谱聚类原理简述

Arrow Luo

2016 年 10 月 2 日

1 简述

Spectral clustering(谱聚类) 是一种基于图论的聚类方法，它能够识别任意形状的样本空间并收敛于全局最优解。其基本的思想是将样本数据进行相似性计算得到相似度矩阵，然后将相似矩阵转换到 Laplacian 矩阵 (拉普拉斯矩阵)，做 Laplacian 矩阵的特征值分解，将得到的前 k 个特征向量按列排序后按行做 $k-means$ 聚类，得到最终的聚类结果。

为了便于算法理解，本文档前半部分给出了 Spectral clustering 算法的求解过程；使得不熟悉谱聚类的读者对算法有个大概了解；文档后半部分主要是解释这样求解的依据。熟悉的朋友可以跳过这个部分。在此申明，笔者水平有限，若发现任何求解错误，可以通过 infocom525@gmail.com 联系。

2 Laplacian 矩阵分类

Laplacian 矩阵有两个大类，共三种形式，如图??。

非规范化(unnormalized) Laplacian $\left\{ L := D - W \right.$

规范化(normalized) Laplacian $\left\{ \begin{array}{l} L_{sym} := D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}} \\ L_{rw} := D^{-1}(D-W) \end{array} \right.$

D 是一个 $N \times N$ 的对角阵，第i个对角线元素 $d(i) = \sum_j w(i,j)$

W是一个$N \times N$ 的相似度矩阵，元素值为相似度 w_{ij}

图 1. SC 聚类常见 Laplacian 矩阵

3 Spectral clustering 算法伪代码

为了描述尽量准确，算法先用中文描述，然后是算法的英文描述。中文实质是英文的翻译；算法中的第 4 步分别针对 L、L_{sym} 和 L_{rw} 进行特征向量的求解，第 6 步属于 L_{sym} 独有；算法中前 k 个特征值对应的特征向量指的是最小的 k 个特征值。更多细节可以参考（Luxburg, U.V. (2007) . A tutorial on spectral clustering. Statistics & Computing, 17(17), 395-416.）。

算法 1 谱聚类算法

1: **输入：**相似矩阵 $S \in \mathbb{R}^{n \times n}$ 和目标聚类个数 k.

2: 创建加权的邻接矩阵 W.

3: L: 创建非规范化 Laplacian 矩阵 L.

4: **获得前 k 个特征向量**

- L: 计算 L 的前 k 个特征向量 $u_1, \cdots, u_k$.
- L_{sym}: 计算广义特征问题 $D^{-\frac{1}{2}}LD^{-\frac{1}{2}}u = \lambda u$ 的前 k 个广义特征向量 $u_1, \cdots, u_k$.
- L_{rw}: 计算 $Lu = \lambda Du$ 的前 k 个特征向量 $u_1, \cdots, u_k$.

5: 创建矩阵 $U \in \mathbb{R}^{n \times k}$ 其中列向量为 $u_1, \cdots, u_k$.

6: L_{sym}**：将 U 按行规范为 1，具体公式 $u_{ij} = u_{ij}/(\sum_k u_{ik}^2)^{1/2}$.**

7: 对应 U 的第 i 行，生成向量 $y_i \in \mathbb{R}^k$ $(i = 1, \cdots, n)$.

8: 将点 $(y_i)_{i=1,\cdots,n} \in \mathbb{R}^k$ 应用 k-means 算法聚为类 $C_1, \cdots, C_k$.

9: **输出：**类别 $A_1, \cdots, A_k$，其中 $A_i = \{j|y_j \in C_i\}$.

Algorithm 1 Spectral clustering

1: **Input:** Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

2: Construct a similarity graph. Let W be its weighted adjacency matrix.

3: L: Compute the unnormalized Laplacian L.

4: **Compute the first k eigenvectors**

- L: Compute the first k eigenvectors $u_1, \cdots, u_k$ of L.
- L_{sym}: Compute the first k generalized eigenvectors $u_1, \cdots, u_k$ of the generalized eigenproblem $D^{-\frac{1}{2}}LD^{-\frac{1}{2}}u = \lambda u$
- L_{rw}: Compute the first k eigenvectors $u_1, \cdots, u_k$ of the eigenproblem $Lu = \lambda Du$.

5: Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \cdots, u_k$ as columns.

6: L_{sym}: **Form the matrix U by normalizing the rows to norm 1, that is set $u_{ij} = u_{ij}/(\sum_k u_{ik}^2)^{1/2}$.**

7: For $i = 1, \cdots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the $i-th$ row of U.

8: Cluster the point $(y_i)_{i=1,\cdots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters$C_1, \cdots, C_k$.

9: **Output:** Clusters $A_1, \cdots, A_k$ with $A_i = \{j|y_j \in C_i\}$.

可以看到上述伪代码仅仅给出了算法的大致流程，其中有许多细节需要做更深的处理；

1. 如果只有数值点，怎么计算相似度矩阵；
2. L、L_{sym} 和 L_{rw} 是怎么得到的，原理是什么；
3. 怎么计算 Laplacian 矩阵的特征向量；

上述的第三个关于计算特征值和特征向量的问题，是谱聚类算法的重要问题；并且特征值和特征向量的求解本身就是复杂问题，可以作为单独的文档去阐述，笔者限于篇幅限制，在此不做展开；感兴趣的朋友可以去看 matlab 中关于“eigs”的代码和文档；还有 ARPARK 的一些信息，matlab 以及现行的关于特征分解的程序 (如 spark mllib) 底层基本都是调用 ARPACK 的实现；ARPACK 底层算法是隐式重启 Arnoldi 方法 (Implicitly Restarted Arnoldi Method(IRAM)) 和

隐式重启 Lanczos 方法 (Implicitly Restarted Lanczos Method(IRLM))；对特征分解感兴趣的读者可以搜索 IRAM 和 IRLM 相关论文学习。笔者后期也会对特征分解进行剖析，相关文档可以到 (http://www.dragonyun.com/user_homepage.dhtml?uid=32896) 留意更新。

下面主要是对第一和第二个问题的一些解答。

4 相似度刻画

有很多刻画点对相似度构建相似度矩阵的方法，这里主要介绍常用的三种方法：

1. ε-邻接图

连接所有距离小于 ε 的点对，这样连接点之间的距离就控制在了 ε 的规模。除了点的连接与否之外图中没有包括更多的数据信息。因此，ε-邻接图是一个无权图。

2. k-最近邻图

如果 v_j 包含在 v_i 的 k 个最近点集中，就将 v_j 到 v_i 进行有向连接；这样的 k-最近邻图就是一个有向图，而且不一定是对称的。有两种方式可以完成有向图到无向图的转换。第一种就是直接忽略掉连接边直接的方向，然后如果 v_j 包含在 v_i 的 k 个最近点集中或者 v_i 包含在 v_j 的 k 个最近点集中，就将 v_i 和 v_j 直接连接起来；第二种做法比第一种做法更严格，如果 v_j 包含在 v_i 的 k 个最近点集中并且 v_i 包含在 v_j 的 k 个最近点集中，就将 v_i 和 v_j 直接连接起来。

3. 全连接图

用相似度来刻画点对之间的关系并将它们全部连接起来，即相似度就是边权重；一个简单的相似度刻画矩阵是 Guassian 相似度函数 $s(x_i, x_j) = exp(-\|x_i - x_j\|^2/(2\sigma^2))$，$\sigma$ 参数用来控制邻居的宽度，这个参数的作用和一个种方式中的 ε 参数的作用是类似的。

5 L、L_{sym} 和 L_{rw} 的推导简述

要解释 L、L_{sym} 和 L_{rw} 的来由，需要先了解谱聚类的作用机制。谱聚类是一种基于图论的聚类方法，首先就需要从图划分开始讲解。

图划分 常见图划分有三种方式；Mincut 划分 (Stoer & Wagner, 1997.)、RatioCut 划分 (Hagen & Kahng, 1992.) 和 NCut 划分 (Shi & Malik, 2000.)

首先对下面将会涉及到的一些标记进行说明；图划分如图??。

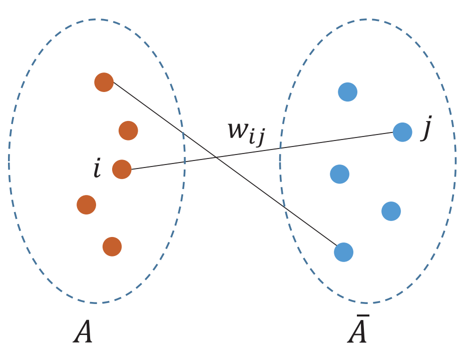

图 2. 图划分参数说明

w_{ij}：节点 i 和节点 j 间的相似度。
$W(A,B) := \sum_{i\in A, j\in B} w_{ij}$
$|A|$：点集 A 中点的个数。
$vol(A)$：点集 A 中点到其他所有点的权重之和 $\sum_{i\in A} w_{ij}$。

Mincut

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

这个公式表示将图划分为 k 个区域；遍历每个区域中的点到除该区域外的点，做权重累积；最后求总划分权重。根据聚类使类间相似度越小越好的原则，$cut(A_1,\cdots,A_k)$ 越小越好。求得其最小值对应的划分就是一个好的聚类。

但是仔细思考不难发现，上述公式求解的划分非常容易将孤立点独立出来；因为按照公式计算的结果，孤立点往往是公式计算的最小值。

RatioCut

$$RatioCut(A_i,\cdots,A_k) := \sum_{i=1}^{k} \frac{cut(A_i,\bar{A}_i)}{|A_i|}$$

为了克服 Mincut 中常常将单个点划分为单个类的窘境。一个简单的想法就是加个分母均衡一下分子，使尽可能多的点聚为一个类。RatioCut 分母使用的是类中点个数 $|A_i|$，这样单个点的分子小，分母更小；使得能够达到均衡的目的。

NCut

$$NCut(A_i,\cdots,A_k) := \sum_{i=1}^{k} \frac{cut(A_i,\bar{A}_i)}{vol(A_i)}$$

和 NCut 有类似的思想，不过分母使用的是 $vol(A_i)$ 而不是 $|A_i|$，相比较 $|A_i|$，$vol(A_i)$ 能够包括更多的信息，除了点数外还有类中点到其它点边权重。使得 $NCut(A_i,\cdots,A_k)$ 看起来更为合理，但是实际聚类结果有待进一步考察。

上述三个图划分公式除 Mincut 之外的 RatioCut 和 NCut 都是不错的划分公式，只需要求解其最小值，就能得到合理的划分。但是在实际情况中；这两个公式的求解都是 NP-hard 的。

$$\begin{cases} RatioCut(A_i,\cdots,A_k) := \sum_{i=1}^{k} \frac{cut(A_i,\bar{A}_i)}{|A_i|} \\ NCut(A_i,\cdots,A_k) := \sum_{i=1}^{k} \frac{cut(A_i,\bar{A}_i)}{vol(A_i)} \end{cases}$$

那如何求解 RatioCut 和 NCut 的最小值呢，下面是关于 RatioCut 和 NCut 的推导，为简化推导的复杂性，这里仅仅是将图划分为 A 和 B。相关细节可以参考 (Shi, J. & Malik, J. (2000). Normalized cuts and image segmentation. IEEE Transactions on Pattern)

RatioCut 推导 给出图 V 的一个划分 A 和 B。为了便于数学推导，如果点 i 属于 A，则令 $x_i = 1$；如果点 i 属于 B，则令 $x_i = -1$。则

$$\begin{aligned} RatioCut(A,B) &= \frac{cut(A,B)}{|A|} + \frac{cut(A,B)}{|B|} \\ &= \frac{\sum_{x_i>0,x_j<0} -w_{ij}x_ix_j}{\sum_{x_i>0} x_i} + \frac{\sum_{x_i<0,x_j>0} -w_{ij}x_ix_j}{\sum_{x_i<0} -x_i} \end{aligned} \tag{1}$$

下面做了一个很有技巧的处理；

1) 令 D 是一个以 $d(i)=\sum_j w(i,j)$ 为对角线的对角阵；

2) 令 W 是一个以 w_{ij} 为元素的对称阵；

3) 应用 $\frac{1+x}{2}$ 和 $\frac{1-x}{2}$ 代替 $x_i=1$ 和 $x_i=-1$；

4) $k=\frac{\sum_{x_i>0} x_i}{\sum_i |x_i|}=\frac{1^T(\frac{1+x}{2})}{1^T 1}$。

这样处理以后可以将 $RatioCut(A,B)$ 装换为矩阵的形式。

$$
\begin{aligned}
4RatioCut(A,B) &= \frac{(1+x)^T(D-W)(1+x)}{k1^T1}+\frac{(1-x)^T(D-W)(1-x)}{(1-k)1^T1} \\
&= \frac{(x^T(D-W)x+1^T(D-W)1)}{k(1-k)1^T1}+\frac{2(1-2k)1^T(D-W)x}{k(1-k)1^T1} \\
&= \frac{(x^T(D-W)x+1^T(D-W)1)+2(1-2k)1^T(D-W)x}{k(1-k)1^T1} \\
&\quad -\frac{2(x^T(D-W)x+1^T(D-W)1)}{1^T1}+\frac{2(x^T(D-W)x)}{1^T1}+\frac{2(1^T(D-W)1)}{1^T1} \\
&= \frac{(1-2k+2k^2)(x^T(D-W)x+1^T(D-W)1)+2(1-2k)1^T(D-W)x}{k(1-k)1^T1} \\
&\quad +\frac{2(x^T(D-W)x)}{1^T1} \\
&= \frac{\frac{(1-2k+2k^2)}{(1-k)^2}(x^T(D-W)x+1^T(D-W)1)+\frac{2(1-2k)}{(1-k)^2}1^T(D-W)x}{\frac{k}{1-k}1^T1} \\
&\quad +\frac{2(x^T(D-W)x)}{1^T1}
\end{aligned} \tag{2}
$$

令 $b=\frac{k}{1-k}$，因为 $1^T(D-W)1=0$；所以 4RatioCut(A,B) 可以做如下转换。

$$
\begin{aligned}
4RatioCut(A,B) &= \frac{(1+b^2)(x^T(D-W)x+1^T(D-W)1)+2(1-b^2)1^T(D-W)x}{b1^T1} \\
&\quad +\frac{2b(x^T(D-W)x)}{b1^T1} \\
&= \frac{(1+b^2)(x^T(D-W)x+1^T(D-W)1)+2(1-b^2)1^T(D-W)x}{b1^T1} \\
&\quad +\frac{2b(x^T(D-W)x)}{b1^T1}-\frac{2b1^T(D-W)1}{b1^T1} \\
&= \frac{(1+x)^T(D-W)(1+x)}{b1^T1}+\frac{b^2(1-x)^T(D-W)(1-x)}{b1^T1} \\
&\quad -\frac{2b(1-x)^T(D-W)(1-x)}{b1^T1} \\
&= \frac{[(1+x)-b(1-x)]^T(D-W)[(1+x)-b(1-x)]}{b1^T1} \\
&= \frac{[\frac{1}{\sqrt{b}}(1+x)-\sqrt{b}(1-x)]^T(D-W)[\frac{1}{\sqrt{b}}(1+x)-\sqrt{b}(1-x)}{1^T1}
\end{aligned} \tag{3}
$$

令 $y=\frac{1}{\sqrt{b}}(1+x)-\sqrt{b}(1-x)$，则

$$
4RatioCut(A,B)=\frac{y^T(D-W)y}{1^T1} \tag{4}
$$

所以

$$
min_x RatioCut(x)=min_y\frac{y^T(D-W)y}{1^T1}=min_y\big(y^T(D-W)y\big) \tag{5}
$$

上述公式说明，要求解 RatioCut(x) 的最小值，就是等价的求解 $\left(y^T(D-W)y\right)$ 的最小值。而求解 $y^T(D-W)y$ 的最小值就是等价地求解 $(D-W)$ 的最小特征值，y 即是对应的特征向量。

这就是 $L := D-W$ 的来由。

$$L := D - W \tag{6}$$

NCut 推导 同“RatioCut 推导”类似。

给出图 V 的一个划分 A 和 B。为了便于数学推导，如果点 i 属于 A，则令 $x_i = 1$；如果点 i 属于 B，则令 $x_i = -1$。则

$$\begin{aligned} NCut(A,B) &= \frac{cut(A,B)}{vol(A)} + \frac{cut(A,B)}{vol(B)} \\ &= \frac{\sum_{x_i>0,x_j<0} -w_{ij}x_ix_j}{\sum_{x_i>0} d_i} + \frac{\sum_{x_i<0,x_j>0} -w_{ij}x_ix_j}{\sum_{x_i<0} d_i} \end{aligned} \tag{7}$$

同 RatioCut 的处理类似。

1) 令 D 是一个以 $d(i) = \sum_j w(i,j)$ 为对角线的对角阵；
2) 令 W 是一个以 w_{ij} 为元素的对称阵；
3) 应用 $\frac{1+x}{2}$ 和 $\frac{1-x}{2}$ 代替 $x_i = 1$ 和 $x_i = -1$；
4) $k = \frac{\sum_{x_i>0} d_i}{\sum_i d_i} = \frac{\sum_{x_i>0} d_i}{1^T D1}$。

这样处理以后可以将 $NCut(A,B)$ 装换为矩阵的形式。

$$\begin{aligned} 4NCut(A,B) &= \frac{(1+x)^T(D-W)(1+x)}{k1^TD1} + \frac{(1-x)^T(D-W)(1-x)}{(1-k)1^TD1} \\ &= \frac{(x^T(D-W)x + 1^T(D-W)1)}{k(1-k)1^TD1} + \frac{2(1-2k)1^T(D-W)x}{k(1-k)1^TD1} \\ &= \frac{(x^T(D-W)x + 1^T(D-W)1) + 2(1-2k)1^T(D-W)x}{k(1-k)1^TD1} \\ &\quad - \frac{2(x^T(D-W)x + 1^T(D-W)1)}{1^TD1} + \frac{2(x^T(D-W)x)}{1^TD1} + \frac{2(1^T(D-W)1)}{1^TD1} \\ &= \frac{(1-2k+2k^2)(x^T(D-W)x + 1^T(D-W)1) + 2(1-2k)1^T(D-W)x}{k(1-k)1^TD1} \\ &\quad + \frac{2(x^T(D-W)x)}{1^TD1} \\ &= \frac{\frac{(1-2k+2k^2)}{(1-k)^2}(x^T(D-W)x + 1^T(D-W)1) + \frac{2(1-2k)}{(1-k)^2}1^T(D-W)x}{\frac{k}{1-k}1^TD1} \\ &\quad + \frac{2(x^T(D-W)x)}{1^TD1} \end{aligned} \tag{8}$$

令 $b = \frac{k}{1-k}$，因为 $1^T(D-W)1 = 0$；所以 4NCut(A,B) 可以做如下转换。

$$
\begin{aligned}
4NCut(A,B) &= \frac{(1+b^2)(x^T(D-W)x+1^T(D-W)1)+2(1-b^2)1^T(D-W)x}{b1^TD1} \\
&+ \frac{2b(x^T(D-W)x)}{b1^TD1} \\
&= \frac{(1+b^2)(x^T(D-W)x+1^T(D-W)1)+2(1-b^2)1^T(D-W)x}{b1^TD1} \\
&+ \frac{2b(x^T(D-W)x)}{b1^TD1} - \frac{2b1^T(D-W)1}{b1^TD1} \\
&= \frac{(1+x)^T(D-W)(1+x)}{b1^TD1} + \frac{b^2(1-x)^T(D-W)(1-x)}{b1^TD1} \\
&- \frac{2b(1-x)^T(D-W)(1-x)}{b1^TD1} \\
&= \frac{[(1+x)-b(1-x)]^T(D-W)[(1+x)-b(1-x)]}{b1^TD1}
\end{aligned} \tag{9}
$$

令 $y=(1+x)-b(1-x)$，则

$$
4NCut(A,B) = \frac{y^T(D-W)y}{b1^TD1} \tag{10}
$$

由于 $b=\frac{k}{1-k}=\frac{\sum_{x_i>0} d_i}{\sum_{x_i<0} d_i}$，有下面变换：

$$
y^TDy = \sum_{x_i>0} d_i + b^2\sum_{x_i<0} d_i = b\sum_{x_i<0} d_i + b^2\sum_{x_i<0} d_i = b(\sum_{x_i<0} d_i + b\sum_{x_i<0} d_i) = b1^TD1
$$

所以

$$
4NCut(A,B) = \frac{y^T(D-W)y}{y^TDy} \tag{11}
$$

所以

$$
min_x NCut(x) = min_y \frac{y^T(D-W)y}{y^TDy} \tag{12}
$$

如果了解广义瑞利商 (Rayleigh quotient)；会发现 $\frac{y^T(D-W)y}{y^TDy}$ 的最小值求解就是求解 $(D-W)y=\lambda Dy$ 广义特征值的最小值。

到这里 NCut(x) 的求解已经非常接近目标了；通过广义 Rayleigh 商求解就可以得到。如果不熟悉广义 Rayleigh 商的读者可以参考《矩阵分析与应用》。

求解 $(D-W)y=\lambda Dy$ 和求解 $D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}}z=\lambda z$ 等价，其中 $z=D^{\frac{1}{2}}y$。因为 D 是对角阵，所以 $(D-W)y=\lambda Dy$ 和 $D^{-1}(D-W)y=\lambda y$ 等价。

这就是 L_{sym} 和 L_{rw} 的来由

$$
\begin{aligned}
L_{sym} &:= D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}} \\
L_{rw} &:= D^{-1}(D-W)
\end{aligned} \tag{13}
$$

6 L_{sym} 求解 matlab 代码

可以在 https://github.com/ArrowLuo/Spectral_cluster_matlab 找到如下代码和测试数据。

```
sigma = 0.2;
k = 2;

colors = ['gh'; 'rd'; 'co'; 'ms'; 'yh';  'wo'; 'gs'; 'rh'; 'cd'; 'mo'; 'ys'; 'wd'; 'go'; 'rs'; 'ch'; 'md'; 'yo'; 'wh'];

data = dlmread('data/twocircles.data');
figure();
clf;
hold on;

[n,m] = size(data);

W = zeros(n, n);
for i=1:n
    for j=1:n
        if i ~= j
            dist = norm(data(i, :)-data(j, :));
            W(i, j) = exp(-(dist * dist)/(2*sigma*sigma));
        end
    end
end

D = diag(sum(W));
DSR = inv(sqrtm(D));

L = DSR *  W * DSR;
[X1, DD] = eigs(L, k, 'LM');

Y = zeros(n, k);
for i=1:n
    Y(i, :) = X1(i, :)./norm(X1(i, :));
end

[MInd, kM] = kmeans(Y, k);

for i=1:k
    scatter(data(MInd == i,1), data(MInd == i,2), 15, char(colors(i, :)));
end
hold off;
```