AI300学院

机器学习之无监督学习

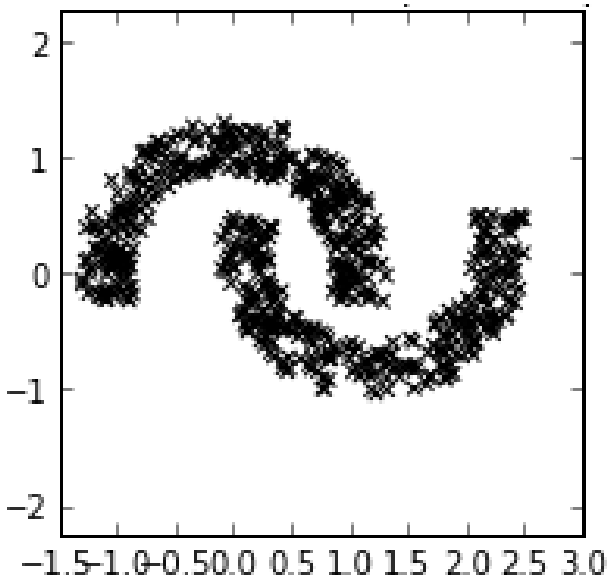
聚类算法-Spectral

倪冰冰
上海交通大学

AI300学院

K-Means & GMM

- K-means和GMM算法的缺陷
  1. 当数据属于非凸条件时效果较差 (*Spectral Clustering*)

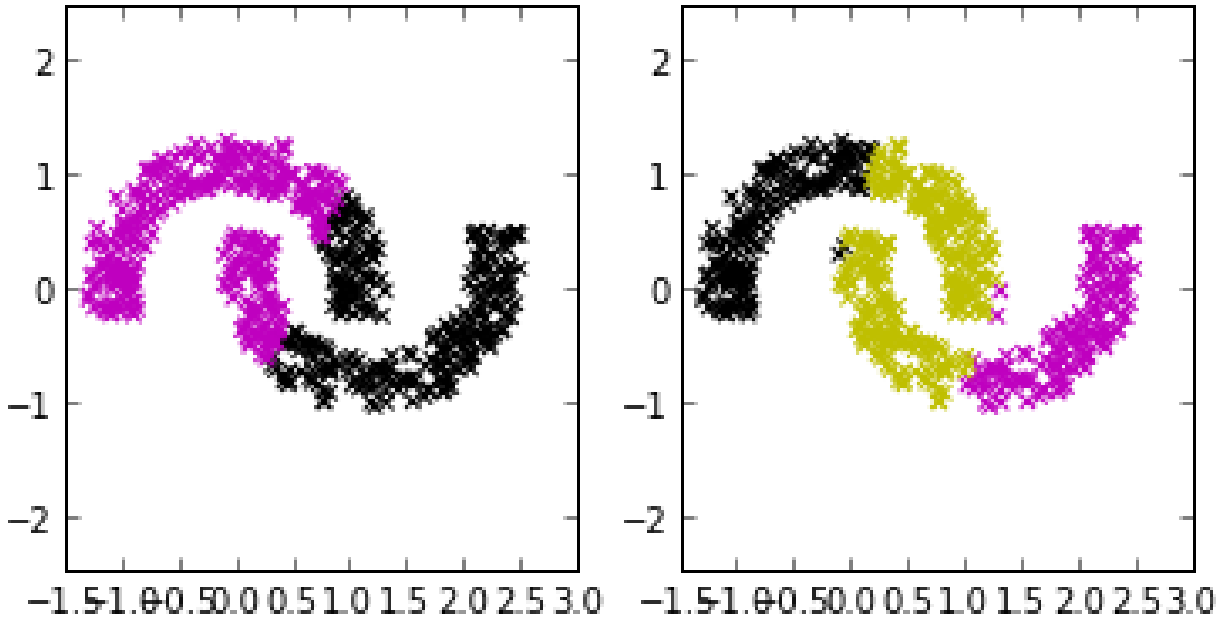


非凸数据分布

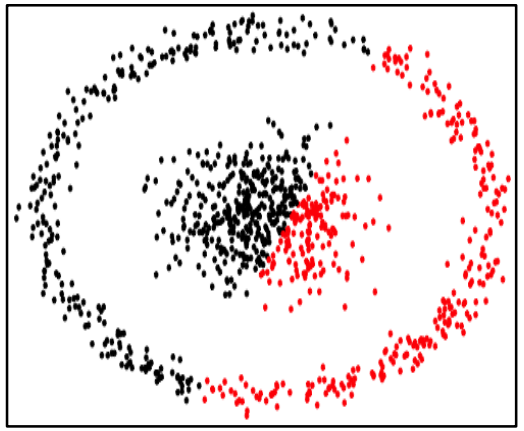
AI300学院

K-Means & GMM

- K-means和GMM算法的缺陷
  1. 当数据属于非凸条件时效果较差 (*Spectral Clustering*)



K-means 结果

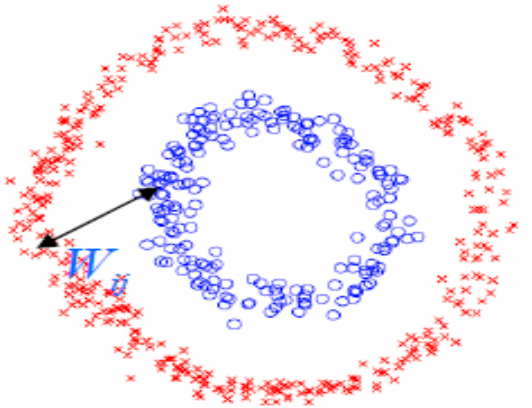


GMM 结果

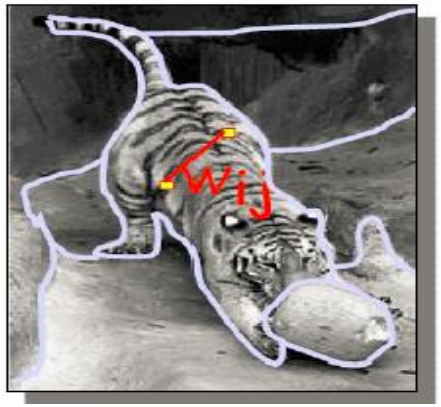
Graph-based Clustering

- 谱聚类=给每个图节点打标签

  - Data Grouping

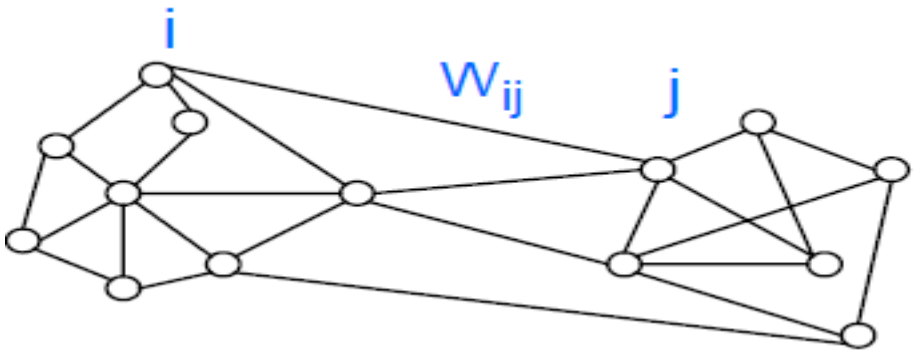


  - Image sigmentation



Clustering = labeling each node!

$$W_{ij} = f(d(x_i, x_j))$$



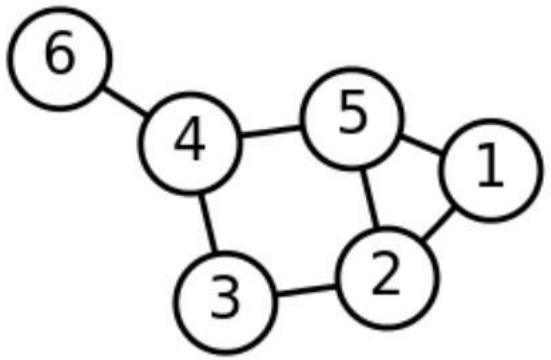
$$G = \{V, E\}$$

- 每个节点 V 代表一个数据样本 (例如pixel)
- 每条边 W_{ij} 代表两个样本间近似程度

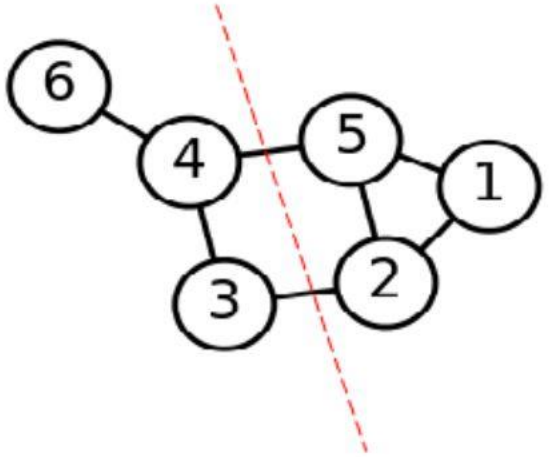
Graph-based Clustering

- 谱聚类的基本流程：

1.构图。将采样点数据构造程一张网图，表示为G(V,E)，其中V表示图中的点，E表示点与点之间的边。



2.切图。将第一步构造出来的网图按照一定的准则，切分成不同的子图，这些子图就是我们得到的聚类结果



Graph-based Clustering

- 谱聚类构图。

1.首先我们引入相似度函数（similarity function），这一函数的目的是用于计算两个样本点之间的举例，一般采用欧式距离或者高斯距离：

欧式距离：$$s_{i,j} = \|x_i - x_j\|^2$$

高斯距离：$$w_{ij} = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}$$

n个数据样本
W: n×n矩阵

据此我们可以构筑相似矩阵：$$W = [w]_{ij}$$

Graph-based Clustering

- 谱聚类构图。

2.得到相似矩阵S后一般有三种方法构图方法，或者说构建邻接矩阵W的方法：

（1）ε-neighborhood：

$$W_{i,j} = \begin{cases} 0, & if \quad s_{i,j} > \varepsilon \\ \varepsilon, & if \quad s_{i,j} \leq \varepsilon \end{cases}$$

（2）fully connected：

$$W_{i,j} = S_{i,j} = [s]_{i,j}$$

Graph-based Clustering

- 谱聚类构图。

2.得到相似矩阵S后一般有三种方法构图方法，或者说构建邻接矩阵W的方法：

（3）k-nearest neighborhood，又分为两种：

$$W_{i,j} = W_{j,i} = \begin{cases} 0, & if \quad x_i \notin KNN(X_j) \quad or \quad x_j \notin KNN(X_i) \\ e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}, & if \quad x_i \in KNN(X_j) \quad and \quad x_j \in KNN(X_i) \end{cases}$$

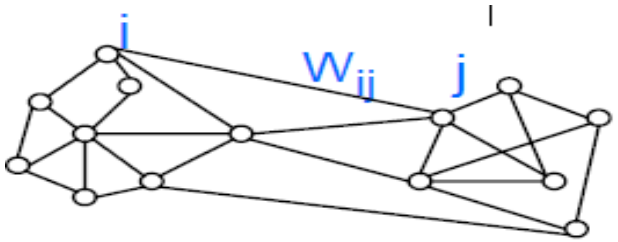
$$W_{i,j} = W_{j,i} = \begin{cases} 0, & if \quad x_i \notin KNN(X_j) \quad and \quad x_j \notin KNN(X_i) \\ e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}, & if \quad x_i \in KNN(X_j) \quad or \quad x_j \in KNN(X_i) \end{cases}$$

Graph-based Clustering

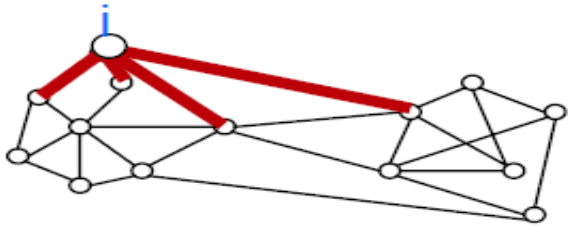
- 重要定义

边 (ij) : sample pairs with $W_{ij} > 0$

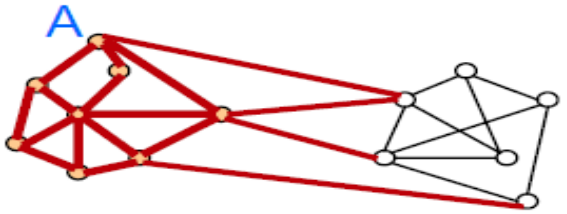
Similarity matrix: $W = [W_{ij}]$



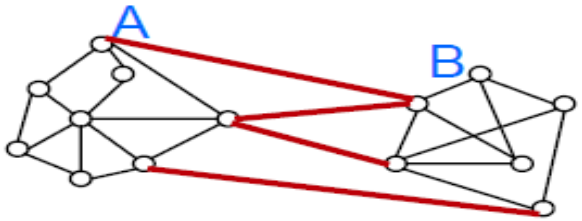
Degree of the node i: $d_i = \Sigma_{j \in V} W_{ij}$, 目的是测量某个节点的联通程度



Degree of the subgraph A: $A_i = \sum_{i \in A} d_i$, 目的是测量某个子图的联通程度



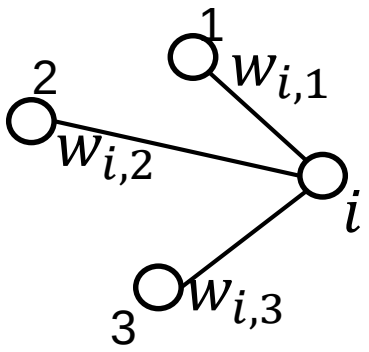
图割Cut of a graph:$Cut(A, B) = \sum_{i \in A} \sum_{j \in B} W_{ij}$, 目的是测量两个子图间的联通程度



Graph-based Clustering

- 谱聚类构图。
  3.计算阶矩（degree matrix）D:

$$D_{i,j} = \begin{cases} 0, & if \quad i \neq j \\ \sum_j w_{i,j}, & if \quad i = j \end{cases}$$



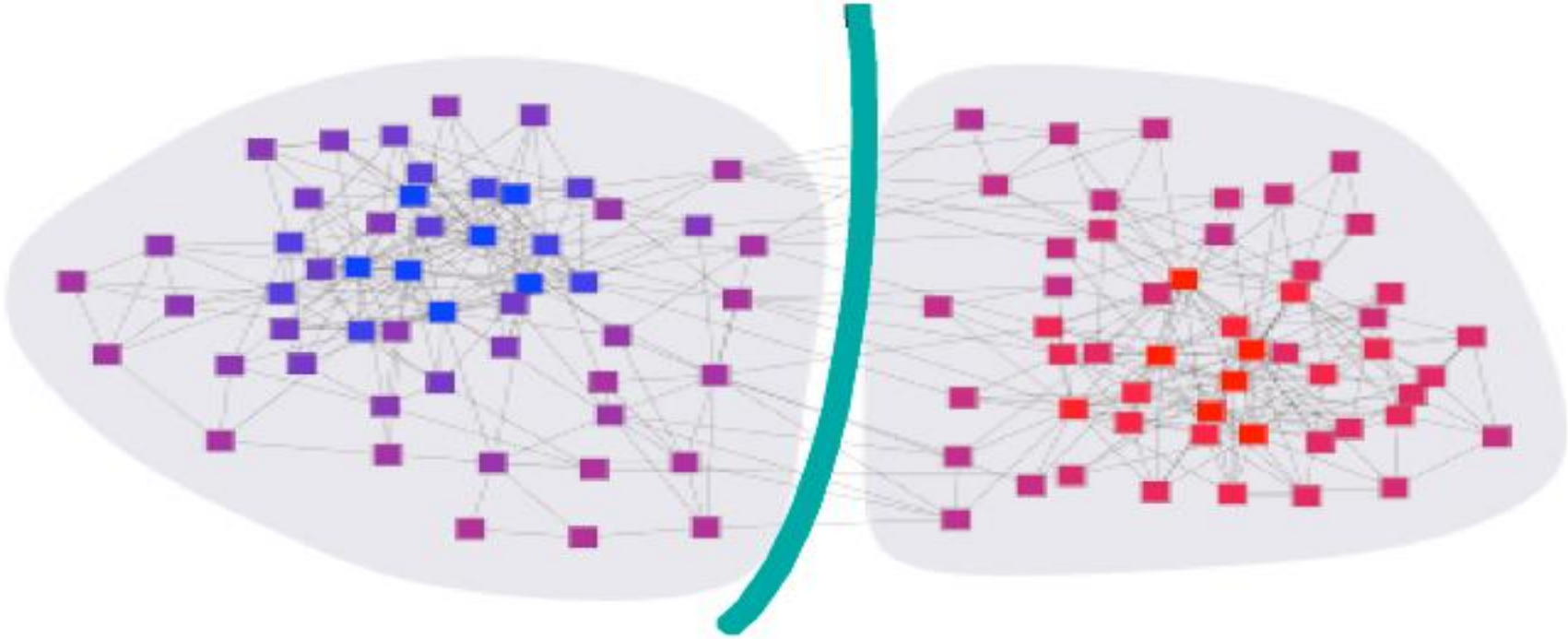
4.计算拉普拉斯矩阵：

$$L: \ L = D - W$$

$$\begin{pmatrix} d_{1,1} & & & & \\ & d_{2,2} & & & \\ & & d_{3,3} & & \\ & & & \ddots & \\ & & & & d_{n,n} \end{pmatrix}$$

Graph-based Clustering

- Graph partition = clustering 聚类=图割?



Is clustering equivalent to minimize $Cut(A, B)$ *?等价于最小化图割?*

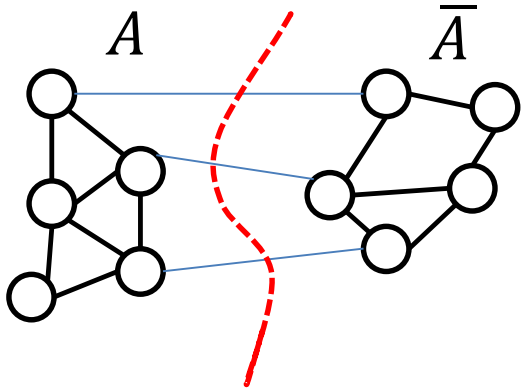
Graph-based Clustering

- 谱聚类图割

首先补充背景知识，假设V为所有样本点的集合，$\{A_1, A_2 \cdots A_k\}$表示V的子集集合，其中$A_1 \cup A_2 \cup \cdots \cup A_k = V$且$A_1 \cap A_2 \cap \cdots \cap A_k = \emptyset$则子集与子集之间连边的权重和为：

$$cut(A_1, A_2, \cdots, A_k) = \frac{1}{2}\sum_{i}^{k} W(A_i, \bar{A}_i)$$

其中$\bar{A}_i$为A_i的补集，$W(A_i, \bar{A}_i)$为A_i与其他子集的连边的和



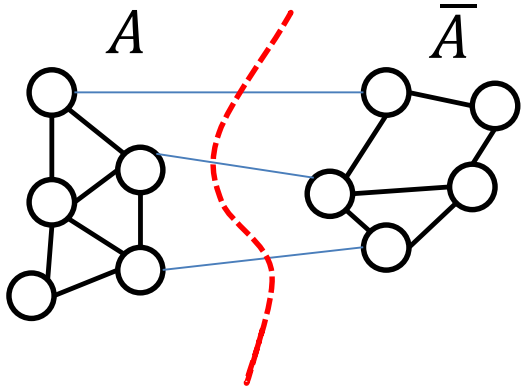
$$W(A_i, \bar{A}_i) = \sum_{m \in A_i, n \in \bar{A}_i} w_{m,n}$$

Graph-based Clustering

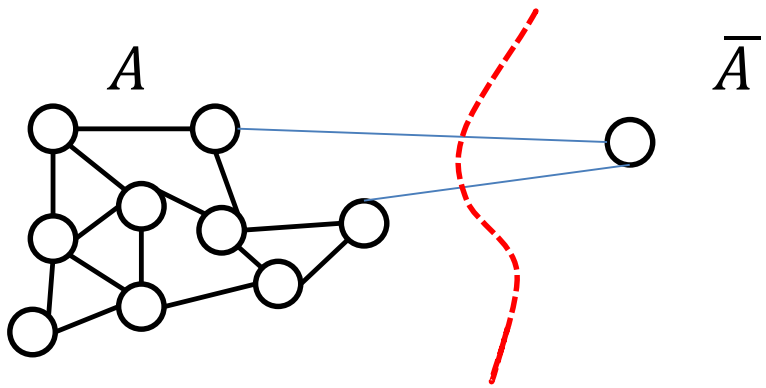
- 谱聚类切图

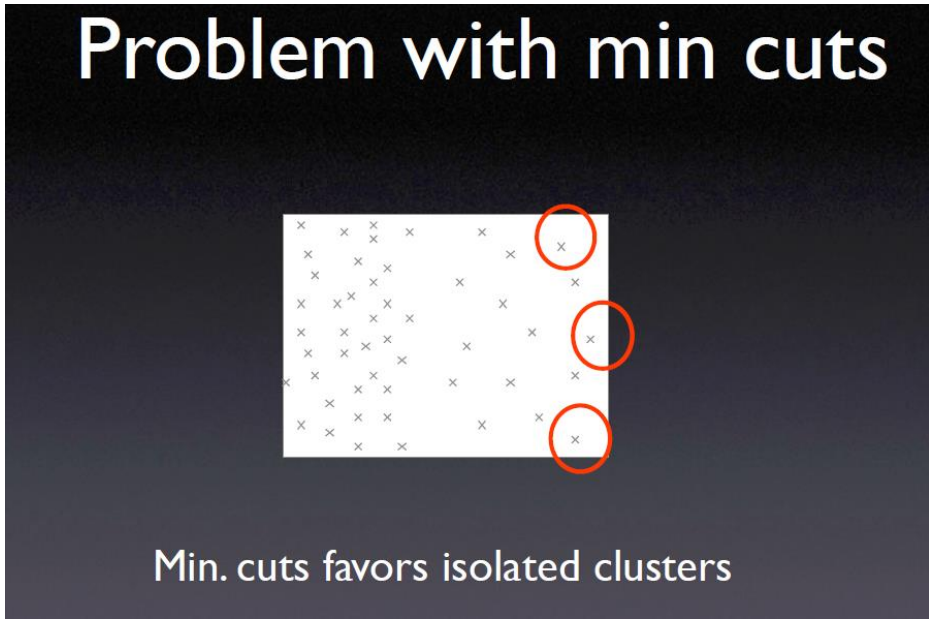
我们切图的目的是使得每个子图内的结构相似，我们可以理解为连边的权重平均都比较大，且相互连接，而每个子图间则尽量没有边相连，或者连边的权重很低，我们的目的可以做如下表述：

$$min \quad cut(A_1, A_2, \cdots, A_k)$$



Good or not?

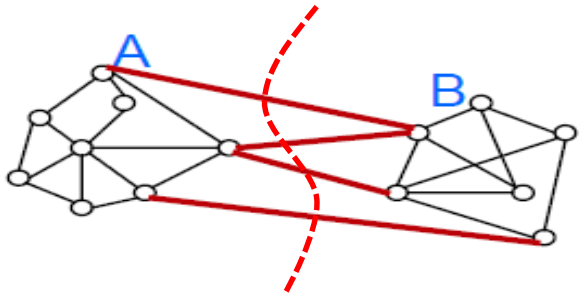




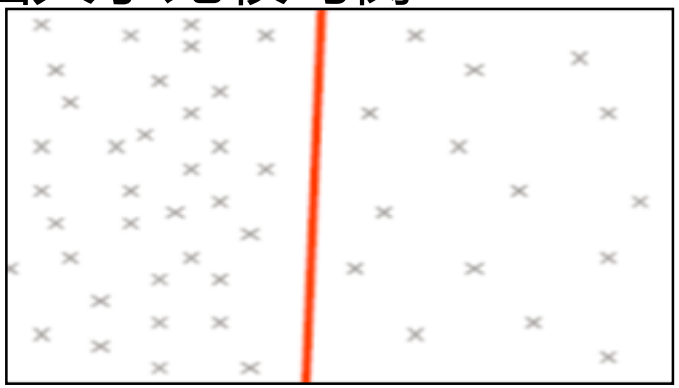
Graph-based Clustering

- 解决方案: 归一化的图割

$$\mathrm{Ncut}(A,B) = \frac{\mathrm{cut}(A,B)}{d_A} + \frac{\mathrm{cut}(A,B)}{d_B}$$



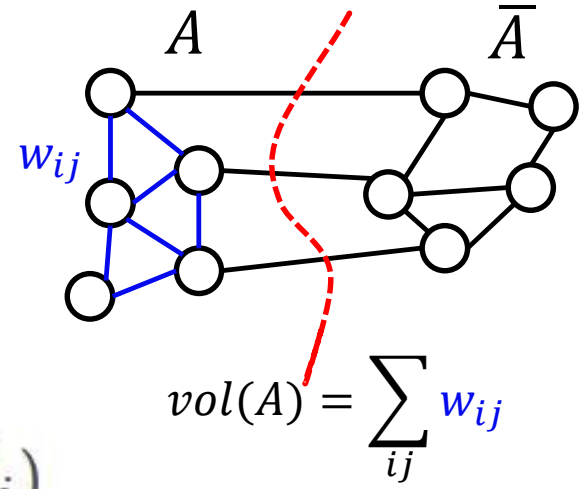
- 使得被分割后的子图大小比较均衡



但是要找到这样的分割是NP-hard的问题!

Graph-based Clustering

需要归一化的图割!



定义Volume: $vol(A) = \sum_{ij} w_{ij}$,where $i \in A$ and $j \in A$

由此得出归一化图割: $Ncut(A_1, A_2, \cdots, A_k) = \frac{1}{2}\sum_{i}^{k} \frac{W(A_i, \bar{A}_i)}{vol(A_i)}$

$$Ratiocut(A_1, A_2, \cdots, A_k) = \frac{1}{2}\sum_{i}^{k} \frac{W(A_i, \bar{A}_i)}{|A_i|}$$

$|A_i|$为A_i中点的个数，$vol(A_i)$为A_i中所有边的权重和

Graph-based Clustering

$$A \cup B = V$$
$$A \cap B = \Phi$$

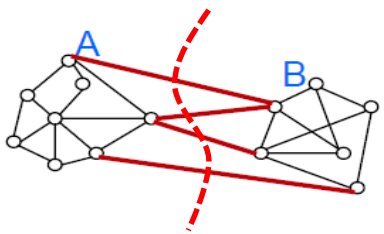
- 解析定义 *N-cut* 问题

定义 N-by-1 indicator vector $\boldsymbol{x}$, $x_i = 1$ 表示第i个节点属于子图A,
$x_i = 0$ 表示第i个节点不属于子图A, 即属于A的补图$V - A = B$

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

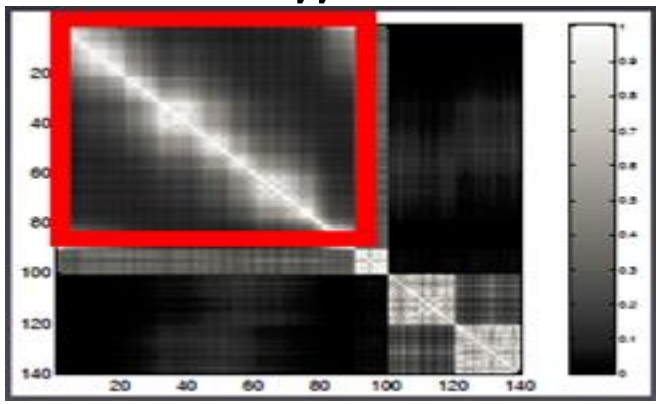
$$Cut(A, V - A) = \sum_{i \in A} \sum_{j \in V - A} W_{ij} = \sum_{i \in A} d_i - \sum_{i \in A} \sum_{j \in A} W_{ij}$$

$$\boldsymbol{x}^T D \boldsymbol{x} \qquad \boldsymbol{x}^T W \boldsymbol{x}$$



$\boldsymbol{x}^T$

W



$\boldsymbol{x}$

Graph-based Clustering

- 新的目标函数

$$\min_{x} \boldsymbol{x}^T(D-W)\boldsymbol{x} \qquad s.t.\boldsymbol{x}^T D\boldsymbol{x} \geq \epsilon$$

$$\Rightarrow \quad (D-W)\boldsymbol{x} = \lambda D\boldsymbol{x}$$

Rayleigh quotient theorem

将x所能取值的范围松弛，从{0,1}扩大至实数

- 有意思的是 $(D-W)\mathbf{1} = 0$, 第一个特征向量是$\mathbf{1}$, 对应的特征值为0

- 用其他特征向量来表示数据!

Graph-based Clustering

- 谱聚类切图

我们以Ncut为例，此处我们略过详细的推导过程，之后按照如下步骤：

（1）对拉普拉斯矩阵进行标准化操作：

$$D^{-1/2}LD^{-1/2}$$

（2）计算标准化操作后拉普拉斯矩阵最小的k_1个特征值对应的特征向量F

（3）将各自对应的特征向量F组成的矩阵按行标准化，最终组成$\mathrm{n} \times k_1$的特征矩阵P。

（4）对P中的每一行作为一个k_1维的样本，用聚类方法进行聚类，聚类维数为k_2。

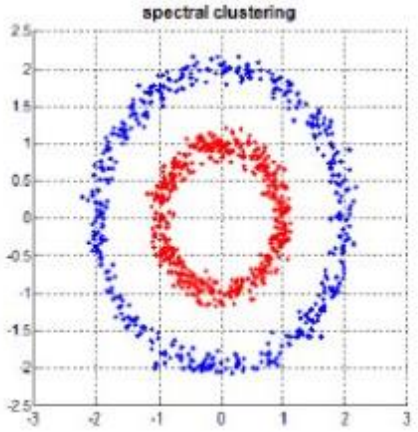
（5）最终得到分类结果。

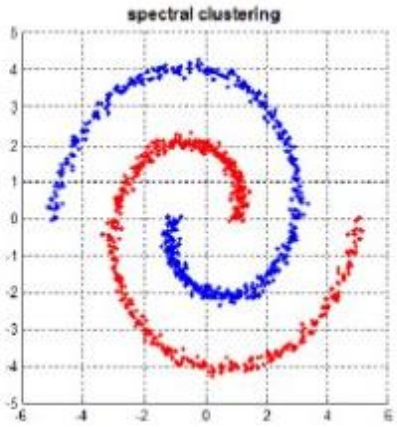
AI300学院

Graph-based Clustering

- 优势：
  1.对数据有很好的表征
  2.可以对非凸性的聚类做出很好的解决方法

- 缺点
  拓展性不好





Spectral Clustering实战

Spectral Clustering scikit-learn的python实现

【对scikit-learn中Spectral Clustering概述】

在scikit-learn的类库中，sklearn.cluster.SpectralClustering实现了基于Ncut的谱聚类，没有实现基于RatioCut的切图聚类。

对于相似矩阵的建立，也只是实现了基于K邻近法和全连接法的方式，没有基于ϵ-邻近法的相似矩阵。

参数分析：

- n_cluster：聚类分组数目
- Affinity：相似矩阵的建立方式
- 核函数参数 gamma，degree，coef0，kernel_params
- eigen_solver：在降维计算特征值特征向量的时候，使用的工具
- assign_labels：即最后的聚类方法的选择，有K-Means算法和 discretize算法两种算法可以选择

AI300学院

Spectral Clustering实战

Spectral Clustering scikit-learn的python实现

【SpectralClustering聚类】

选择最常用的高斯核来建立相似矩阵，用K-Means来做最后的聚类。

```
%%生成500个个6维的数据集，分为5个簇。
import numpy as np
from sklearn import datasets
X, y = datasets.make_blobs(n_samples=500, n_features=6,
centers=5, cluster_std=[0.4, 0.3, 0.4, 0.3, 0.4],
random_state=11)
%%默认谱聚类效果
from sklearn.cluster import SpectralClustering
y_pred = SpectralClustering().fit_predict(X)
from sklearn import metrics
print "Calinski-Harabasz Score",
metrics.calinski_harabaz_score(X, y_pred)
```

Calinski-Harabasz Score 14908.9325026

Spectral Clustering实战

Spectral Clustering scikit-learn的python实现

【SpectralClustering聚类】

由于我们使用的是高斯核，那么我们一般需要对n_clusters和gamma进行调参。

```
for index, gamma in enumerate((0.01,0.1,1,10)):
    for index, k in enumerate((3,4,5,6)):
        y_pred = SpectralClustering(n_clusters=k,
                    gamma=gamma).fit_predict(X)
        print "Calinski-Harabasz Score with gamma=",
                gamma, "n_clusters=", k,"score:",
metrics.calinski_harabaz_score(X, y_pred)
```

```
Calinski-Harabasz Score with gamma= 0.01 n_clusters= 3 score: 1979.77096092
Calinski-Harabasz Score with gamma= 0.01 n_clusters= 4 score: 3154.01841219
Calinski-Harabasz Score with gamma= 0.01 n_clusters= 5 score: 23410.63895
Calinski-Harabasz Score with gamma= 0.01 n_clusters= 6 score: 19303.7340877
Calinski-Harabasz Score with gamma= 0.1 n_clusters= 3 score: 1979.77096092
Calinski-Harabasz Score with gamma= 0.1 n_clusters= 4 score: 3154.01841219
Calinski-Harabasz Score with gamma= 0.1 n_clusters= 5 score: 23410.63895
Calinski-Harabasz Score with gamma= 0.1 n_clusters= 6 score: 19427.9618944
Calinski-Harabasz Score with gamma= 1 n_clusters= 3 score: 687.787319232
Calinski-Harabasz Score with gamma= 1 n_clusters= 4 score: 196.926294549
Calinski-Harabasz Score with gamma= 1 n_clusters= 5 score: 23410.63895
Calinski-Harabasz Score with gamma= 1 n_clusters= 6 score: 19384.9657724
Calinski-Harabasz Score with gamma= 10 n_clusters= 3 score: 43.8197355672
Calinski-Harabasz Score with gamma= 10 n_clusters= 4 score: 35.2149370067
Calinski-Harabasz Score with gamma= 10 n_clusters= 5 score: 29.1784898767
Calinski-Harabasz Score with gamma= 10 n_clusters= 6 score: 47.3799111856
```

最好的n_clusters是5，而最好的高斯核参数是1或者0.1.

Spectral Clustering实战

Spectral Clustering scikit-learn的python实现

【SpectralClustering聚类】

选择最常用的高斯核来建立相似矩阵，用K-Means来做最后的聚类。

```
y_pred = SpectralClustering(gamma=0.1).fit_predict(X)
print "Calinski-Harabasz Score",
metrics.calinski_harabaz_score(X, y_pred)
```

Calinski-Harabasz Score 14950.4939717

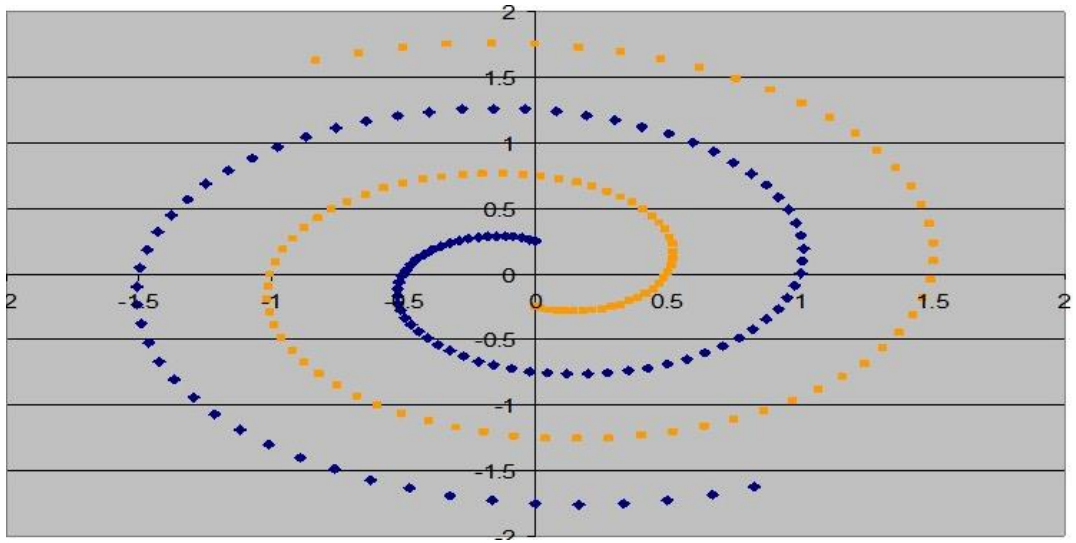
n_clusters一般还是调参选择比较好。

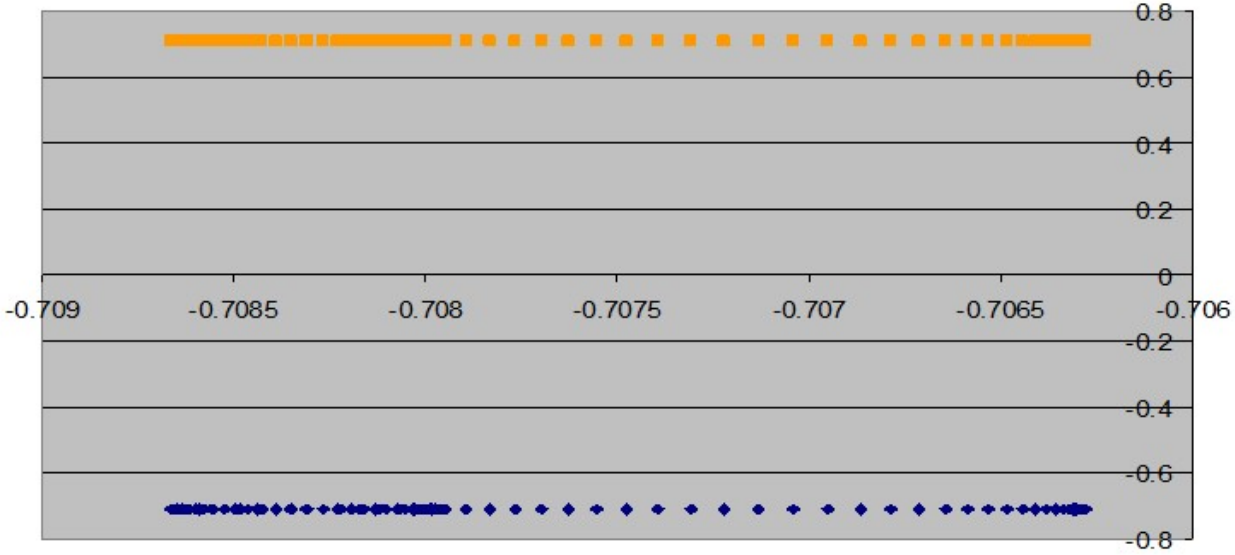
Spectral Clustering实战

Spectral Clustering scikit-learn的python实现

【Spectral Clistering特点】

谱聚类能够识别任意形状的样本空间且收敛于全局最优解





AI300学院

AI300学院