PR homework11

李逸思 自动化系 2016310707

Clustering algorithm

算法描述

K means

1. 随机初始化样本中心点

重复以下步骤，迭代直至收敛：

Step1

$$c^{(i)} = \arg\min_j \left\| x^{(i)} - \mu_j \right\|^2$$

Step2

$$\mu_j = \frac{\sum_{i=1}^{m} I\left(c^{(i)} = j\right) x^{(i)}}{\sum_{i=1}^{m} I\left(c^{(i)} = j\right)}$$

Hierarchical clustering

1. 初始情况下，每个样本点为一个类别，计算类类之间的相似度
2. 在各类之间找到最近的两个类，把它们归为一类（总类别数减少一）
3. 重新计算新生成的类与原有旧的类别之间的相似度
4. 重复 2、3 步骤直至所有样本被归为一类

Spectral clustering

1. 输入待分类数据和类别数 k
2. 对待分类数据计算两两间欧氏距离，构造相似性图，得到加权邻接矩阵 W，计算度矩阵 D

$$D = diag\left(d_1, d_2, ...d_n\right)，\ 其中 d_i = \sum_{j=1}^{n} w_{ij}$$

3. 计算 Graph Lapacian 矩阵

未归一化：$L=D-W$

归一化：$L=I-D^{-1}W$

4. 计算 L 的前 k 个特征向量 $u_1,u_2,\ldots u_k$,并由列向量 $u_1,u_2,\ldots u_k$ 构成矩阵 U
5. 设 y_i 是 U 的第 i 行构成的向量，使用 kmeans 聚类方法将点 y_i 聚为 k 类，$C_1,..\ C_k$
6. 输出最终聚类 $A_1,..\ A_k$,其中

$$A_i=\{j|y_j\in C_i\}$$

仿真实验

三种算法的 matlab 实现见附件代码。从 MNIST 数据集中取 1000 条数据（包含 0-9 每种手写数字各 100 条数据），对三种聚类算法进行测试。

2.1 三种算法时间复杂度分析如下：

Algorithm	Time complexity
k-means	对 $c^{(i)}=\arg\min_j\left\|x^{(i)}-\mu_j\right\|^2$ 的优化是 NP-hard 问题，但是对于给定的迭代次数 i，设将 n 个待分类样本点分为 k 类，则算法复杂度为 O(nki)
Hierarchical clustering	$O\left(n^2\log(n)\right)$
Spectral clustering	计算特征向量算法复杂度为 $O\left(n^3\right)$，调用 kmeans 算法复杂度为 O(nki)，则总体算法复杂度为 $O\left(n^3\right)$

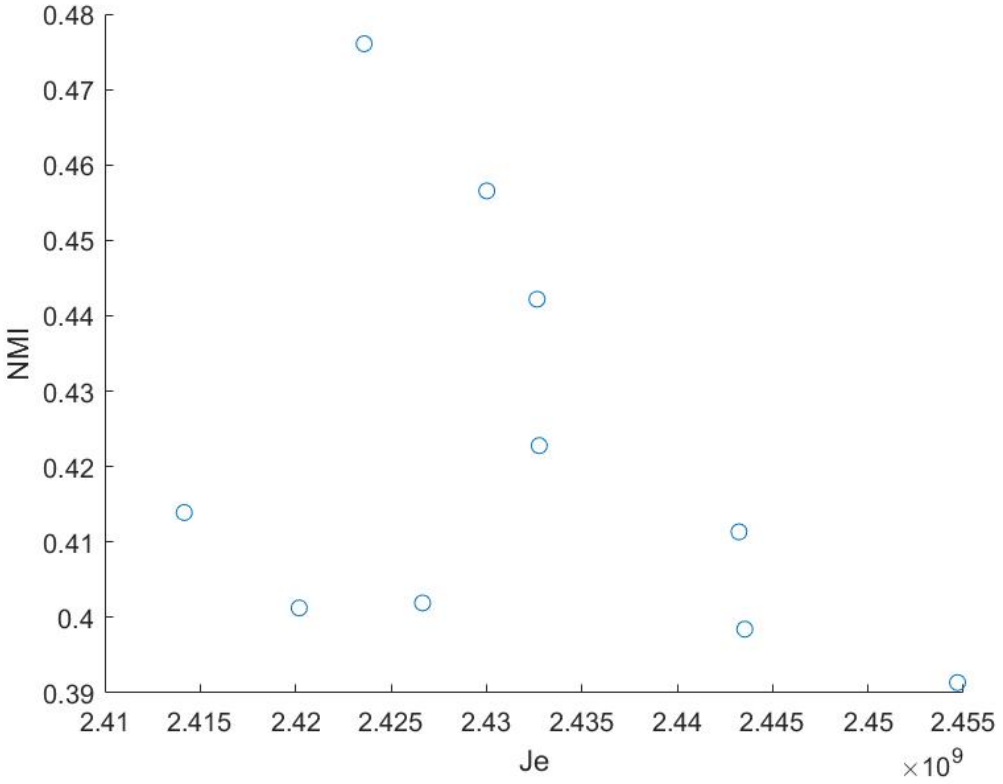
由以上时间复杂度可知， Hierarchical clustering 和 Spectral clustering 时间复杂度较高，大样本数据下运行较慢。

Matlab 仿真测试，采用相同数据，相同类别参数，三种算法运行时间如下：

	k-means	Hierarchical clustering	Spectral clustering
运行时间(s)	1.244538	4.318186	1.792931

2.2 将类别参数设为实际类别数 C=10，进行仿真，有：

2.2.1 对于 kmeans ，初始化结果会影响聚类结果，这是因为 kmeans 是一种局部寻优的贪婪算法，我们应采用多次试验取最优的方法来避免算法陷入局部最优。仿真 10 次，得到 J_e 和 NMI 如下：



可以看到，总体而言，J_e越大，NMI 越小。这是因为 NMI 越大说明聚类结果和实际越接近，J_e越小说明目标函数优化情况越好，也说明了聚类结果更好，因此 J_e和 NMI 呈负相关关系

2.2.2 对于 Hierarchical clustering，如何度量类类之间的相似度是一个关键问题，常见的度量方法有，

最近距离（single linkage）：$\Delta\left(\Gamma_i,\Gamma_j\right)=\min\limits_{\substack{y\in\Gamma_i\\ \tilde{y}\in\Gamma_j}}\delta\left(y,\tilde{y}\right)$

最远距离（complete linkage）：$\Delta\left(\Gamma_i,\Gamma_j\right)=\max\limits_{\substack{y\in\Gamma_i\\ \tilde{y}\in\Gamma_j}}\delta\left(y,\tilde{y}\right)$

均值距离（group average）：$\Delta\left(\Gamma_i,\Gamma_j\right)=\min\delta\left(\mathrm{m}_i,\mathrm{m}_j\right)$

其中，single linkage 由于用两类中点的最近距离衡量两个类别的相似度，导致两个类可能整体相距较远，但个别点相距较近从而被合并，最终会得到较为松散的 cluster；complete linkage 由于用两类中点的最近距离衡量两个类别的相似度，导致两个类可能整体非常相似，但个别点相距较远从而不能被合并；group average 对两个类中点的两两距离取平均，得到的结果相对 single linkage complete linkage 和不那么极端，但计算量较前两者大。三种度量方法各有优劣，依据不同问题选择不同度量方法会有不同效果。

这里用最近距离、最远距离和均值距离分别对测试数据进行聚类，得到三种方法的 NMI 值如下：

	Single linkage	Complete linkage	Group linkage
NMI	0.0517	0.3057	0.2235

可以看到，以 NMI 作为评价指标，complete linkage 最优。

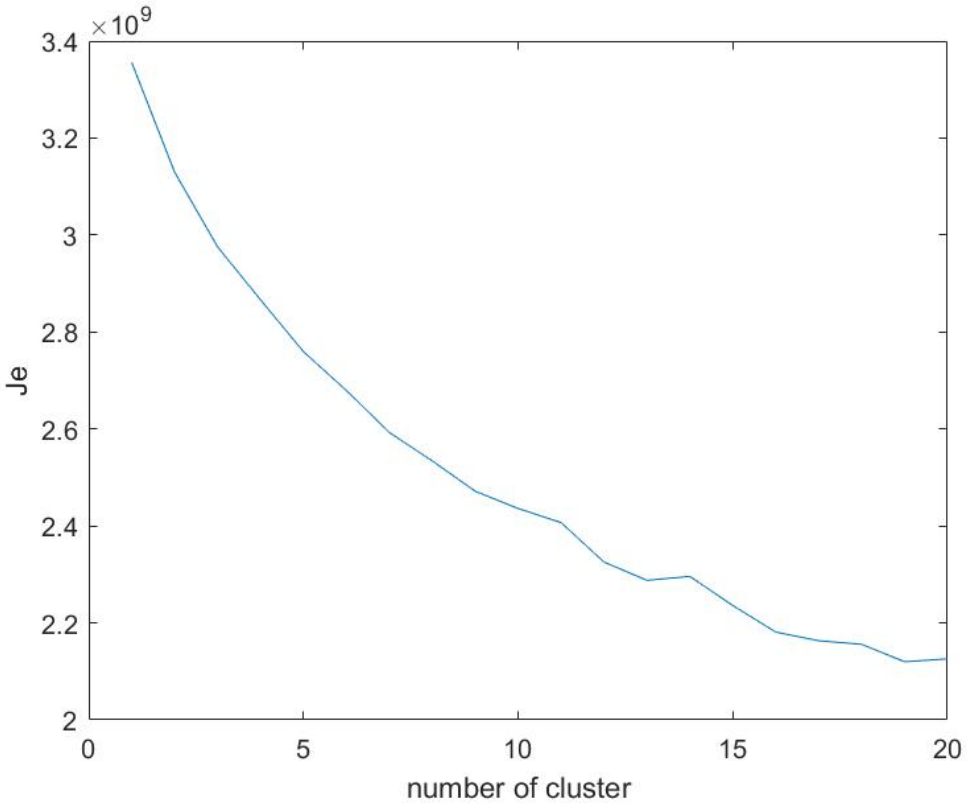
2.2.3 分别采用“euclidean”，“cityblock”，“minkowski”三种距离来度量相似性，计算 Graph Lapacian 矩阵 L 时分别采用归一化的计算方法和未归一化的计算方法，得到 NMI 值如下：

	euclidean	cityblock	minkowski
未归一化的 L	0.0271	0.0276	0.0303
归一化的 L	0.3592	0.3606	0.3352

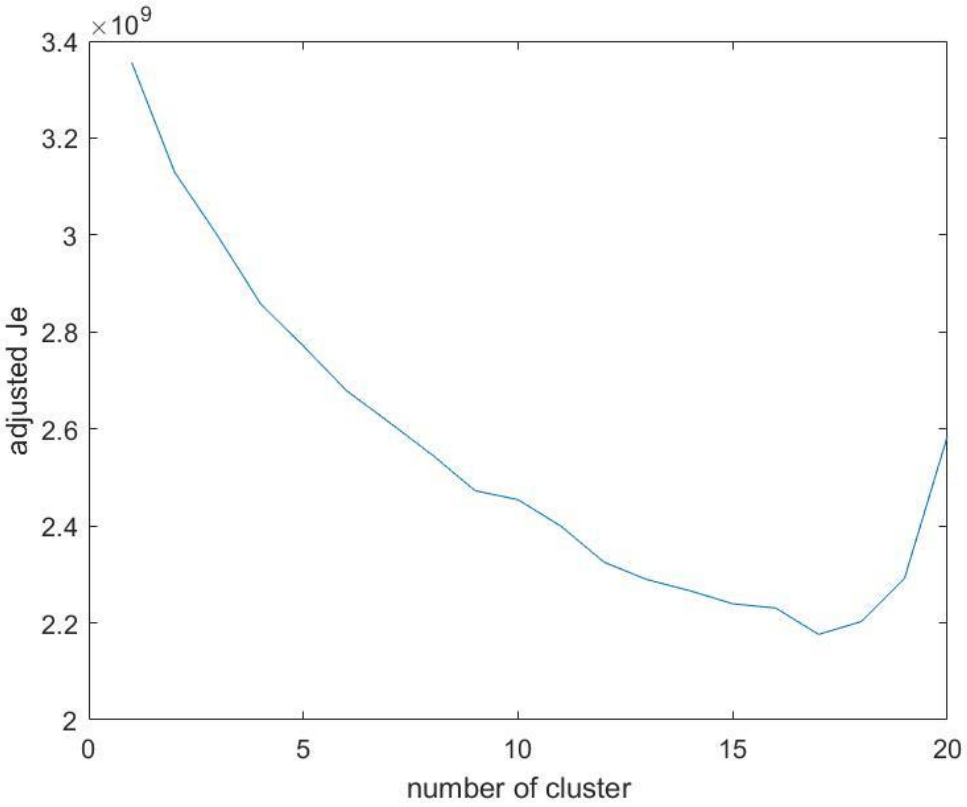
以 NMI 值作为聚类结果评价指标，以上度量中，采用归一化的 L 并采用 cityblock distance 度量相似性得到的聚类结果最好。

2.3 实际情况下，我们不知道数据的实际类别数，这时可以将类别数 C 从小到大取值，考察不同取值下的分类情况。

以 kmeans 为例，取类别数 1~20 时，考察 J_e 随类别数 C 的变化如下图：



分类类别数越多则平方误差越小，但是过大的 C 值使得聚类没有实际意义，因此我们引入对过大 C 值的惩罚项，定义 $L=J_e+\exp(aC)$（其中 a 为常系数项），考察 L 随类别数 C 的变化，例如取 a=1 时，得到 L 随类别数 C 的变化如下图：



从图中看到当 a 取 1 时，类别数取 16 是最优选择。

对于上述的确定类别数的方法，常系数项 a 的选择对类别数的确定有着重要影响，为了找出合适的 a 的取值，在有训练数据的情况下，可以先根据已知类别数的训练数据确定 a 的取值，然后由此 a 值对测试数据确定类别数。

2.4 三种聚类方法的聚类结果和理论聚类结果分别如下：

聚类算法		NMI	运行时间
实际情况		1	~
k-means		0.4487	0.8s
Hierarchical clustering	Group linkage	0.2235	8.9s
	Single linkage	0.0517	4.4s
	Complete linkage	0.3057	4.3s
Spectral clustering	euclidean	0.4086	1.7s
	cityblock	0.3926	1.5s
	minkowski	0.3743	1.6s

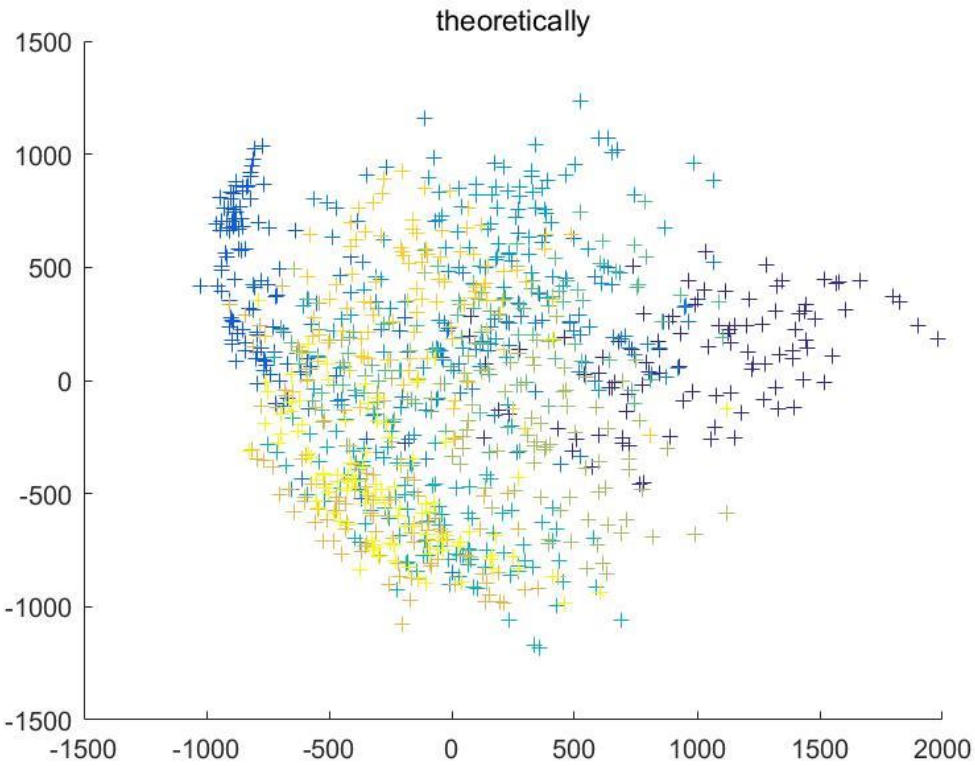


图 2.4.1 理论聚类结果

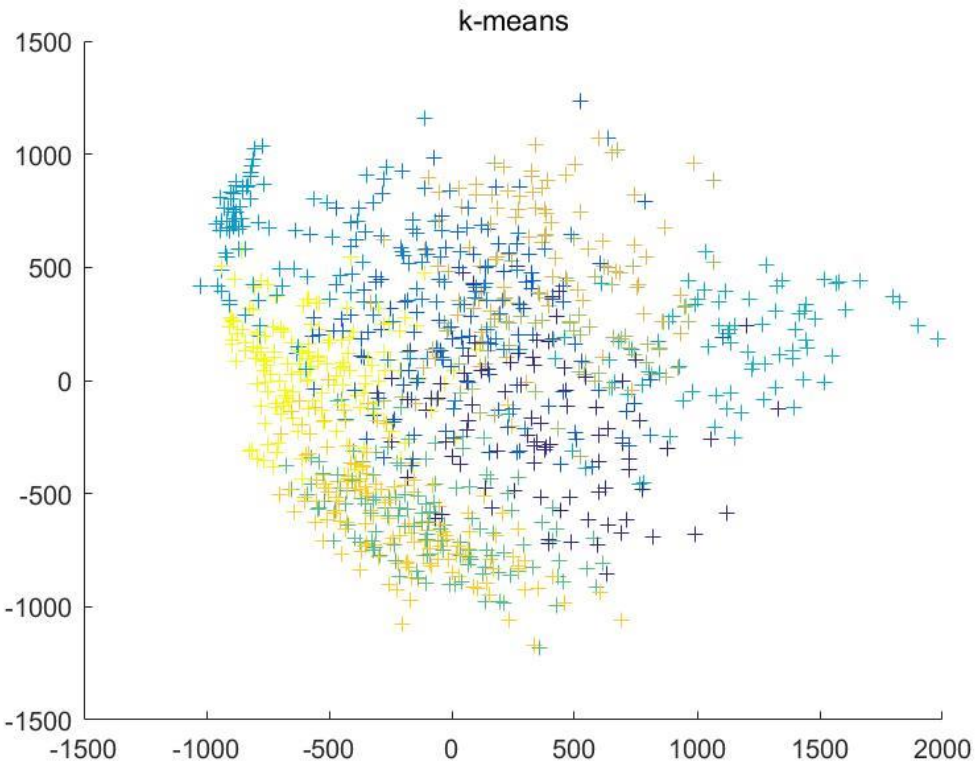
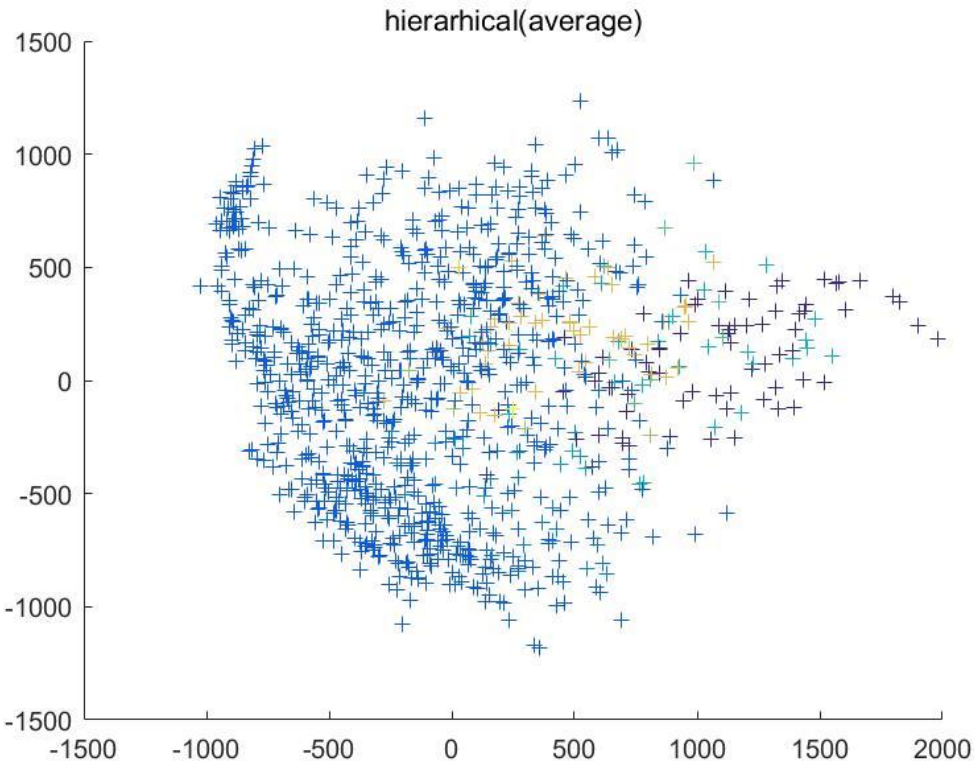
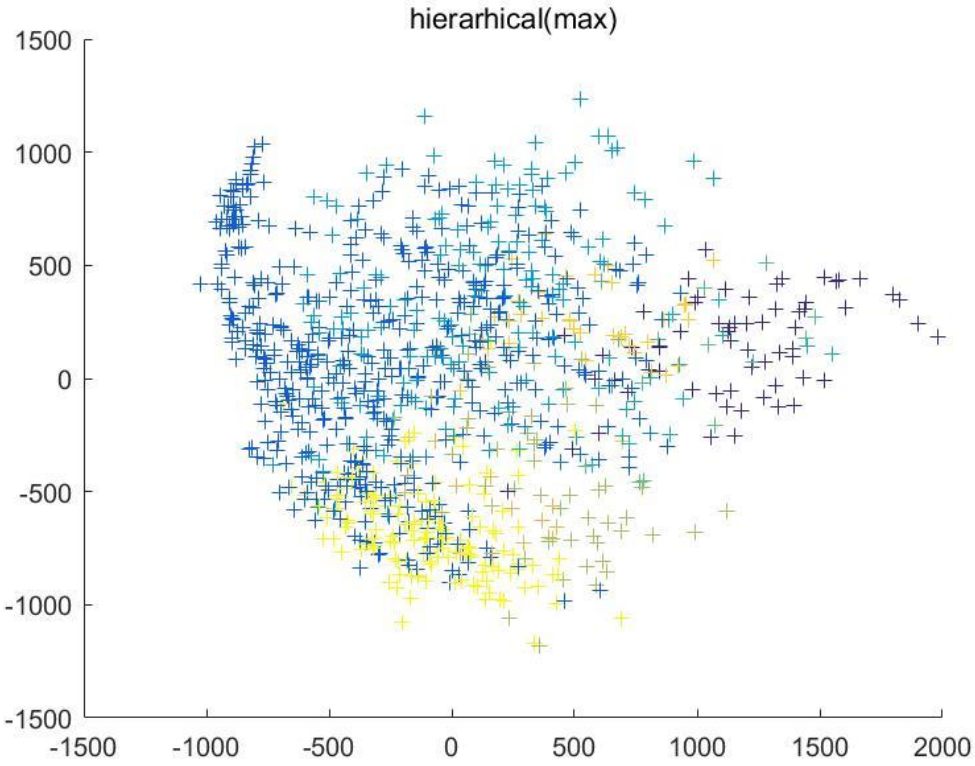


图 2.4.2 k-means 聚类结果


hierarhical(average)
1500
1000
500
0
-500
-1000
-1500
-1500
-1000
-500
0
500
1000
1500
2000


hierarhical(max)
1500
1000
500
0
-500
-1000
-1500
-1500
-1000
-500
0
500
1000
1500
2000

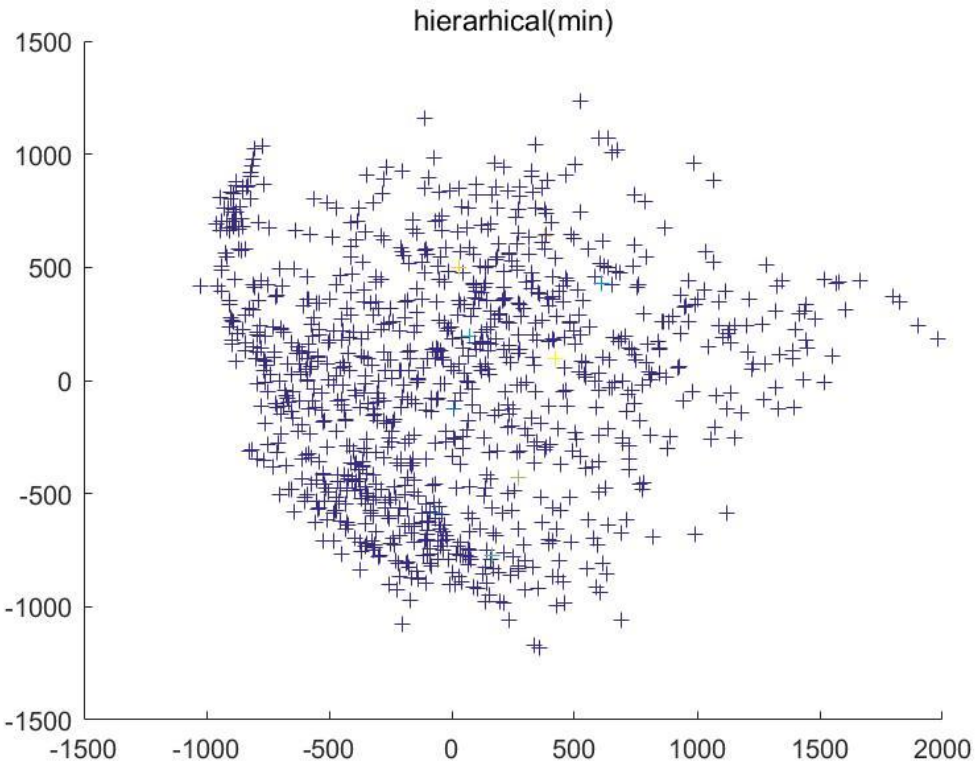
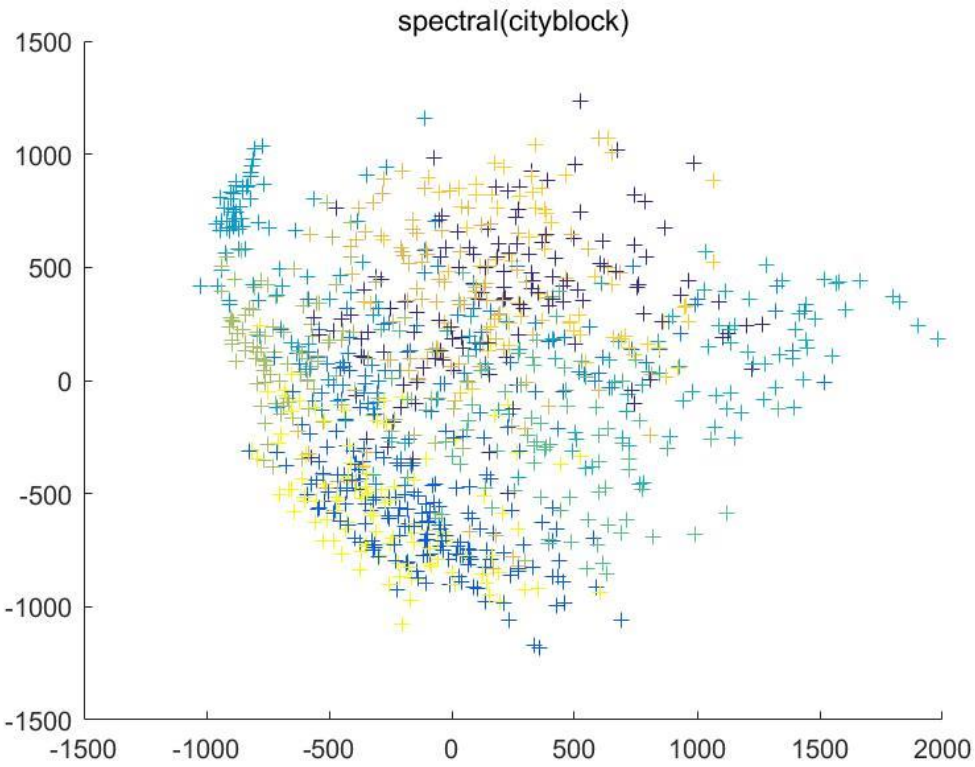
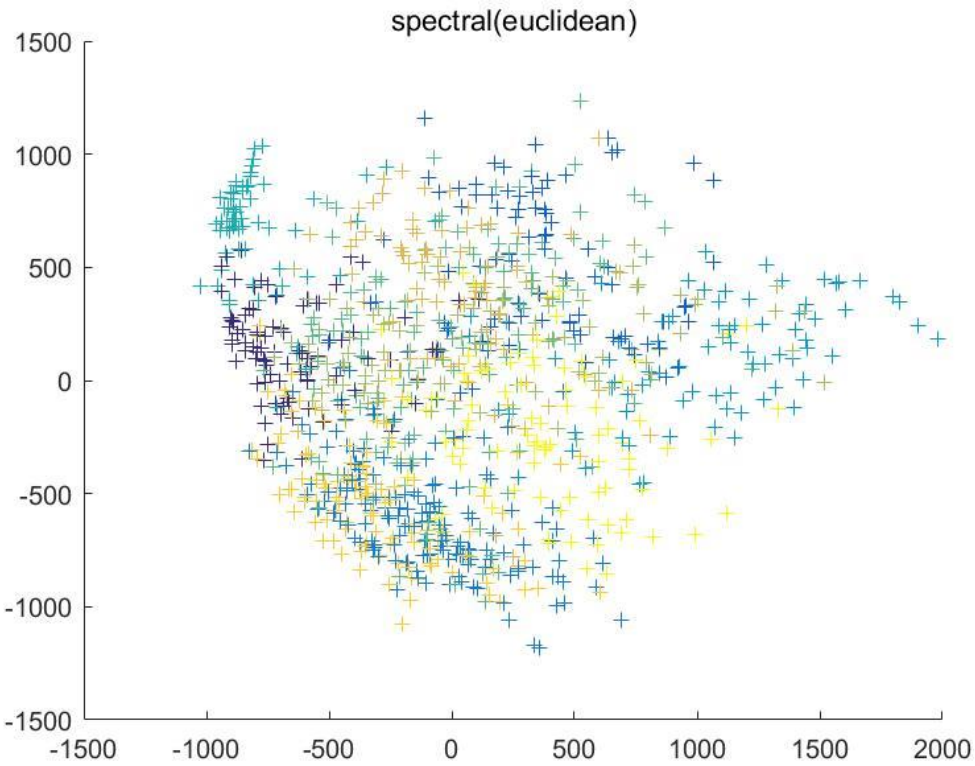


图 2.4.3 hierarchical clustering 聚类结果





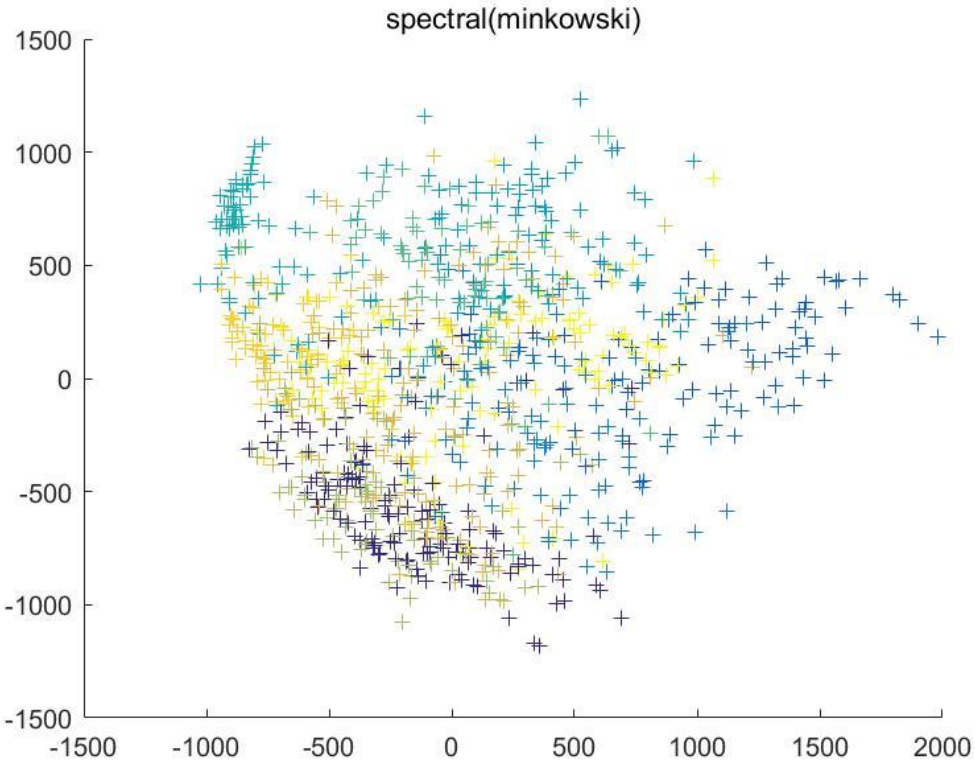


图 2.4.4 spectral clustering 聚类结果

综合聚类结果与实际情况的接近程度以及算法运行时间，我认为 k-means 是一种很好的分类算法，算法实现复杂度较低，算法时间复杂度也较低，聚类结果也较好。此外，谱聚类也是一种很好的聚类方法，从算法实现难度、算法时间复杂度和聚类结果上都有较好的性能。