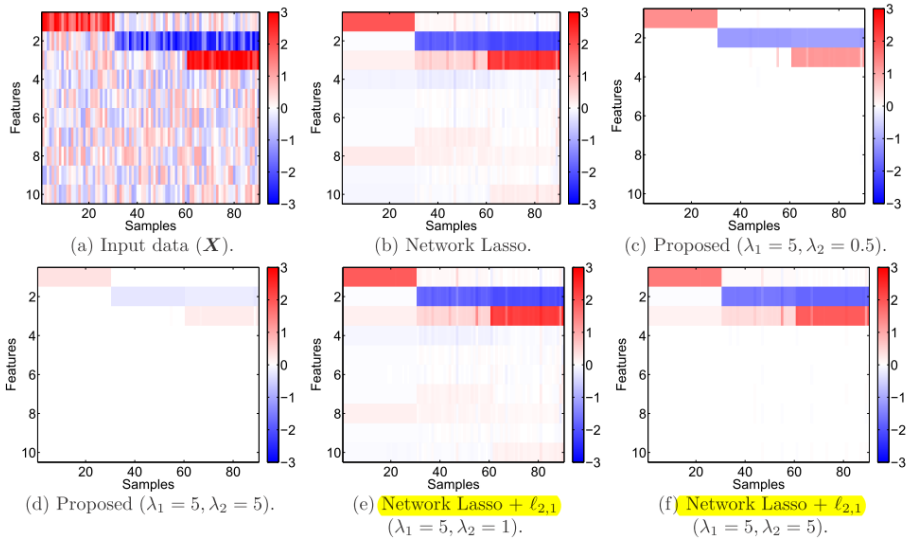

图 6-1 多种算法特征矩阵 A 热力图对比

参考文献

[1]Wang, Binhuan, et al. “Sparse Convex Clustering.” *Journal of Computational and Graphical Statistics*, vol. 27, no. 2, 2018, pp. 393–403.

[2]https://stats.stackexchange.com/questions/89030/rand-index-calculation

[3]https://github.com/stxupengyu/sparse-convex-clustering-demonstration

[4]https://archive.ics.uci.edu/ml/machine-learning-databases/libras/

[5]Yamada, Makoto, et al. “Localized Lasso for High-Dimensional Regression.” International Conference on Artificial Intelligence and Statistics, 2017, pp. 325–333.

[6]https://github.com/elong0527/scvxclustr