法律声明

- 本课件包括演示文稿、示例、代码、题库、视频和声音等内容，小象学院和主讲老师拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意及内容，我们保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：小象
  - 新浪微博：ChinaHadoop

聚类实践

小象学院 ChinaHadoop.cn 邹博

本次目标

- 谱聚类的算法
  - 考虑谱聚类和PCA的关系
- 聚类代码剖析

复习：实对称阵的特征值是实数

- 首先 $A\overline{x} = \overline{A}\overline{x} = \overline{Ax} = \overline{\lambda x} = \overline{\lambda}\overline{x}$
- 因为 $$\overline{x}^T(Ax) = \overline{x}^T(Ax) = \overline{x}^T\lambda x = \lambda\overline{x}^T x$$
$$\overline{x}^T(Ax) = \left(\overline{x}^T A^T\right)x = (A\overline{x})^T x = \left(\overline{\lambda}\overline{x}\right)^T x = \overline{\lambda}\overline{x}^T x$$
- 从而
$$\lambda\overline{x}^T x = \overline{\lambda}\overline{x}^T x \Rightarrow \left(\lambda - \overline{\lambda}\right)\overline{x}^T x = 0$$
- 而
$$\overline{x}^T x = \sum_{i=1}^{n} \overline{x_i} x_i = \sum_{i=1}^{n} |x_i|^2 \neq 0$$
- 所以
$$\lambda - \overline{\lambda} = 0 \Rightarrow \lambda = \overline{\lambda}$$

实对称阵不同特征值的特征向量正交

- 令实对称矩阵为A，其两个不同的特征值$\lambda_1\lambda_2$对应的特征向量分别是$\mu_1\mu_2$；
  - $\lambda_1\lambda_2\ \mu_1\mu_2$都是实数或是实向量。

$$\begin{cases} A\mu_1 = \lambda_1\mu_1 \\ A\mu_2 = \lambda_2\mu_2 \end{cases} \Rightarrow \mu_1^T \underline{A\mu_2} = \mu_1^T \underline{\lambda_2\mu_2}$$

$$\Rightarrow \left(A^T\mu_1\right)^T \mu_2 = \lambda_2\mu_1^T\mu_2 \Rightarrow \left(\underline{A\mu_1}\right)^T \mu_2 = \lambda_2\mu_1^T\mu_2$$

$$\Rightarrow \left(\underline{\lambda_1\mu_1}\right)^T \mu_2 = \lambda_2\mu_1^T\mu_2$$

$$\Rightarrow \lambda_1\mu_1^T\mu_2 = \lambda_2\mu_1^T\mu_2$$

$$\xrightarrow{\lambda_1 \neq \lambda_2} \mu_1^T\mu_2 = 0$$

谱和谱聚类

- 方阵作为线性算子，它的所有特征值的全体统称方阵的**谱**。
  - 方阵的谱半径为最大的特征值
  - 矩阵A的**谱半径**：(A^TA)的最大特征值
- 谱聚类是一种基于图论的聚类方法，通过对样本数据的**拉普拉斯矩阵**的**特征向量**进行聚类，从而达到对样本数据聚类的目的。

谱分析的整体过程

- 给定一组数据$x_1, x_2, \ldots x_n$，记任意两个点之间的相似度(“距离”的减函数)为$s_{ij}=\langle xi, xj \rangle$，形成相似度图(similarity graph)：G=(V,E) 。如果x_i和x_j之间的相似度s_{ij}大于一定的阈值，那么，两个点是连接的，权值记做s_{ij}。
- 接下来，可以用相似度图来解决样本数据的聚类问题：找到图的一个划分，形成若干个组(Group)，使得不同组之间有较低的权值，组内有较高的权值。

若干概念

- 无向图G=(V,E)
- 邻接矩阵$W=(w_{ij})_{i,j=1,\ldots,n}$
- 顶点的度di → 度矩阵D (对角阵)

$$d_i=\sum_{j=1}^{n} w_{ij}$$

若干概念

- 子图A的指示向量

$$\mathbb{1}_A = (f_1, \ldots, f_n)' \in \mathbb{R}^n$$
$$f_i = 1 \text{ if } v_i \in A$$
$$f_i = 0 \text{ otherwise}$$

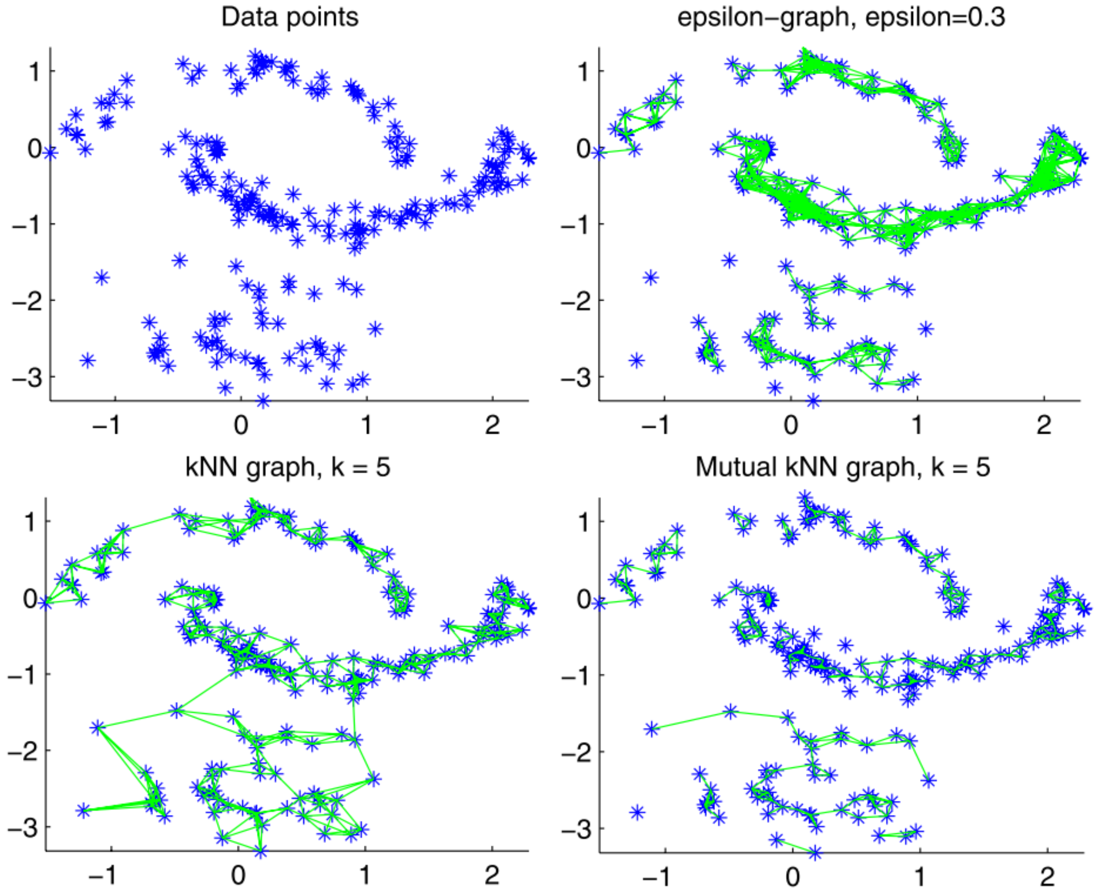
- A和B是图G的不相交子图，则定义子图的连接权：

$$W(A, B) := \sum_{i \in A, j \in B} w_{ij}$$

相似度图G的建立方法

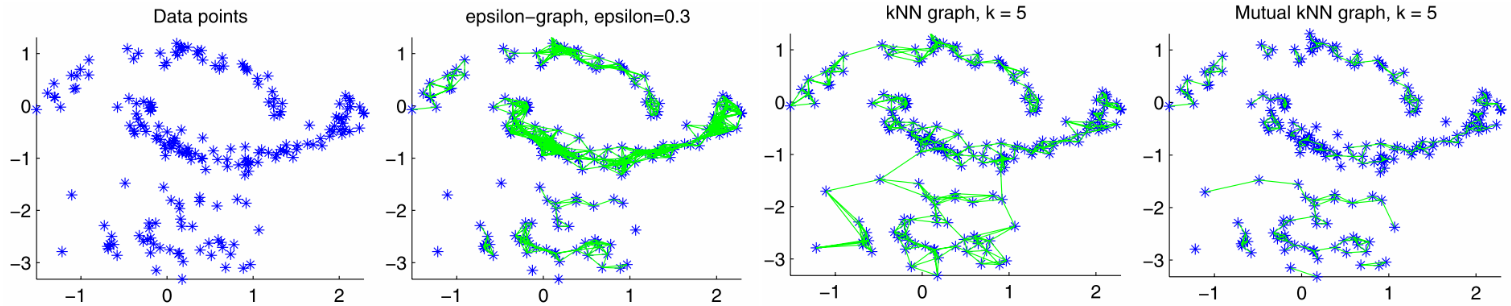
- 全连接图：距离越大，相似度越小
  - 高斯相似度 $s(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$
- ε近邻图
  - 给定参数ε/如何选择ε?
    - 图G的权值的均值
    - 图G的最小生成树的最大边
- k近邻图(k-nearest neighbor graph)
  - 若vi的k最近邻包含vj，vj的k最近邻不一定包含vi：有向图
  - 忽略方向的图，往往简称“k近邻图”
  - 两者都满足才连接的图，称作“互k近邻图(mutual)”

相似度图G的举例



Data points
epsilon−graph, epsilon=0.3
kNN graph, k = 5
Mutual kNN graph, k = 5

权值比较



- ε近邻图：ε=0.3，“月牙部分”非常紧的连接了，但“高斯部分”很多都没连接。当数据有不同的“密度”时，往往发生这种问题。
- k近邻图：可以解决数据存在不同密度时有些无法连接的问题，甚至低密度的“高斯部分”与高密度的“月牙部分”也能够连接。同时，虽然两个“月牙部分”的距离比较近，但k近邻还可以把它们区分开。
- 互k近邻图：它趋向于连接相同密度的部分，而不连接不同密度的部分。这种性质介于ε近邻图和k近邻图之间。如果需要聚类不同的密度，这个性质非常有用。
- 全连接图：使用高斯相似度函数可以很好的建立权值矩阵。但缺点是建立的矩阵不是稀疏的。
- 总结：首先尝试使用k近邻图。

拉普拉斯矩阵及其性质

- 拉普拉斯矩阵：$L = D - W$

$$f^T L f = f^T D f - f^T W f = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

$$= \frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} d_j f_j^2\right)$$

$$= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\left(f_i - f_j\right)^2$$

- L是对称半正定矩阵，最小特征值是0，相应的特征向量是全1向量。

拉普拉斯矩阵的定义

- 计算点之间的邻接相似度矩阵W
  - 若两个点的相似度值越大，表示这两个点越相似；
  - 同时，定义$w_{ij}=0$表示v_i,v_j两个点没有任何相似性(无穷远)
- W的第i行元素的和为v_i的度。形成顶点度对角阵D
  - d_{ii}表示第i个点的度
  - 除主对角线元素，D其他位置为0
- 未正则的拉普拉斯矩阵：$L = D - W$
- 正则拉普拉斯矩阵
  - 对称拉普拉斯矩阵 $L_{sym} = D^{-\frac{1}{2}} \cdot L \cdot D^{\frac{1}{2}} = I - D^{-\frac{1}{2}} \cdot W \cdot D^{\frac{1}{2}}$
  - 随机游走拉普拉斯矩阵 $L_{rw} = D^{-1}L = I - D^{-1}W$
    - Random walk

谱聚类算法：未正则拉普拉斯矩阵

- 输入：n个点$\{p_i\}$，簇的数目k
  - 计算$n \times n$的相似度矩阵W和度矩阵D；
  - 计算拉普拉斯矩阵L=D-W；
  - 计算L的前k个特征向量$u_1,u_2,\ldots,u_k$；
  - 将k个列向量$u_1,u_2,\ldots,u_k$组成矩阵U，$U \in R^{n \times k}$；
  - 对于i=1,2,...,n,令$y_i \in R^k$是U的第i行的向量；
  - 使用k-means算法将点$(y_i)_{i=1,2,\ldots,n}$聚类成簇$C_1,C_2,\ldots C_k$；
  - 输出簇$A_1,A_2,\ldots A_k$，其中，$A_i=\{j|y_j \in C_i\}$

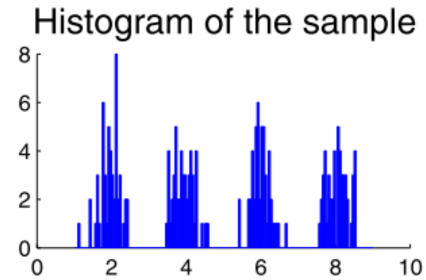
谱聚类算法：随机游走拉普拉斯矩阵

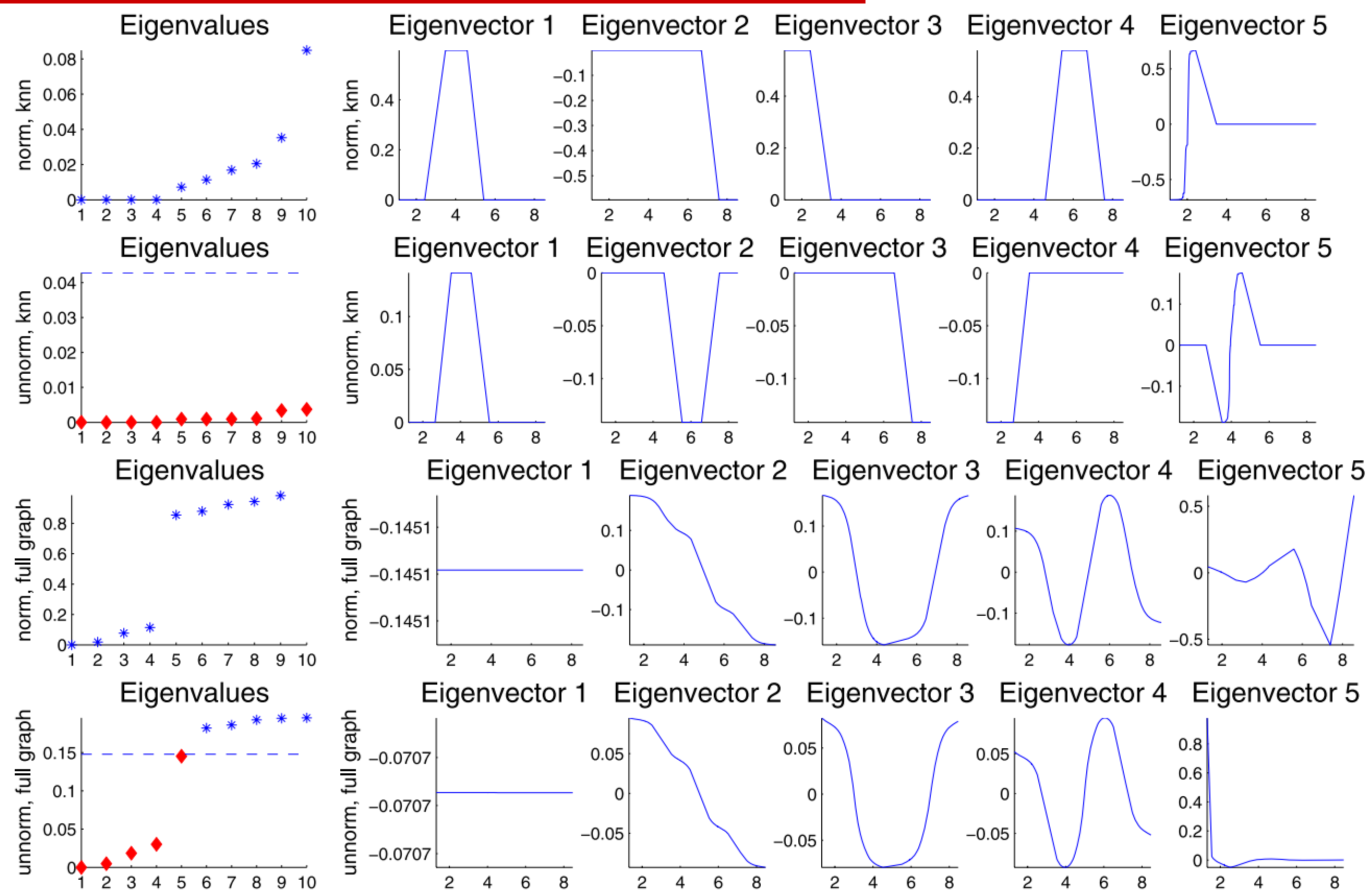
- 输入：n个点$\{p_i\}$，簇的数目k
  - 计算n×n的相似度矩阵W和度矩阵D；
  - 计算正则拉普拉斯矩阵$L_{rw}=D^{-1}(D-W)$；
  - 计算L_{rw}的前k个特征向量$u_1,u_2,\ldots,u_k$；
  - 将k个列向量$u_1,u_2,\ldots,u_k$组成矩阵U，$U\in R^{n\times k}$；
  - 对于i=1,2,…,n,令$y_i\in R^k$是U的第i行的向量；
  - 使用k-means算法将点$(y_i)_{i=1,2,\ldots,n}$聚类成簇$C_1,C_2,\ldots C_k$；
  - 输出簇$A_1,A_2,\ldots A_k$，其中，$Ai=\{j|y_j\in Ci\}$

谱聚类算法：对称拉普拉斯矩阵

- 输入：n个点$\{p_i\}$，簇的数目k
  - 计算n×n的相似度矩阵W和度矩阵D；
  - 计算正则拉普拉斯矩阵$L_{sym}=D^{-1/2}(D-W)\,D^{-1/2}$；
  - 计算L_{sym}的前k个特征向量$u_1,u_2,\dots,u_k$；
  - 将k个列向量$u_1,u_2,\dots,u_k$组成矩阵U，$U\in R^{n\times k}$；
  - 对于$i=1,2,\dots,n$，令$y_i\in R^k$是U的第i行的向量；
  - 对于$i=1,2,\dots,n$，将$y_i\in R^k$依次单位化，使得$|y_i|=1$；
  - 使用k-means算法将点$(y_i)_{i=1,2,\dots,n}$聚类成簇$C_1,C_2,\dots C_k$；
  - 输出簇$A_1,A_2,\dots A_k$，其中，$A_i=\{j|y_j\in C_i\}$

一个实例


Histogram of the sample

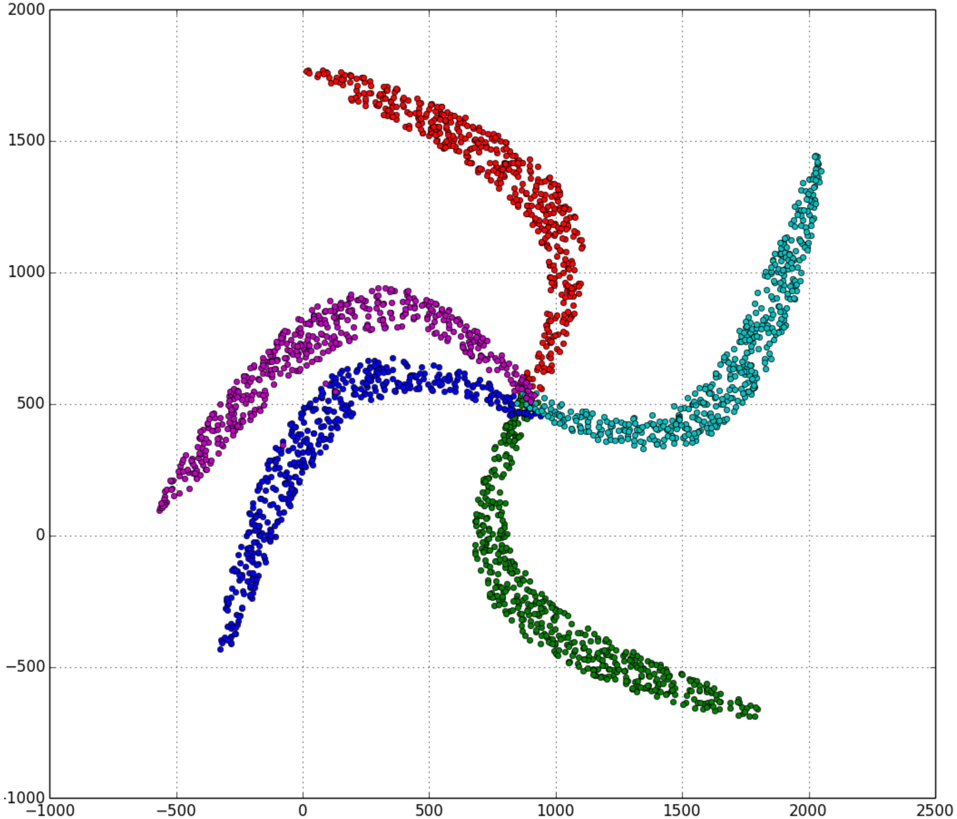

Eigenvalues
Eigenvector 1
Eigenvector 2
Eigenvector 3
Eigenvector 4
Eigenvector 5
norm, knn
unnorm, knn
norm, full graph
unnorm, full graph

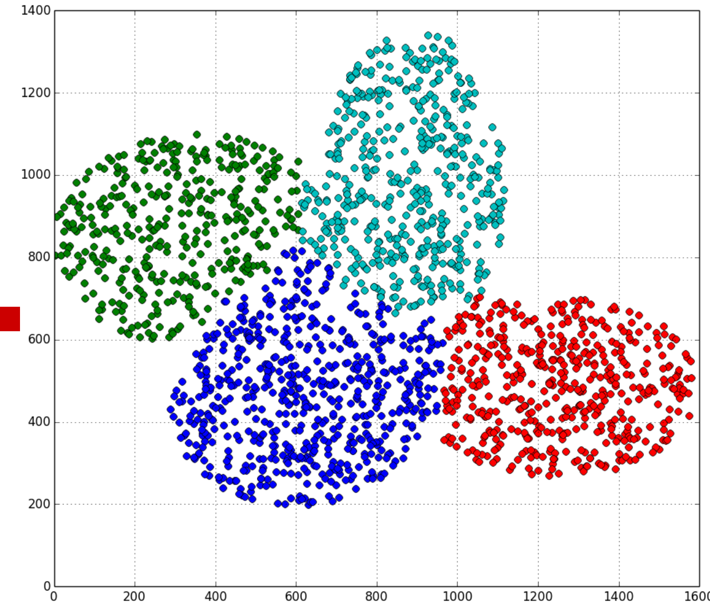
Code

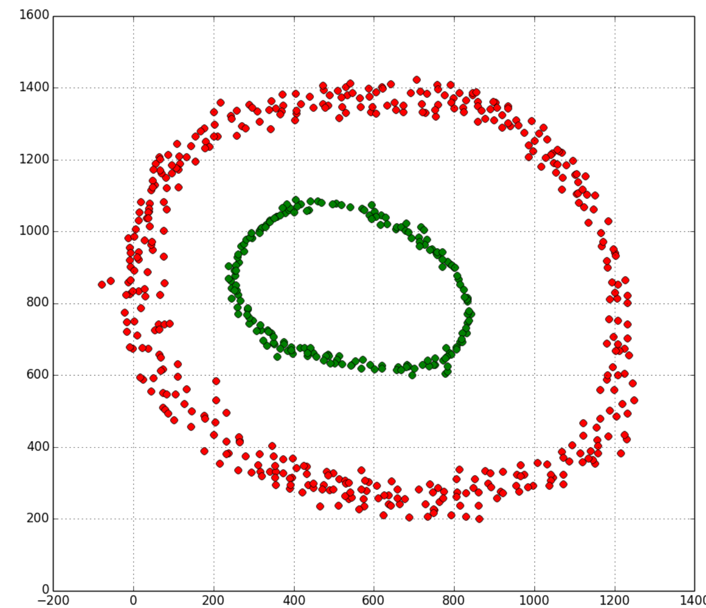
```python
def spectral_cluster(data):
    lm = laplace_matrix(data)
    eg_values, eg_vectors = linalg.eig(lm)
    idx = eg_values.argsort()
    eg_vectors = eg_vectors[:, idx]

    m = len(data)
    eg_data = [[] for x in range(m)]
    for i in range(m):
        eg_data[i] = [0 for x in range(k)]
        for j in range(k):
            eg_data[i][j] = eg_vectors[i][j]
    return k_means(eg_data)
```

```python
def laplace_matrix(data):
    m = len(data)
    w = [[] for x in range(m)]
    for i in range(m):
        w[i] = [0 for x in range(m)]
    nearest = [0 for x in range(neighbor)]

    for i in range(m):
        zero_list(nearest)
        for j in range(i+1, m):
            w[i][j] = similar(data, i, j)
            if not is_neighbor(w[i][j], nearest):
                w[i][j] = 0
            w[j][i] = w[i][j]   #对称
        w[i][i] = 0
    for i in range(m):
        s = 0
        for j in range(m):
            s += w[i][j]
        if s == 0:
            print "矩阵第", i, "行全为0"
            continue
        for j in range(m):
            w[i][j] /= s
            w[i][j] = -w[i][j]
        w[i][i] += 1      #单位阵主对角线为1
    return w
```

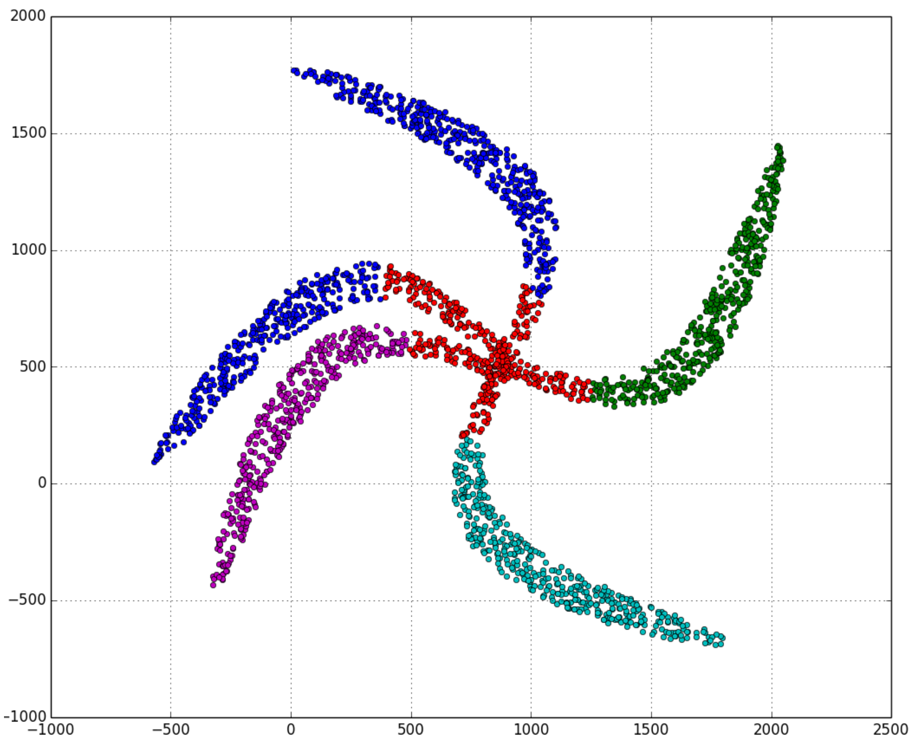
聚类效果

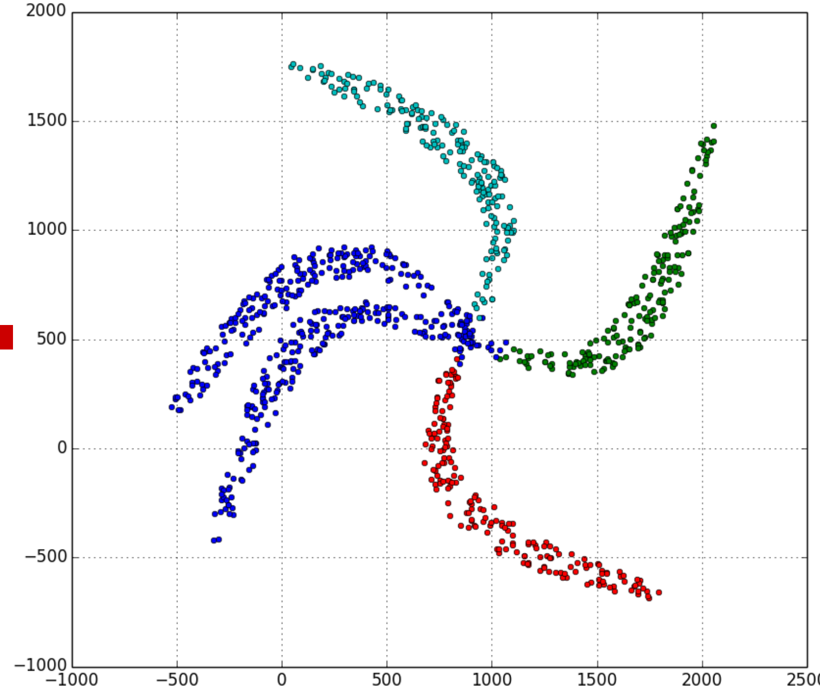

2000
1500
1000
500
0
−500
-1000
−1000
−500
0
500
1000
1500
2000
2500

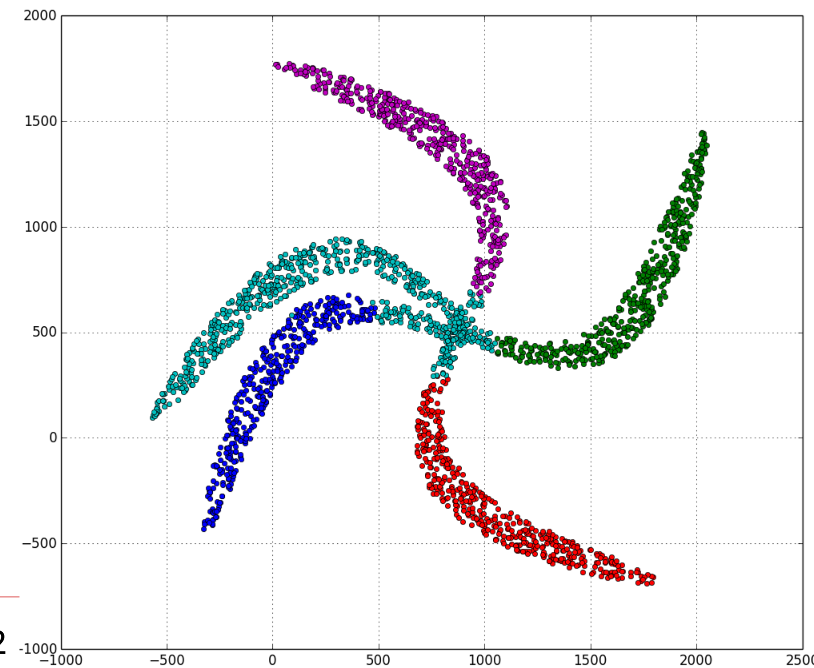

1400
1200
1000
800
600
400
200
0
0
200
400
600
800
1000
1200
1400
1600


1600
1400
1200
1000
800
600
400
200
0
−200
0
200
400
600
800
1000
1200
1400

聚类失败的情况


2000
1500
1000
500
0
−500
−1000
−1000
−500
0
500
1000
1500
2000
2500


2000
1500
1000
500
0
−500
−1000
−1000
−500
0
500
1000
1500
2000
2500


2000
1500
1000
500
0
−500
-1000
−1000
−500
0
500
1000
1500
2000
2500

进一步思考

- 谱聚类中的K如何确定？$k^* = \arg\max_k |\lambda_{k+1} - \lambda_k|$
- 最后一步K-Means的作用是什么？
  - 目标函数是关于子图划分指示向量的函数，该向量的值根据子图划分确定，是离散的。该问题是NP的，转换成求连续实数域上的解，最后用K-Means算法离散化。
- 未正则/对称/随机游走拉普拉斯矩阵，首选哪个？
  - 随机游走拉普拉斯矩阵
- 谱聚类可以用切割图/随机游走/扰动论等解释。

随机游走和拉普拉斯矩阵的关系

- 图论中的随机游走是一个随机过程，它从一个顶点跳转到另外一个顶点。谱聚类即找到图的一个划分，使得随机游走在相同的簇中停留而几乎不会游走到其他簇。
- 转移矩阵：从顶点vi跳转到顶点vj的概率正比于边的权值wij

$$p_{ij} = w_{ij} / d_i \qquad P = D^{-1} W$$

标签传递算法

- 对于部分样本的标记给定，而大多数样本的标记未知的情形，是半监督学习问题。
- 标签传递算法(Label Propagation Algorithm,LPA)，将标记样本的标记通过一定的概率传递给未标记样本，直到最终收敛。

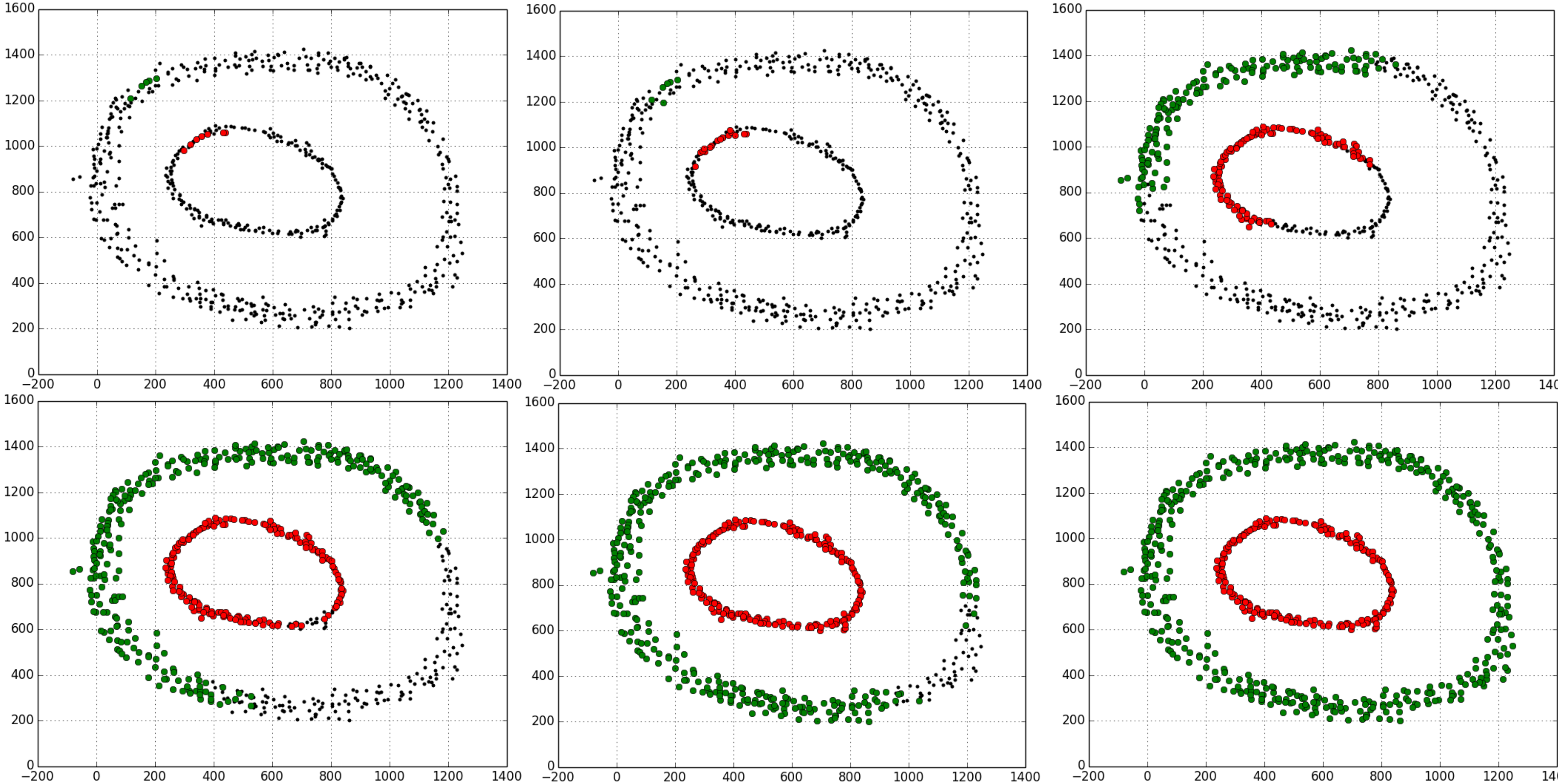
Code

```python
def label_propagation(data, a):
    p = transition_matrix(data)
    m = len(data)
    n = len(data[0])
    for times in range(100):
        for i in range(a, m):
            j = calc_label(p, i)
            label = data[j][n-1]
            if label > 0:
                data[i][n-1] = label

def calc_label(p, i):
    n = len(p[i])
    k = random.random()    # k∈[0,1)
    r = n-1
    for j in range(n):
        if p[i][j] > k:
            r = j
            break
    return r
```

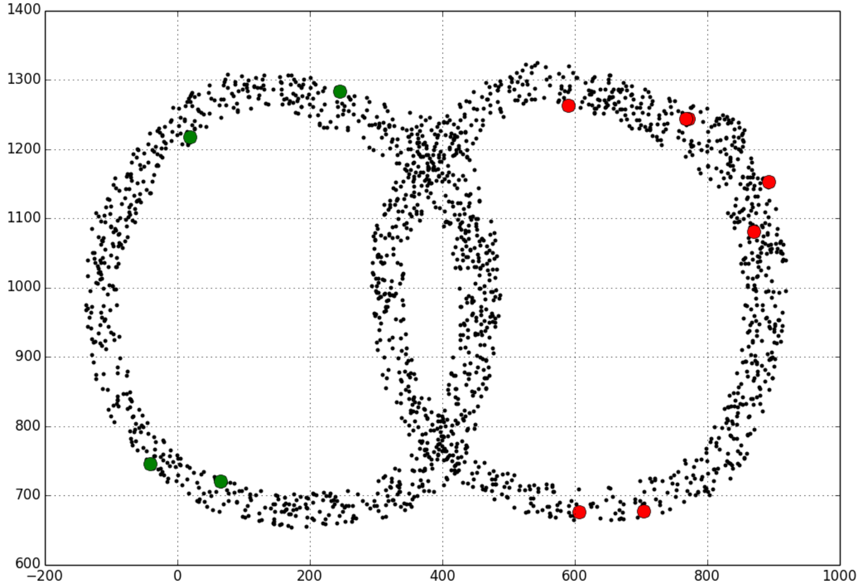
```python
def transition_matrix(data):
    m = len(data)
    p = [[] for x in range(m)]
    for i in range(m):
        p[i] = [0 for x in range(m)]
    nearest = [0 for x in range(neighbor)]

    for i in range(m):
        zero_list(nearest)
        for j in range(i+1, m):
            p[i][j] = similar(data, i, j)
            if not is_neighbor(p[i][j], nearest):
                p[i][j] = 0
            p[j][i] = p[i][j]    #对称
        p[i][i] = 0
    for i in range(m):
        s = 0
        for j in range(m):
            s += p[i][j]
        if s == 0:
            print "矩阵第", i, "行全为0"
            continue
        for j in range(m):
            p[i][j] /= s
            if j != 0:
                p[i][j] += p[i][j-1]
    return p
```

标签传递过程

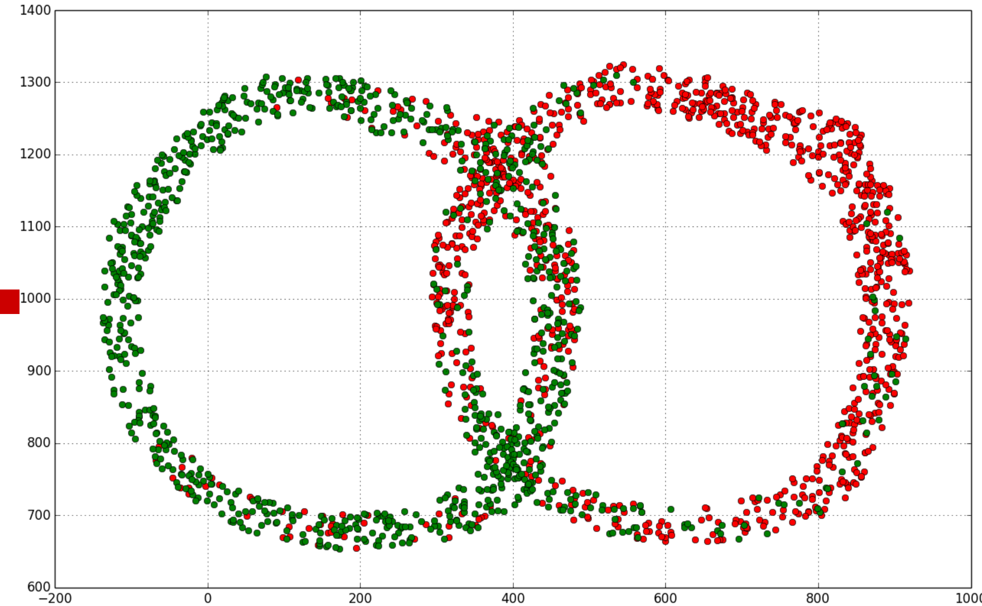
初始	1	10
20	30	40



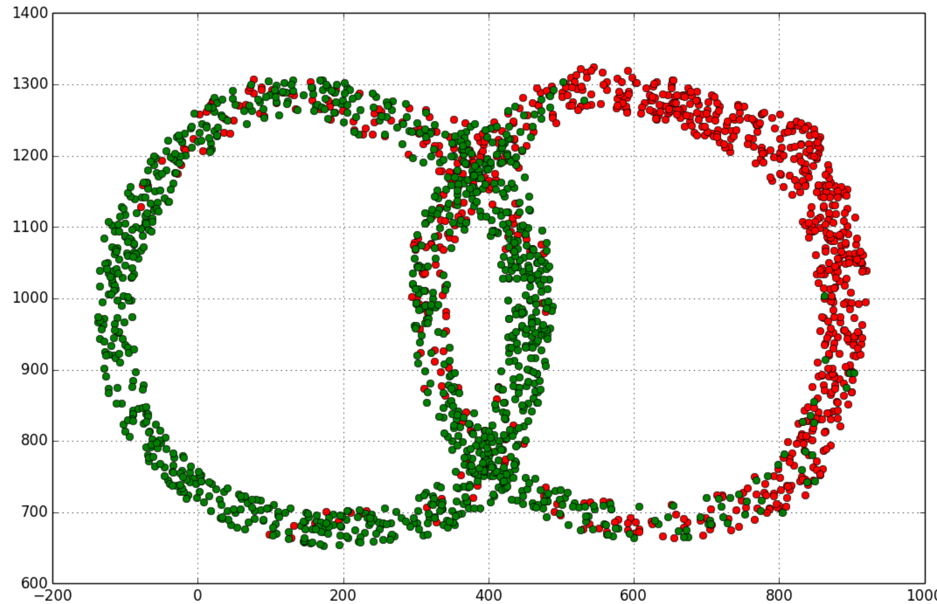
带宽/邻域影响



1400
1300
1200
1100
1000
900
800
700
600
−200
0
200
400
600
800
1000



1400
1300
1200
1100
1000
900
800
700
600
−200
0
200
400
600
800
1000



1400
1300
1200
1100
1000
900
800
700
600
−200
0
200
400
600
800
1000

聚类的衡量指标

- 均一性
  - Homogeneity
  - 一个簇只包含一个类别的样本，则满足均一性

$$h=\begin{cases}1 & if\ \ H(C)=0\\ 1-\dfrac{H(C|K)}{H(C)} & otherwise\end{cases}$$

- 完整性
  - Completeness
  - 同类别样本被归类到相同簇中，则满足完整性

$$c=\begin{cases}1 & if\ \ H(K)=0\\ 1-\dfrac{H(K|C)}{H(K)} & otherwise\end{cases}$$

- V-measure

$$v_\beta=\frac{(1+\beta)\cdot h\cdot c}{\beta\cdot h+c}$$

  - 均一性和完整性的加权平均

ARI

C	Y_1	Y_2	$\cdots$	Y_s	sum
X_1	n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
sum	b_1	b_2	$\cdots$	b_s	N

- 数据集S共有N个元素，两个聚类结果分别是：

$$X=\{X_1,X_2,\cdots X_r\}\quad Y=\{Y_1,Y_2,\cdots Y_s\}$$

- X和Y的元素个数为：$a=\{a_1,a_2,\cdots a_r\}\quad b=\{b_1,b_2,\cdots b_s\}$
- 记：$n_{ij}=|X_i\cap Y_i|$
- 则：

$$ARI=\frac{Index-EIndex}{MaxIndex-EIndex}=\frac{\sum_{i,j}C_{n_{ij}}^2-\left[\left(\sum_i C_{a_i}^2\right)\cdot\left(\sum_j C_{b_j}^2\right)\right]\Big/C_n^2}{\frac{1}{2}\left[\left(\sum_i C_{a_i}^2\right)+\left(\sum_j C_{b_j}^2\right)\right]-\left[\left(\sum_i C_{a_i}^2\right)\cdot\left(\sum_j C_{b_j}^2\right)\right]\Big/C_n^2}$$

AMI

- 使用与ARI相同的记号，根据信息熵，得：
- 互信息/正则化互信息：

$$MI(X,Y)=\sum_{i=1}^{r}\sum_{j=1}^{s}P(i,j)\log\frac{P(i,j)}{P(i)P(j)} \qquad NMI(X,Y)=\frac{MI(X,Y)}{\sqrt{H(X)H(Y)}}$$

- X服从超几何分布，求互信息期望：

$$E[MI]=\sum_{x}P(X=x)MI(X,Y)=\sum_{x=\max(1,a_i+b_i-N)}^{\min(a_i,b_i)}\left[MI\cdot\frac{a_i!b_j!(N-a_i)!(N-b_j)!}{N!x!(a_i-x)!(b_j-x)!(N-a_i-b_j+x)!}\right]$$

- 借鉴ARI，有：$AMI(X,Y)=\dfrac{MI(X,Y)-E[MI(X,Y)]}{\max\{H(X),H(Y)\}-E[MI(X,Y)]}$

轮廓系数Silhouette

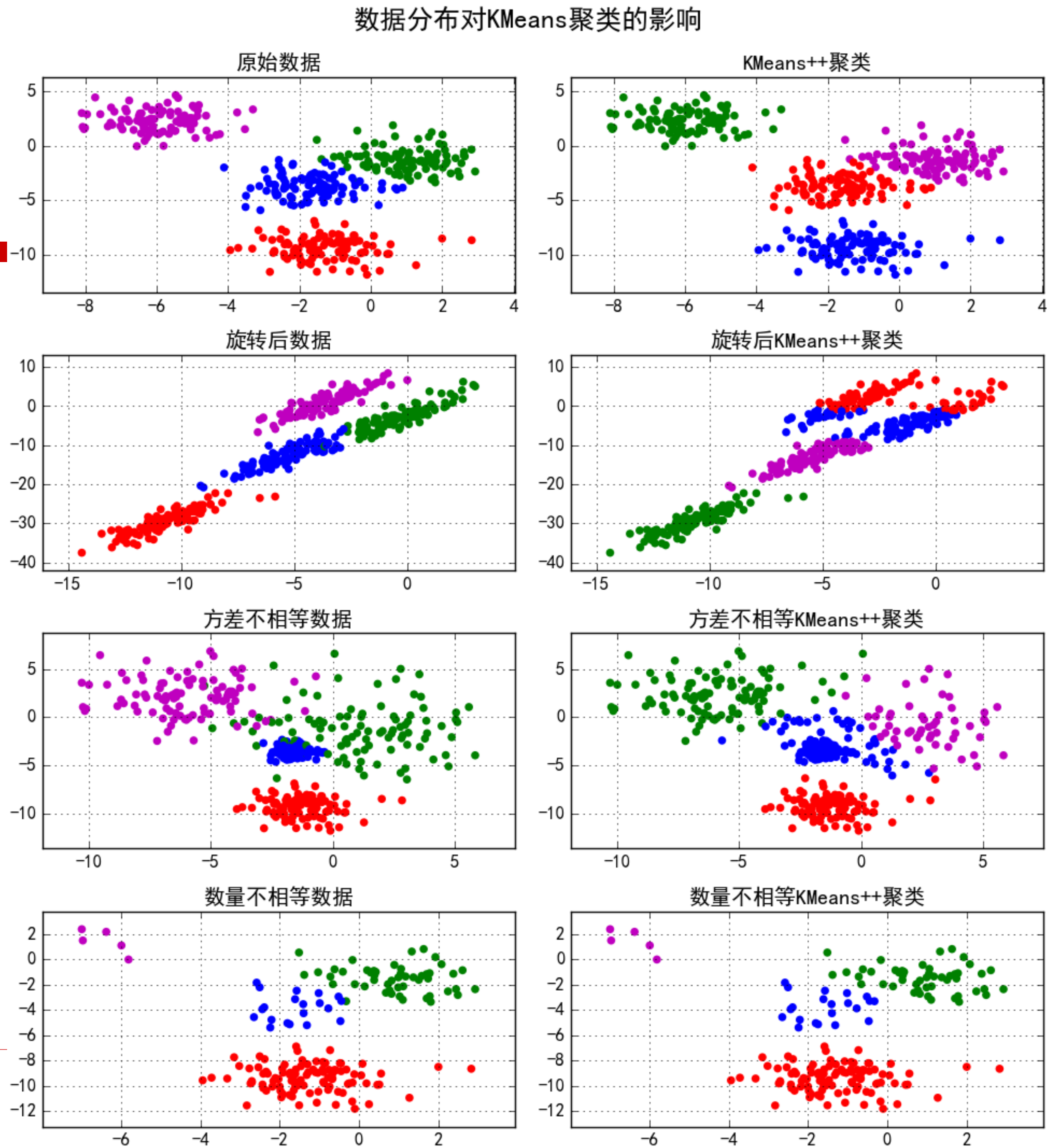
- 样本i到同簇其他样本的平均距离a_i，样本i到最近的其他簇的所有样本的平均距离b_i

$$s(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$$

$$s(i)=\begin{cases}1-\frac{a(i)}{b(i)}, & a(i)<b(i)\\ 0, & a(i)=b(i)\\ \frac{b(i)}{a(i)}-1, & a(i)>b(i)\end{cases}$$

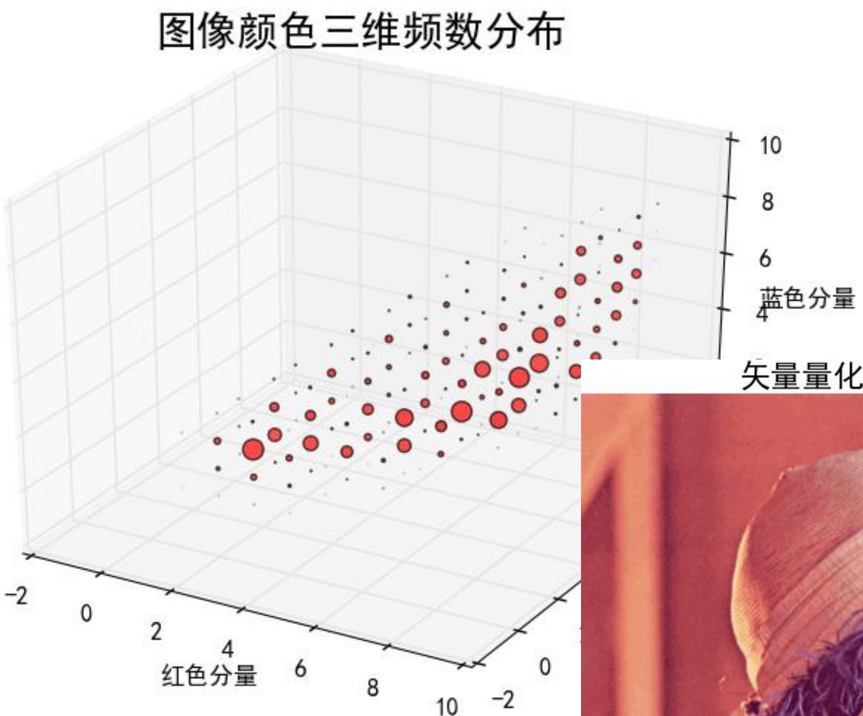
- 所有样本轮廓系数的平均值称为聚类结果的轮廓系数。

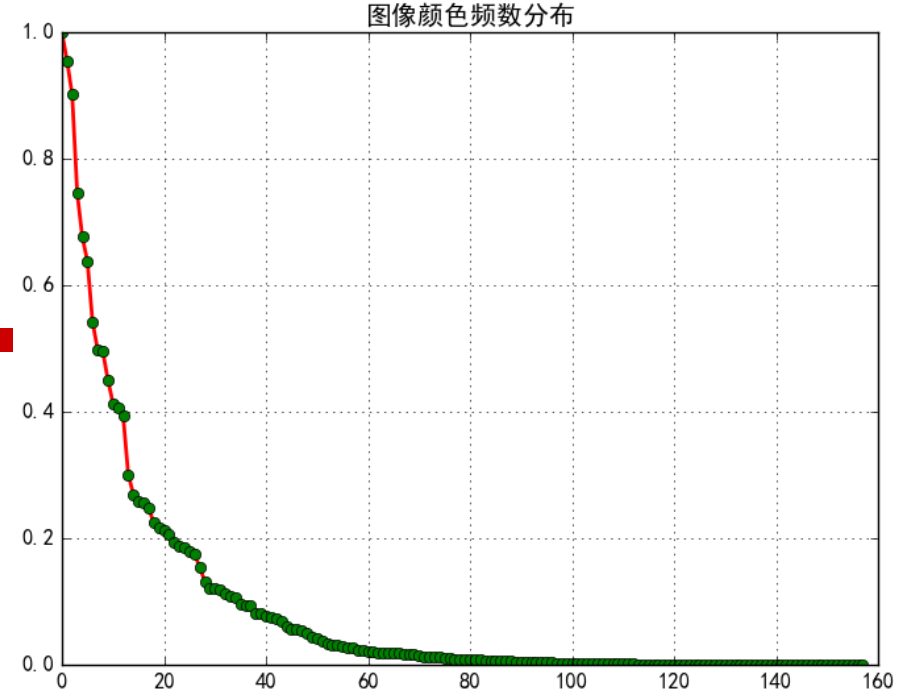
K-Means

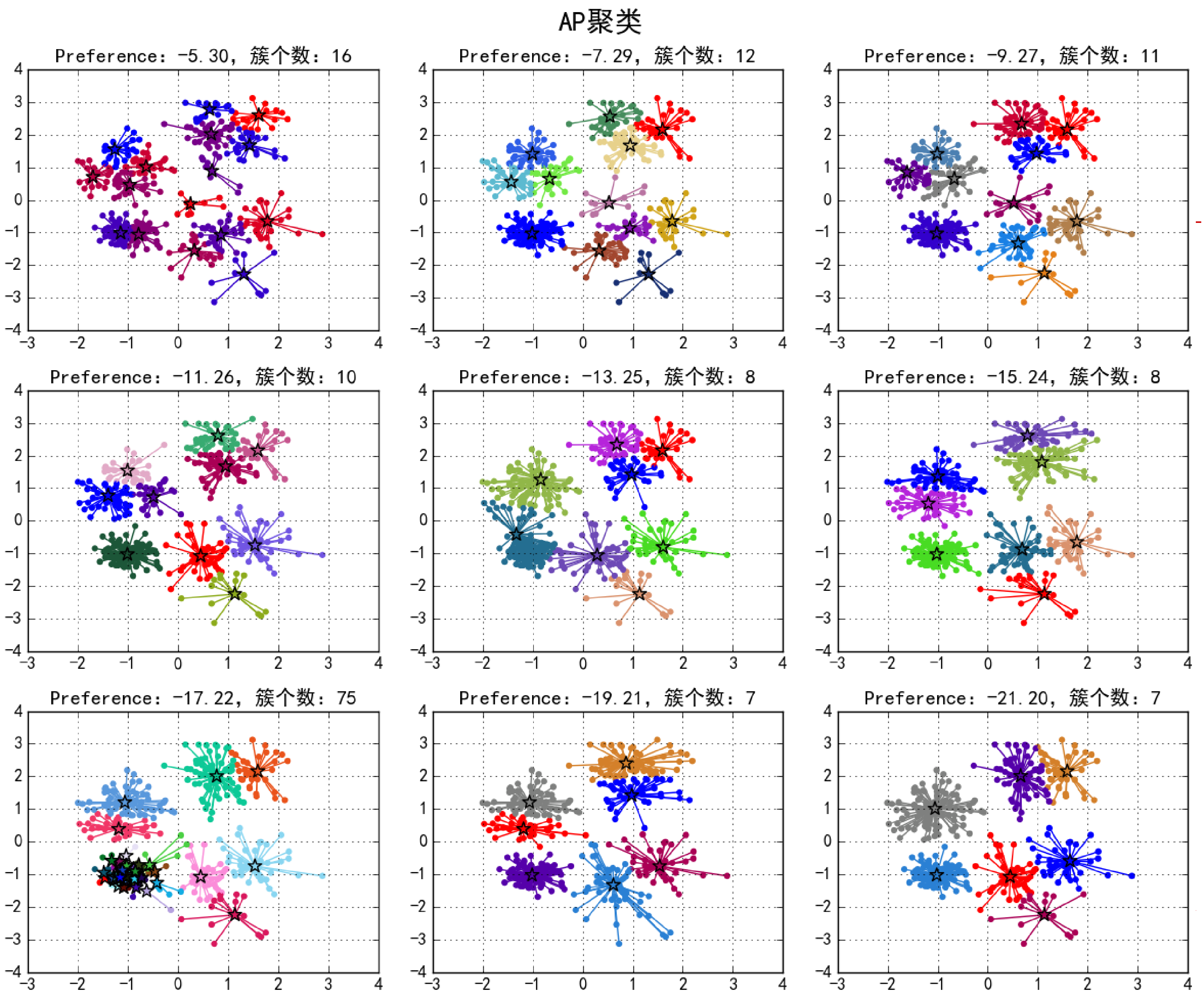


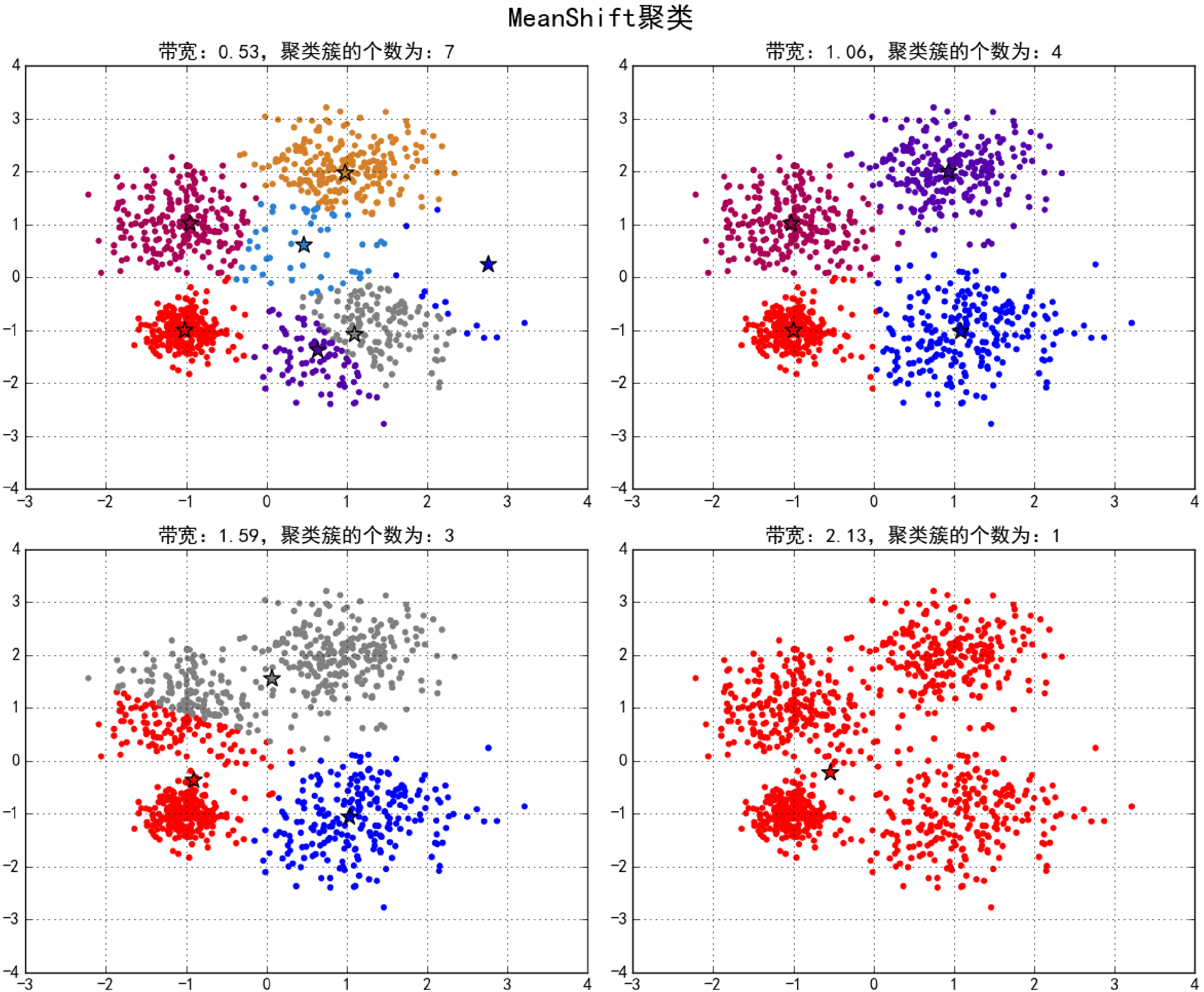
数据分布对KMeans聚类的影响
原始数据
KMeans++聚类
旋转后数据
旋转后KMeans++聚类
方差不相等数据
方差不相等KMeans++聚类
数量不相等数据
数量不相等KMeans++聚类

Vector Quantization

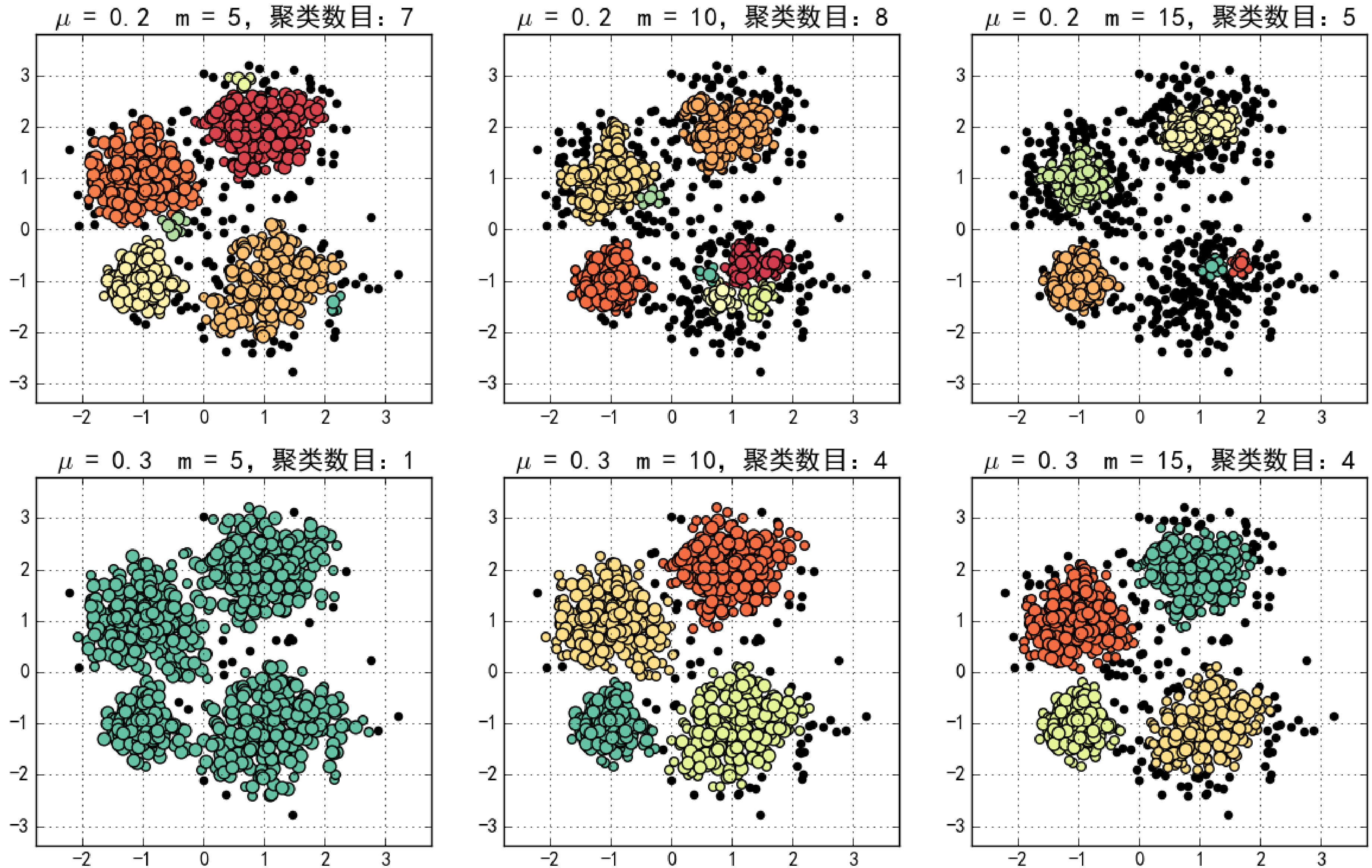





AP聚类
Preference：-5.30，簇个数：16
Preference：-7.29，簇个数：12
Preference：-9.27，簇个数：11
Preference：-11.26，簇个数：10
Preference：-13.25，簇个数：8
Preference：-15.24，簇个数：8
Preference：-17.22，簇个数：75
Preference：-19.21，簇个数：7
Preference：-21.20，簇个数：7

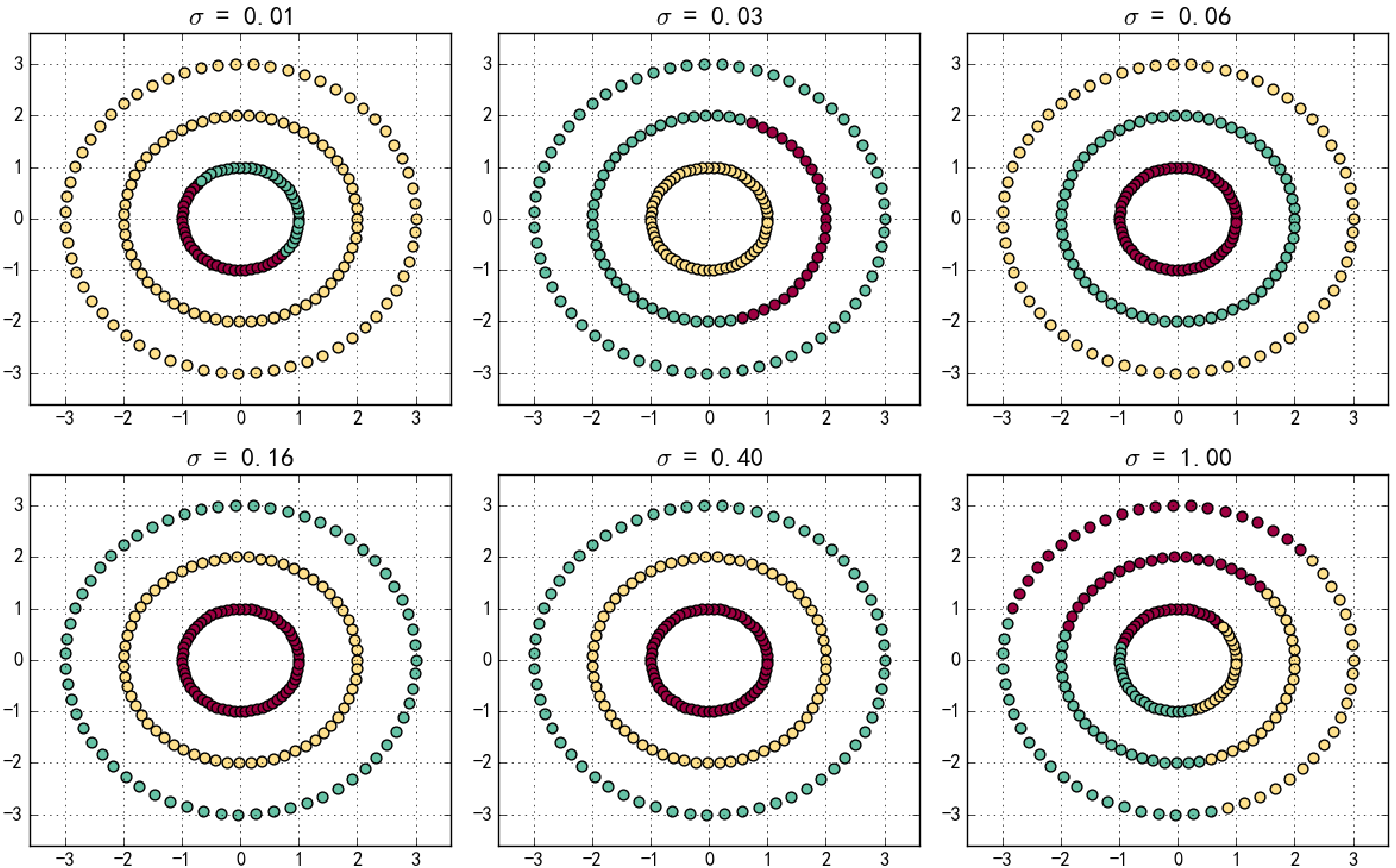

MeanShift聚类
带宽：0.53，聚类簇的个数为：7
带宽：1.06，聚类簇的个数为：4
带宽：1.59，聚类簇的个数为：3
带宽：2.13，聚类簇的个数为：1

DBSCAN聚类



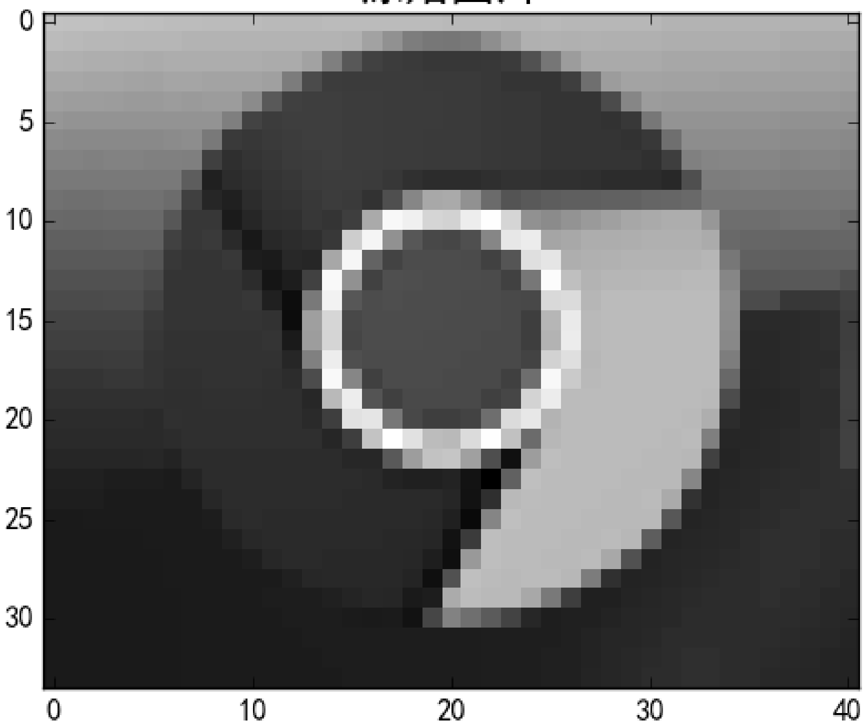
μ = 0.2 m = 5，聚类数目：7
μ = 0.2 m = 10，聚类数目：8
μ = 0.2 m = 15，聚类数目：5
μ = 0.3 m = 5，聚类数目：1
μ = 0.3 m = 10，聚类数目：4
μ = 0.3 m = 15，聚类数目：4

谱聚类

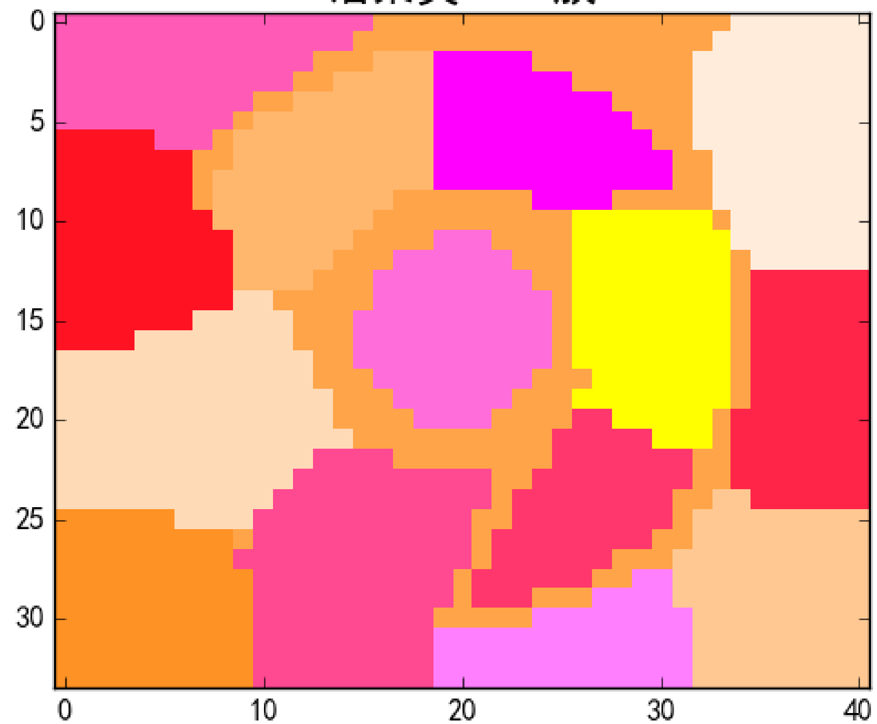


谱聚类与图像切割

原始图片



谱聚类：15簇



Demo

参考文献

- Andrew Rosenberg, Julia Hirschberg, *V-Measure: A conditional entropy-based external cluster evaluation measure*, 2007.
- W. M. Rand. *Objective criteria for the evaluation of clustering methods*. Journal of the American Statistical Association. 1971
- Nguyen Xuan Vinh, Julien Epps, James Bailey, *Information theoretic measures for clusterings comparison*, ICML 2009
- Peter J. Rousseeuw, *Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis*. Computational and Applied Mathematics 20: 53–65, 1987
- https://en.wikipedia.org/wiki/Rand_index
- https://en.wikipedia.org/wiki/Adjusted_mutual_information

我们在这里

- http://wenda.ChinaHadoop.cn
  - 视频/课程/社区
- 微博
  - @ChinaHadoop
  - @邹博_机器学习
- 微信公众号
  - 小象
  - 大数据分析挖掘

感谢大家！

恳请大家批评指正！

附：谱聚类的图切割推导

拉普拉斯矩阵的性质

- 定理：令G是权值非负的无向图，拉普拉斯矩阵L的特征值0的重数k等于图G的连通分量数。记G的连通分量为$A_1, A_2, \ldots A_k$，则特征值0的特征向量由下列指示向量确定。

$$\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$$

正则拉普拉斯矩阵的性质

- (λ，u)是L_{rw}的特征值和特征向量，当且仅当(λ，$D^{1/2}$u)是L_{sym}的特征值和特征向量；
- (0，$\mathbf{1}$)是L_{rw}的特征值和特征向量，(0，$D^{1/2}\mathbf{1}$)是L_{sym}的特征值和特征向量；
- L_{sym}和L_{rw}是半正定的，有n个非负实特征值

$$f' L_{\mathrm{sym}} f = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}}\right)^2 \quad , \; f \in \mathbb{R}^n$$

正则拉普拉斯矩阵的性质

- 定理：令G是权值非负的无向图，正则拉普拉斯矩阵L_{sym}和L_{rw}的特征值0的重数k等于图G的连通分量数。记G的连通分量为$A_1,A_2 \ldots A_k$，则特征值0的特征向量由下列指示向量确定。

$$L_{rw} \qquad \mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$$

$$L_{sym} \quad D^{1/2}\,\mathbb{1}_{A_1}, \ldots, D^{1/2}\,\mathbb{1}_{A_k}$$

切割图

- 聚类问题的本质：
- 对于定值k和图G，选择一组划分：$A_1, A_2, \ldots, A_k$，最小化下面的式子：

$$\mathrm{cut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, \bar{A}_i)$$

修正目标函数

- 上述的目标函数存在问题：在很多情况下，minCut的解，将图分成了一个点和其余的n-1个点。为了避免这个问题，目标函数应该显示的要求$A_1, A_2, \ldots, Ak$足够大。

$$\mathrm{RatioCut}(A_1, \ldots, A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i, \bar{A}_i)}{|A_i|} = \sum_{i=1}^{k}\frac{\mathrm{cut}(A_i, \bar{A}_i)}{|A_i|}$$

$$\mathrm{Ncut}(A_1, \ldots, A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i, \bar{A}_i)}{\mathrm{vol}(A_i)} = \sum_{i=1}^{k}\frac{\mathrm{cut}(A_i, \bar{A}_i)}{\mathrm{vol}(A_i)}$$

分析分母对目标函数的影响

$$\mathrm{RatioCut}(A_1,\ldots,A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i,\bar{A}_i)}{|A_i|} = \sum_{i=1}^{k}\frac{\mathrm{cut}(A_i,\bar{A}_i)}{|A_i|}$$

$$\mathrm{Ncut}(A_1,\ldots,A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i,\bar{A}_i)}{\mathrm{vol}(A_i)} = \sum_{i=1}^{k}\frac{\mathrm{cut}(A_i,\bar{A}_i)}{\mathrm{vol}(A_i)}$$

- 上述目标函数以Ai的点数或者权值作为被除数，使得函数$\sum_{i=1}^{k}\frac{1}{|A_i|}$的最小值在$|A_i|$相等的时候达到；函数$\sum_{i=1}^{k}\frac{1}{vol(A_i)}$的最小值在$vol(A_i)$相等的时候达到。从而，目标函数能够试图得到“平衡”的簇。
  - 带等式约束的极值问题，约束条件：

$$\sum_{i=1}^{k}|A_i| = n \qquad \sum_{i=1}^{k}vol(A_i) = sum(D)$$

当k=2时的RatioCut

- 目标函数：$\min_{A \subset V} \mathrm{RatioCut}(A, \bar{A})$
- 定义向量f=(f1,f2,...fn)^T，
  - 它其实是分割子图的指示向量

$$f_i = \begin{cases} \sqrt{|\bar{A}|/|A|} & \text{if } v_i \in A \\ -\sqrt{|A|/|\bar{A}|} & \text{if } v_i \in \bar{A} \end{cases}$$

RatioCut与拉普拉斯矩阵的关系

$$
\begin{aligned}
f'Lf &= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \\
&= \frac{1}{2}\sum_{i\in A, j\in \bar{A}} w_{ij}\left(\sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}}\right)^2 \\
&\quad + \frac{1}{2}\sum_{i\in \bar{A}, j\in A} w_{ij}\left(-\sqrt{\frac{|\bar{A}|}{|A|}} - \sqrt{\frac{|A|}{|\bar{A}|}}\right)^2 \\
&= \mathrm{cut}(A,\bar{A})\left(\frac{|\bar{A}|}{|A|} + \frac{|A|}{|\bar{A}|} + 2\right) \\
&= \mathrm{cut}(A,\bar{A})\left(\frac{|A|+|\bar{A}|}{|A|} + \frac{|A|+|\bar{A}|}{|\bar{A}|}\right) \\
&= |V|\cdot \mathrm{RatioCut}(A,\bar{A}).
\end{aligned}
$$

目标函数

$$\min_{A\subset V} f'Lf, \quad f_i = \begin{cases} \sqrt{|\bar{A}|/|A|} & \text{if } v_i \in A \\ -\sqrt{|A|/|\bar{A}|} & \text{if } v_i \in \bar{A} \end{cases}$$

- 该目标函数的自变量部分，是不同的子图划分；从而得到不同的f指示向量。优化的目标，是使得该目标函数取值最小。
  - 由于f只能取2个值，离散化的定义域导致问题是NP的。

考察f的性质

$$\sum_{i=1}^{n} f_i = \sum_{i\in A}\sqrt{\frac{|\bar{A}|}{|A|}} - \sum_{i\in \bar{A}}\sqrt{\frac{|A|}{|\bar{A}|}} = |A|\sqrt{\frac{|\bar{A}|}{|A|}} - |\bar{A}|\sqrt{\frac{|A|}{|\bar{A}|}} = 0$$

- 上式可以看做f和全1向量的点乘，从而 $f \perp \mathbf{1}$

$$\| f \|^2 = \sum_{i=1}^{n} f_i^2 = |A|\frac{|\bar{A}|}{|A|} + |\bar{A}|\frac{|A|}{|\bar{A}|} = |\bar{A}| + |A| = n$$

- 上式说明，f的模是定值。

目标函数约束条件的放松relaxation

- 将f的严格定义用f的性质代替，向量f各个分量的取值从离散若干个值延拓到整个实数域，从而得到：

$$\min_{f\in\mathbb{R}^n} f'Lf, \quad \text{s.t. } f \perp \mathbb{1},\ \| f \| = \sqrt{n}$$

基于子图划分的结论

$$\min_{f \in \mathbb{R}^n} f'Lf, \quad \text{s.t. } f \perp \mathbb{1}, \ \|f\| = \sqrt{n}$$

- 若f为$\mathbb{1}$，可以使得f'Lf最小，显然这对应着全连接图——不切割任何边。
- 由于L是对称阵，因此，L非0的特征值对应的特征向量一定与$\mathbb{1}$正交。因为要求最小，因此，求次小的特征值即可；次小特征值对应的特征向量即对应着2划分的策略。
  - 聚类标准：次小特征向量各个分量的正负。
- k划分即是求前k小的特征向量，使用某简单的聚类方法即可：如K-means聚类。

附：将子图划分从2扩展到k

The relaxation of the RatioCut minimization problem in the case of a general value k follows a similar principle as the one above. Given a partition of V into k sets $A_1, \ldots, A_k$, we define k indicator vectors $h_j = (h_{1,j}, \ldots, h_{n,j})'$ by

$$h_{i,j} = \begin{cases} 1/\sqrt{|A_j|} & \text{if } v_i \in A_j, \\ 0 & \text{otherwise} \end{cases}$$

$$(i = 1, \ldots, n;\ j = 1, \ldots, k)$$

附：考察指示向量组成的矩阵

Then we set the matrix $H \in \mathbb{R}^{n\times k}$ as the matrix containing those k indicator vectors as columns. Observe that the columns in H are orthonormal to each other, that is $H'H = I$. Similar we can see that

$$h_i' L h_i = \frac{\mathrm{cut}(A_i, \bar{A}_i)}{|A_i|}$$

$$h_i' L h_i = (H' L H)_{ii}$$

附：目标函数

$$\mathrm{RatioCut}(A_1,\ldots,A_k)=\sum_{i=1}^{k} h_i' L h_i = \sum_{i=1}^{k} (H' L H)_{ii} = \mathrm{Tr}(H' L H)$$

$$\min_{H \in \mathbb{R}^{n \times k}} \mathrm{Tr}(H' L H),\ \text{s.t. } H' H = I$$

- 根据Rayleigh-Ritz定理，L的前k个特征向量即为$h_1, h_2, \ldots, h_k$。