

刘建平Pinard

十年码农，对数学统计学，数据挖掘，机器学习，大数据平台，大数据平台应用开发，大数据可视化感兴趣。

博客园    首页    新随笔    联系    订阅    管理

谱聚类 ( spectral clustering ) 原理总结

谱聚类 ( spectral clustering ) 是广泛使用的聚类算法，比起传统的K-Means算法，谱聚类对数据分布的适应性更强，聚类效果也很优秀，同时聚类的计算量也小很多，更加难能可贵的是实现起来也不复杂。在处理实际的聚类问题时，个人认为谱聚类是应该首先考虑的几种算法之一。下面我们就对谱聚类的算法原理做一个总结。

1. 谱聚类概述

谱聚类是从图论中演化出来的算法，后来在聚类中得到了广泛的应用。它的主要思想是把所有的数据看做空间中的点，这些点之间可以用边连接起来。距离较远的两个点之间的边权重值较低，而距离较近的两个点之间的边权重值较高，通过对所有数据点组成的图进行切图，让切图后不同的子图间边权重和尽可能的低，而子图内的边权重和尽可能的高，从而达到聚类的目的。

乍一看，这个算法原理的确简单，但是要完全理解这个算法的话，需要对图论中的无向图，线性代数和矩阵分析都有一定的了解。下面我们就从这些需要的基础知识开始，一步步学习谱聚类。

2. 谱聚类基础之一：无向权重图

由于谱聚类是基于图论的，因此我们首先温习下图的概念。对于一个图 $G$ ，我们一般用点的集合 $V$ 和边的集合 $E$ 来描述。即为 $G(V, E)$ 。其中 $V$ 即为我们数据集里面所有的点 $(v_1, v_2, \dots, v_n)$ 。对于 $V$ 中的任意两个点，可以有边连接，也可以没有边连接。我们定义权重 $w_{ij}$ 为点 $v_i$ 和点 $v_j$ 之间的权重。由于我们是无向图，所以 $w_{ij} = w_{ji}$ 。

对于有边连接的两个点 $v_i$ 和 $v_j$ ， $w_{ij} > 0$ ，对于没有边连接的两个点 $v_i$ 和 $v_j$ ， $w_{ij} = 0$ 。对于图中的任意一个点 $v_i$ ，它的度 $d_i$ 定义为和它相连的所有边的权重之和，即

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

利用每个点度的定义，我们可以得到一个 $n \times n$ 的度矩阵 $D$ ，它是一个对角矩阵，只有主对角线有值，对应第 $i$ 行的第 $i$ 个点的度数，定义如下：

$$D = \begin{pmatrix} d_1 & \dots & \dots \\ \dots & d_2 & \dots \\ \vdots & \vdots & \ddots \\ \dots & \dots & d_n \end{pmatrix}$$

利用所有点之间的权重值，我们可以得到图的邻接矩阵 $W$ ，它也是一个 $n \times n$ 的矩阵，第 $i$ 行的第 $j$ 个值对应我们的权重 $w_{ij}$ 。

除此之外，对于点集 $V$ 的一个子集 $A \subset V$ ，我们定义：

$|A| := \text{子集} A \text{ 中点的个数}$

$vol(A) := \sum_{i \in A} d_i$

3. 谱聚类基础之二：相似矩阵

在上一节我们讲到了邻接矩阵 $W$ ，它是由任意两点之间的权重值 $w_{ij}$ 组成的矩阵。通常我们可以自己输入权重，但是在谱聚类中，我们只有数据点的定义，并没有直接给出这个邻接矩阵，那么怎么得到这个邻接矩阵呢？

基本思想是，距离较远的两个点之间的边权重值较低，而距离较近的两个点之间的边权重值较高，不过这仅仅是定性，我们需要定量的权重值。一般来说，我们可以通过样本点距离度量的相似矩阵 $S$ 来获得邻接矩阵 $W$ 。

构建邻接矩阵 $W$ 的方法有三类。 $\epsilon$ -邻近法，K邻近法和全连接法。

对于 $\epsilon$ -邻近法，它设置了一个距离阈值 $\epsilon$ ，然后用欧式距离 $s_{ij}$ 度量任意两点 $x_i$ 和 $x_j$ 的距离。即相似矩阵的 $s_{ij} = ||x_i - x_j||_2^2$ ，然后根据 $s_{ij}$ 和 $\epsilon$ 的大小关系，来定义邻接矩阵 $W$ 如下：

$$W_{ij} = \begin{cases} 0 & s_{ij} > \epsilon \\ \epsilon & s_{ij} \leq \epsilon \end{cases}$$

从上式可见，两点间的权重要不就是 $\epsilon$ ，要不就是0，没有其他的信息了。距离远近度量很不精确，因此在实际应用中，我们很少使用 $\epsilon$ -邻近法。

公告

★珠江追梦，饮岭南茶，恋鄂北家★  
昵称：刘建平Pinard  
园龄：1年2个月  
粉丝：689  
关注：13  
+加关注

2017年12月						
日	一	二	三	四	五	六
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签

随笔分类(101)

0040. 数学统计学(4)  
0081. 机器学习(62)  
0082. 深度学习(10)  
0083. 自然语言处理(23)  
0121. 大数据挖掘(1)  
0122. 大数据平台(1)  
0123. 大数据可视化

随笔档案(101)

2017年8月 (1)  
2017年7月 (3)  
2017年6月 (8)  
2017年5月 (7)  
2017年4月 (5)  
2017年3月 (10)  
2017年2月 (7)  
2017年1月 (13)  
2016年12月 (17)  
2016年11月 (22)  
2016年10月 (8)

常去的机器学习网站

52 NLP  
Analytics Vidhya

第二种定义邻接矩阵  $W$  的方法是K邻近法，利用KNN算法遍历所有的样本点，取每个样本最近的  $k$  个点作为近邻，只有和样本距离最近的  $k$  个点之间的  $w_{ij} > 0$ 。但是这种方法会造成重构之后的邻接矩阵  $W$  非对称，我们后面的算法需要对称邻接矩阵。为了解决这种问题，一般采取下面两种方法之一：

第一种K邻近法是只要一个点在另一个点的K近邻中，则保留  $S_{ij}$

$$W_{ij} = W_{ji} = \begin{cases} 0 & x_i \notin KNN(x_j) \text{ and } x_j \notin KNN(x_i) \\ \exp(-\frac{\|x_i-x_j\|_2^2}{2\sigma^2}) & x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i) \end{cases}$$

第二种K邻近法是必须两个点互为K近邻中，才能保留  $S_{ij}$

$$W_{ij} = W_{ji} = \begin{cases} 0 & x_i \notin KNN(x_j) \text{ or } x_j \notin KNN(x_i) \\ \exp(-\frac{\|x_i-x_j\|_2^2}{2\sigma^2}) & x_i \in KNN(x_j) \text{ and } x_j \in KNN(x_i) \end{cases}$$

第三种定义邻接矩阵  $W$  的方法是全连接法，相比前两种方法，第三种方法所有的点之间的权重值都大于0，因此称之为全连接法。可以选择不同的核函数来定义边权重，常用的有多项式核函数，高斯核函数和Sigmoid核函数。最常用的是高斯核函数RBF，此时相似矩阵和邻接矩阵相同：

$$W_{ij} = S_{ij} = \exp(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2})$$

在实际的应用中，使用第三种全连接法来建立邻接矩阵是最普遍的，而在全连接法中使用高斯径向核RBF是最普遍的。

## 4. 谱聚类基础之三：拉普拉斯矩阵

单独把拉普拉斯矩阵(Graph Laplacians)拿出来介绍是因为后面的算法和这个矩阵的性质息息相关。它的定义很简单，拉普拉斯矩阵  $L = D - W$ 。  $D$  即为我们第二节讲的度矩阵，它是一个对角矩阵。而  $W$  即为我们第二节讲的邻接矩阵，它可以由我们第三节的方法构建出。

拉普拉斯矩阵有一些很好的性质如下：

- 1) 拉普拉斯矩阵是对称矩阵，这可以由  $D$  和  $W$  都是对称矩阵而得。
- 2) 由于拉普拉斯矩阵是对称矩阵，则它的所有的特征值都是实数。
- 3) 对于任意的向量  $f$ ，我们有

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

这个利用拉普拉斯矩阵的定义很容易得到如下：

$$\begin{aligned} f^T L f &= f^T D f - f^T W f = \sum_{i=1}^n d_i f_i^2 - \sum_{i,j=1}^n w_{ij} f_i f_j \\ &= \frac{1}{2} (\sum_{i=1}^n d_i f_i^2 - 2 \sum_{i,j=1}^n w_{ij} f_i f_j + \sum_{j=1}^n d_j f_j^2) = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 \end{aligned}$$

4) 拉普拉斯矩阵是半正定的，且对应的  $n$  个实数特征值都大于等于0，即  $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ ，且最小的特征值为0，这个由性质3很容易得出。

## 5. 谱聚类基础之四：无向图切图

对于无向图  $G$  的切图，我们的目标是将图  $G(V, E)$  切成相互没有连接的  $k$  个子图，每个子图点的集合为： $A_1, A_2, \dots, A_k$ ，它们满足  $A_i \cap A_j = \emptyset$  且  $A_1 \cup A_2 \cup \dots \cup A_k = V$ 。

对于任意两个子图点的集合  $A, B \subset V, A \cap B = \emptyset$ ，我们定义  $A$  和  $B$  之间的切图权重为：

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

那么对于我们  $k$  个子图点的集合： $A_1, A_2, \dots, A_k$ ，我们定义切图cut为：

$$cut(A_1, A_2, \dots, A_k) = \frac{1}{2} \sum_{i=1}^k W(A_i, \overline{A_i})$$

其中  $\overline{A_i}$  为  $A_i$  的补集，意为除  $A_i$  子集外其他  $V$  的子集的并集。

那么如何切图可以让子图内的点权重和高，子图间的点权重和低呢？一个自然的想法就是最小化  $cut(A_1, A_2, \dots, A_k)$ 。但是可以发现，这种极小化的切图存在问题，如下图：

机器学习库

机器学习路线图

深度学习进阶书

深度学习入门书

积分与排名

积分 - 263288

排名 - 727

阅读排行榜

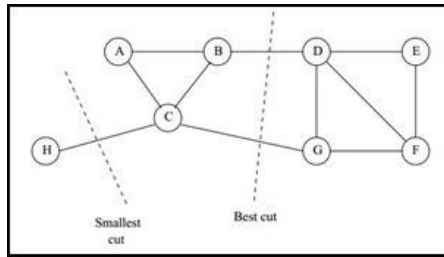
- 1. 梯度下降（Gradient Descent）小结(56483)
- 2. 梯度提升树(GBDT)原理小结(25054)
- 3. scikit-learn决策树算法类库使用小结(19972)
- 4. 线性判别分析LDA原理总结(17311)
- 5. scikit-learn随机森林调参小结(13961)

评论排行榜

- 1. 梯度提升树(GBDT)原理小结(44)
- 2. 集成学习之Adaboost算法原理小结(36)
- 3. 梯度下降（Gradient Descent）小结(36)
- 4. 线性回归原理小结(35)
- 5. 文本主题模型之LDA(二) LDA求解之Gibbs采样算法(34)

推荐排行榜

- 1. 梯度下降（Gradient Descent）小结(23)
- 2. 协同过滤推荐算法总结(9)
- 3. 卷积神经网络(CNN)反向传播算法(8)
- 4. Lasso回归算法：坐标轴下降法与最小角回归法小结(8)
- 5. scikit-learn决策树算法类库使用小结(8)



我们选择一个权重最小的边缘的点，比如C和H之间进行cut，这样可以最小化 $cut(A_1, A_2, \dots, A_k)$ ，但却不是最优的切图，如何避免这种切图，并且找到类似图中“Best Cut”这样的最优切图呢？我们下一节就来看看谱聚类使用的切图方法。

## 6. 谱聚类之切图聚类

为了避免最小切图导致的切图效果不佳，我们需要对每个子图的规模做出限定，一般来说，有两种切图方式，第一种是RatioCut，第二种是Ncut。下面我们分别加以介绍。

### 6.1 RatioCut切图

RatioCut切图为了避免第五节的最小切图，对每个切图，不光考虑最小化 $cut(A_1, A_2, \dots, A_k)$ ，它还同时考虑最大化每个子图点的个数，即：

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

那么怎么最小化这个RatioCut函数呢？牛人们发现，RatioCut函数可以通过如下方式表示。

我们引入指示向量 $h_j = \{h_1, h_2, \dots, h_k\}$   $j = 1, 2, \dots, k$  对于任意一个向量 $h_j$ ，它是一个n维向量（n为样本数），我们定义 $h_{ji}$ 为：

$$h_{ji} = \begin{cases} 0 & v_i \notin A_j \\ \frac{1}{\sqrt{|A_j|}} & v_i \in A_j \end{cases}$$

那么我们对于 $h_i^T L h_i$ ，有：

$$h_i^T L h_i = \frac{1}{2} \sum_{m=1} \sum_{n=1} w_{mn} (h_{im} - h_{in})^2 \quad (1)$$

$$= \frac{1}{2} \left( \sum_{m \in A_i, n \notin A_i} w_{mn} \left( \frac{1}{\sqrt{|A_i|}} - 0 \right)^2 + \sum_{m \notin A_i, n \in A_i} w_{mn} \left( 0 - \frac{1}{\sqrt{|A_i|}} \right)^2 \right) \quad (2)$$

$$= \frac{1}{2} \left( \sum_{m \in A_i, n \notin A_i} w_{mn} \frac{1}{|A_i|} + \sum_{m \notin A_i, n \in A_i} w_{mn} \frac{1}{|A_i|} \right) \quad (3)$$

$$= \frac{1}{2} \left( cut(A_i, \bar{A}_i) \frac{1}{|A_i|} + cut(\bar{A}_i, A_i) \frac{1}{|A_i|} \right) \quad (4)$$

$$= \frac{cut(A_i, \bar{A}_i)}{|A_i|} \quad (5)$$

$$= RatioCut(A_i, \bar{A}_i) \quad (6)$$

上述第（1）式用了上面第四节的拉普拉斯矩阵的性质3。第二式用到了指示向量的定义。可以看出，对于某一个子图i，它的RatioCut对应于 $h_i^T L h_i$ ，那么我们的k个子图呢？对应的RatioCut函数表达式为：

$$RatioCut(A_1, A_2, \dots, A_k) = \sum_{i=1}^k h_i^T L h_i = \sum_{i=1}^k (H^T L H)_{ii} = tr(H^T L H)$$

其中 $tr(H^T L H)$ 为矩阵的迹。也就是说，我们的RatioCut切图，实际上就是最小化我们的 $tr(H^T L H)$ 。注意到 $H^T H = I$ ，则我们的切图优化目标为：

$$\arg \min_H tr(H^T L H) \quad s. t. \quad H^T H = I$$

注意到我们H矩阵里面的每一个指示向量都是n维的，向量中每个变量的取值为0或者 $\frac{1}{\sqrt{|A_j|}}$ ，就有 $2^n$ 种取值，有k个子图的话就有k个指示向量，共有 $k2^n$ 种H，因此找到满足上面优化目标的H是一个NP难的问题。那么是不是就没有办法了呢？

注意观察 $tr(H^T L H)$ 中每一个优化子目标 $h_i^T L h_i$ ，其中 $h$ 是单位正交基，L为对称矩阵，此时 $h_i^T L h_i$ 的最大值为L的最大特征值，最小值是L的最小特征值。如果你对主成分分析PCA很熟悉的话，这里很好理解。在PCA中，我们的目标是找到协方差矩阵（对应此处的拉普拉斯矩阵L）的最大的特征值，而在我们的谱聚类中，我们的目标是找到目标的最小的特征值，得到对应的特征向量，此时对应二分切图效果最佳。也就是说，我们这里要用到维度规约的思想来近似去解决这个NP难的问题。

对于 $h_i^T L h_i$ ，我们的目标是找到最小的L的特征值，而对于 $tr(H^T L H) = \sum_{i=1}^k h_i^T L h_i$ ，则我们的目标就是找到k个最小的特征值，一般来说，k远远小于n，也就是说，此时我们进行了维度规约，将维度从n降到了k，从而近似可以解决这个NP难的问题。

通过找到L的最小的k个特征值，可以得到对应的k个特征向量，这k个特征向量组成一个nxk维度的矩阵，即为我们的H。一般需要对H矩阵按行做标准化，即

$$h_{ij}^* = \frac{h_{ij}}{(\sum_{i=1}^k h_{ii}^2)^{1/2}}$$

由于我们在使用维度规约的时候损失了少量信息，导致得到的优化后的指示向量h对应的H现在不能完全指示各样本的归属，因此一般在得到nxk维度的矩阵H后还需要对每一行进行一次传统的聚类，比如使用K-Means聚类。

## 6.2 Ncut切图

Ncut切图和RatioCut切图很类似，但是把Ratiocut的分母 $|A_i|$ 换成 $vol(A_i)$ 。由于子图样本的个数多并不一定权重就大，我们切图时基于权重也更合我们的目标，因此一般来说Ncut切图优于RatioCut切图。

$$NCut(A_1, A_2, \dots, A_k) = \frac{1}{2} \sum_{j=1}^k \frac{W(A_i, \bar{A}_i)}{vol(A_i)}$$

，对应的，Ncut切图对指示向量 $h$ 做了改进。注意到RatioCut切图的指示向量使用的是 $\frac{1}{\sqrt{|A_j|}}$ 标示样本归属，而Ncut切图使用了子图权重 $\frac{1}{\sqrt{vol(A_j)}}$ 来标示指示向量h，定义如下：

$$h_{ji} = \begin{cases} 0 & v_i \notin A_j \\ \frac{1}{\sqrt{vol(A_j)}} & v_i \in A_j \end{cases}$$

那么我们对于 $h_i^T L h_i$ ，有：

$$h_i^T L h_i = \frac{1}{2} \sum_{m=1} \sum_{n=1} w_{mn} (h_{im} - h_{in})^2 \quad (7)$$

$$= \frac{1}{2} \left( \sum_{m \in A_i, n \notin A_i} w_{mn} \left( \frac{1}{\sqrt{vol(A_j)}} - 0 \right)^2 + \sum_{m \notin A_i, n \in A_i} w_{mn} \left( 0 - \frac{1}{\sqrt{vol(A_j)}} \right)^2 \right) \quad (8)$$

$$= \frac{1}{2} \left( \sum_{m \in A_i, n \notin A_i} w_{mn} \frac{1}{vol(A_j)} + \sum_{m \notin A_i, n \in A_i} w_{mn} \frac{1}{vol(A_j)} \right) \quad (9)$$

$$= \frac{1}{2} \left( cut(A_i, \bar{A}_i) \frac{1}{vol(A_j)} + cut(\bar{A}_i, A_i) \frac{1}{vol(A_j)} \right) \quad (10)$$

$$= \frac{cut(A_i, \bar{A}_i)}{vol(A_j)} \quad (11)$$

$$= NCut(A_i, \bar{A}_i) \quad (12)$$

推导方式和RatioCut完全一致。也就是说，我们的优化目标仍然是

$$NCut(A_1, A_2, \dots, A_k) = \sum_{i=1}^k h_i^T L h_i = \sum_{i=1}^k (H^T L H)_{ii} = tr(H^T L H)$$

但是此时我们的 $H^T H \neq I$ ，而是 $H^T D H = I$ 。推导如下：

$$h_i^T D h_i = \sum_{j=1}^n h_{ij}^2 d_j = \frac{1}{vol(A_i)} \sum_{v_j \in A_i} w_{vj} = \frac{1}{vol(A_i)} vol(A_i) = 1$$

也就是说，此时我们的优化目标最终为：

$$\underbrace{\arg \min_H}_{H} tr(H^T L H) \quad s.t. \quad H^T D H = I$$

此时我们的H中的指示向量 $h$ 并不是标准正交基，所以在RatioCut里面的降维思想不能直接用。怎么办呢？其实只需要将指示向量矩阵H做一个小小的转化即可。

我们令 $H = D^{-1/2} F$ ，则： $H^T L H = F^T D^{-1/2} L D^{-1/2} F$ ， $H^T D H = F^T F = I$ ，也就是说优化目标变成了：

$$\underbrace{\arg \min_F}_F tr(F^T D^{-1/2} L D^{-1/2} F) \quad s.t. \quad F^T F = I$$

可以发现这个式子和RatioCut基本一致，只是中间的L变成了 $D^{-1/2} L D^{-1/2}$ 。这样我们就可以继续按照RatioCut的思想，求出 $D^{-1/2} L D^{-1/2}$ 的最小的前k个特征值，然后求出对应的特征向量，并标准化，得到最后的特征矩阵F。最后对F进行一次传统的聚类（比如K-Means）即可。

一般来说， $D^{-1/2} L D^{-1/2}$ 相当于对拉普拉斯矩阵L做了一次标准化，即 $\frac{L_{ij}}{\sqrt{vol(A_i)vol(A_j)}}$

## 7. 谱聚类算法流程

铺垫了这么久，终于可以总结下谱聚类的基本流程了。一般来说，谱聚类主要的注意点为相似矩阵的生成方式（参见第二节），切图的方式（参见第六节）以及最后的聚类方法（参见第六节）。

最常用的相似矩阵的生成方式是基于高斯核距离的全连接方式，最常用的切图方式是Ncut。而到最后常用的聚类方法为K-Means。下面以Ncut总结谱聚类算法流程。

- 输入：样本集 $D=(x_1, x_2, \dots, x_n)$ ，相似矩阵的生成方式，降维后的维度 $k_1$ ，聚类方法，聚类后的维度 $k_2$
- 输出：簇划分 $C(c_1, c_2, \dots, c_{k_2})$
- 1) 根据输入的相似矩阵的生成方式构建样本的相似矩阵S
  - 2) 根据相似矩阵S构建邻接矩阵W，构建度矩阵D
  - 3) 计算出拉普拉斯矩阵L
  - 4) 构建标准化后的拉普拉斯矩阵 $D^{-1/2}LD^{-1/2}$
  - 5) 计算 $D^{-1/2}LD^{-1/2}$ 最小的 $k_1$ 个特征值所各自对应的特征向量 $f$
  - 6) 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 $n \times k_1$ 维的特征矩阵F
  - 7) 对F中的每一行作为一个 $k_1$ 维的样本，共n个样本，用输入的聚类方法进行聚类，聚类维数为 $k_2$ 。
  - 8) 得到簇划分 $C(c_1, c_2, \dots, c_{k_2})$

## 8. 谱聚类算法总结

谱聚类算法是一个使用起来简单，但是讲清楚却不是那么容易的算法，它需要你有一定的数学基础。如果你掌握了谱聚类，相信你会对矩阵分析，图论有更深入的理解。同时对降维里的主成分分析也会加深理解。

下面总结下谱聚类算法的优缺点。

谱聚类算法的主要优点有：

- 1) 谱聚类只需要数据之间的相似度矩阵，因此对于处理稀疏数据的聚类很有效。这点传统聚类算法比如K-Means很难做到
- 2) 由于使用了降维，因此在处理高维数据聚类时的复杂度比传统聚类算法好。

谱聚类算法的主要缺点有：

- 1) 如果最终聚类的维度非常高，则由于降维的幅度不够，谱聚类的运行速度和最后的聚类效果均不好。
- 2) 聚类效果依赖于相似矩阵，不同的相似矩阵得到的最终聚类效果可能很不同。

( 欢迎转载，转载请注明出处。欢迎沟通交流： [pinard.liu@ericsson.com](mailto:pinard.liu@ericsson.com) )


分类: [0081. 机器学习](#)

标签: [聚类算法](#)

好文要顶

关注我

收藏该文



[刘建平Pinard](#)  
关注 - 13  
粉丝 - 689

[+加关注](#)

- « 上一篇：[用scikit-learn学习DBSCAN聚类](#)
- » 下一篇：[用scikit-learn学习谱聚类](#)

posted @ 2016-12-29 11:11 刘建平Pinard 阅读(12352) 评论(33) 编辑 收藏

### 评论列表

#1楼 2016-12-31 15:57 qiangges2017

都是理论的，我等小白只能瞄一眼，要是 有实例就好了，实例加代码加效果图，是我的最爱，学起来也快。太理论的就得 博主这种数学好的人来做了，，

支持(0) 反对(0)

#2楼[楼主] 2016-12-31 16:09 刘建平Pinard

@ qiangges2016  
感谢你的留言。的确谱聚类的原理数学知识多了点。其实如果只是实现算法本身的话，要不了几行代码。不放代码是因为这个算法有很多成熟实现，放上pet版的代码容易误导人。  
如果你只是希望看怎么使用谱聚类，可以看看我的谱聚类实践篇里的[用scikit-learn学习谱聚类](#)

支持(0) 反对(0)

#3楼 2017-01-02 03:28 xulu1352

太巧妙了。。。博主666，把7算法流程步骤（6） $n \times k_1$ 改下就好了用X 搭配这么好的文章，，降低美感haha

$$n \times k_1$$

支持(0) 反对(0)

#4楼[楼主] 2017-01-02 13:38 刘建平Pinard

@ xulu1352  
的确，感谢指正，已修改。

支持(0) 反对(0)

#5楼 2017-01-18 22:08 算法小丑

算法流程步骤中为什么取 $k_1$ 个特征向量，而不是 $k_2$ 个特征向量呢？

支持(0) 反对(0)

#6楼[楼主] 2017-01-19 07:12 刘建平Pinard

@ 算法小丑  
你好，这里也可以令 $k_1$ 和 $k_2$ 相等，即直接降维到 $k_2$ 。分开表示的原因是让大家知道降维的维数和后面聚类的种类数可以不用一样。

支持(1) 反对(0)

#7楼 2017-01-19 09:13 算法小丑

@ 刘建平Pinard  
谢谢你的回答，我还有一个疑问，比如有一组三维数据点集 $x$ ，我想用谱聚类的方法分成10个簇，那么我取拉普拉斯矩阵的10个特征向量，这样每个样本就变成了一个10维的数据，这算不算维度上升了呢？

支持(0) 反对(0)

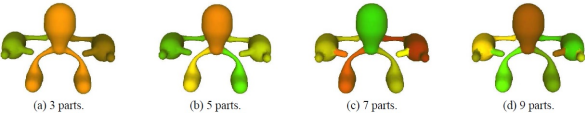
#8楼[楼主] 2017-01-19 10:07 刘建平Pinard

如果你这么使用谱聚类中的拉普拉斯矩阵，的确可以算是对数据升维。  
但是如果是谱聚类，则没人会这么做。:) 谱聚类的优势是解决高维数据和稀疏数据的聚类问题。你把维度变高了，最后一步的K-Means聚类起来就更加复杂了。

支持(0) 反对(0)

#9楼 2017-01-19 11:52 算法小丑

@ 刘建平Pinard  
谢谢你的回答，我是做三维网格处理的，最近在看利用谱聚类对三维网格分割的算法，三维网格数据也可以看成一张图 $G(V, E)$ ，如果我把三维网格分成多个( $>3$ )部分，如下图，那么就要取多个( $>3$ )特征向量，那么我的理解也是把数据给升维了，但一般资料都是说利用谱聚类对数据降维，所以比较纠结这个疑问。



支持(0) 反对(0)

#10楼[楼主] 2017-01-19 12:15 刘建平Pinard

@ 算法小丑  
你的理解没有错。一般资料都是说利用谱聚类对数据降维，主要是因为谱聚类在高维数据聚类上的表现很优秀，同时算法的目标优化利用了降维的思想。其实在低维度聚类上，谱聚类表现也很好。

支持(0) 反对(0)

#11楼 2017-01-22 16:17 算法小丑

@ 刘建平Pinard  
我不太明白为什么RatioCut切图求出的特征向量不用单位化，而Ncut切图求出的特征向量需要单位化？

支持(0) 反对(0)

#12楼[楼主] 2017-01-22 16:21 刘建平Pinard

@ 算法小丑  
你好，都需要单位化的。我里面也都提到了。

通过找到L的最小的 $k$ 个特征值，可以得到对应的 $k$ 个特征向量，这 $k$ 个特征向量组成一个 $n \times k$ 维度的矩阵，即为我们的H。  
一般需要对H里的每一个特征向量做标准化，即 $h_i = h_i / |h_i|$

支持(0) 反对(0)

#13楼 2017-07-24 18:08 有杀气

超级感谢作者，大赞

支持(0) 反对(0)

#14楼	2017-08-17 09:05	NedLevin	6.2中的（7）式k有歧义	支持(0)	反对(0)
#15楼	[楼主]	2017-08-17 10:47	刘建平Pinard  @ NedLevin 你好，感谢指出错误，这里是单个的，所以子图个数 $k$ 不能放在上面。已改正。同时改正6.1对应的位置。	支持(0)	反对(0)
#16楼	2017-09-08 16:05	公羽世无双	矩阵知识大多忘了，看不懂，好气啊！在评论里插个眼，以后好找回来，再看！！	支持(0)	反对(0)
#17楼	2017-09-14 17:43	郝一二三	谱聚类与变分混合高斯（添加Dirichlet distribution），哪个效果更好呢？还是各有千秋？	支持(0)	反对(0)
#18楼	[楼主]	2017-09-15 11:06	刘建平Pinard  @ 郝一二三 你好，这个问题很难一下说清楚，因为这与你的训练数据集的分布，特征，以及你需要求解的问题有关。很难说哪个一定比另一个好，只能说在某些特定的问题中，一个比另一个效果更好。需要在实际中去尝试比较。	支持(0)	反对(0)
#19楼	2017-10-19 17:03	老王桑	博主您好，有两个问题要请教您。  首先是ratio cut 与mininum cut 在算法上有什么区别。我觉得两者都是求出L的特征向量，然后根据特征向量进行聚类。但是其中肯定是有区别的，不知道却别在哪里。  第二个问题是，在normalized cut 中， $H=D^{(-1/2)}F$ ，那么最后在聚类的时候，是对F 聚类还是对H 聚类？	支持(0)	反对(0)
#20楼	[楼主]	2017-10-20 10:16	刘建平Pinard  @ 老王桑 你好 1）两个方法的求解过程基本一样，主要的区别在于切图的原则不同。RatioCut在切图时除了损失函数以外，还考虑每个子图的点的个数尽可能的多。 Ncut则在切图时除了损失函数以外，还考虑子图的权重大小。  2）这个问题我在第7节有讲，是对F进行聚类。	支持(0)	反对(0)
#21楼	2017-10-20 10:24	老王桑	@ 刘建平Pinard 博主您好  感谢您的回复，我不理解的地方是ratio cut 是如何考虑节点数的。即是ratio cut 和minimum cut（最小切图，不是normalized cut）有什么区别。  很明显的normalized cut 与最小切图相比，增加了权重矩阵D。但是ratio cut 与最小切相比，依然是仅仅使用普通的拉普拉斯矩阵，区别仅仅是H 的取值方式不同。  那么这种不同如何体现，或者说ratio cut 在求完L 的特征值和特征向量后的步骤是什么样的。  谢谢博主。	支持(0)	反对(0)
#22楼	[楼主]	2017-10-20 10:48	刘建平Pinard  @ 老王桑 你好，之前看错你的问题，不好意思。  我的理解是，如果使用minimum cut,则损失函数中任意子图 $i$ 的 $ A_i  = 1$ ,使用radio cut,则 $ A_i $ 是实际的子图中节点的个数。除了损失函数的考量，其余部分其实没有区别。  可以认为radio cut是minimum cut的一个进阶。而Ncut又是radioCut的进阶。  而radio cut后面的步骤可以参考第7节的Ncut的步骤，也没有什么区别。	支持(0)	反对(0)
#23楼	2017-10-20 11:10	老王桑			

<div>@ 刘建平Pinard</div> <div>博主您好，在我的理解中，两种方法的步骤是：</div> <div>最小切图</div> <div>计算L→计算特征向量→对特征向量聚类</div> <div>ratio cut</div> <div>计算L→计算特征向量→对特征向量聚类</div> <div>所以两者的步骤是一样的，那么区别是最后聚类的方法不同吗，如果不是的话，区别在哪呢？</div>		支持(0) 反对(0)
#24楼[楼主 ] 2017-10-23 10:29 刘建平Pinard		
<div>@ 老王桑</div> <div>你好，两者的步骤是一样的，最后的聚类方法也是一样的。仅仅是损失函数不同。</div>		支持(0) 反对(0)
#25楼 2017-10-23 10:30 老王桑		
<div>@ 刘建平Pinard</div> <div>博主你好，抱歉问这么初级的问题。</div> <div>那既然步骤是完全一样的，凭什么说是两种方法？</div>		支持(0) 反对(0)
#26楼[楼主 ] 2017-10-23 10:38 刘建平Pinard		
<div>@ 老王桑</div> <div>你好，这个就看你怎么看待现在的这些算法了，对损失函数做了优化，就算一个细微的修改，也可以说它是改进了的算法。：）这个我们可以不用争论。</div>		支持(0) 反对(0)
#27楼 2017-10-23 10:47 老王桑		
<div>@ 刘建平Pinard</div> <div>好的，谢谢博主了。</div>		支持(0) 反对(0)
#28楼 2017-11-05 21:25 smallheart		
<div>博主，您好！关于谱聚类我有两点疑问：</div> <div>1、我看了Ulrike von Luxberg的一篇文章《A Tutorial on Spectral Clustering》，您的第7节Ncut算法流程的第6步是按列进行标准化的，而Ulrike von Luxberg的文章中讲到是按行进行标准化的，这两个是有矛盾的，请问是按行标准化还是按列进行标准化呢？</div> <div>2、在谱聚类已经完成之后，对于新的数据点，如何确定该数据到底属于哪一类呢？</div> <div>谢谢！</div>		支持(0) 反对(0)
#29楼[楼主 ] 2017-11-06 12:01 刘建平Pinard		
<div>@ smallheart</div> <div>你好，按行和按列标准化都是可以的，因为拉普拉斯矩阵是对称的。你可以认为是把<math>H^T</math>和<math>H</math>换了个位置而已。对结果不影响。</div> <div>至于新的点，这一般不是聚类算法要考虑的，毕竟不是监督学习算法。</div>		支持(0) 反对(0)
#30楼 2017-11-06 19:44 smallheart		
<div>@ 刘建平Pinard</div> <div>博主，您好！对于第一个问题我还是没有想通，特征向量构成的矩阵是n*k1维的，按行标准化和按列标准化怎么能一样呢？请博主指点一下，谢谢！</div>		支持(0) 反对(0)
#31楼[楼主 ] 2017-11-07 10:49 刘建平Pinard		
<div>@ smallheart</div> <div>你好，我之前理解你的意思是把整个式子里的<math>H</math>换成<math>H^T</math>，<math>H^T</math>换成<math>H</math>，<math>L</math>换成<math>L^T</math>，那么最后的得到的就是<math>k_1 \times n</math>的特征向量对应的矩阵。</div> <div>然后我去看了Luxberg07，里面并没有做这样的转置。所以上面的这个转化其实并不存在。</div> <div>而的确normalizeCut都是针对每一行的数据的，这点我之前理解错误，原文已经修改，并在6.1节修改了标准化的公式，感谢指出错误。</div>		支持(0) 反对(0)
#32楼 2017-11-10 18:55 smallheart		



@ 刘建平Pinard

博主您好！感谢您的回复，我还有一个问题：特征向量是不是先按列单位化，再按行标准化构成矩阵F?因为（1）每一个特征值对应的非单位化的特征向量有很多，（2）要满足 $(F^T) * F = I$ 的条件，其中 $F^T$ 表示F的转置。不知道我理解对不对？

支持(0) 反对(0)

#33楼[楼主] 2017-11-13 11:55 刘建平Pinard

@ smallheart

你好，是的。我们求实对称矩阵的特征向量如果不特别说明，结果都应该是标准正交基，否则结果有无穷多。而这里我们也加上了标准化的约束条件，所以更加明确。

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

最新IT新闻:

- 还花一两个小时挑房源？Airbnb将推新技术让你全方位无死角了解房源
  - 方舟子：贾跃亭以老赖第一人身份登上纽约时报
  - JS开发者：最喜欢React，Vue.js比Angular值得尝试
  - 5分钟将药送上800米悬崖 京东无人机立功
  - 苹果比以往任何时候都更愿意大手笔投资或收购供应商和初创公司
- » 更多新闻...

最新知识库文章:

- 以操作系统的角度述说线程与进程
  - 软件测试转型之路
  - 门内门外看招聘
  - 大道至简，职场上做人做事做管理
  - 关于编程，你的练习是不是有效的？
- » 更多知识库文章...