


谱聚类 (Spectral Clustering) 算法介绍

转载 KDQG 于 2019-06-10 14:49:14 发布 19030 收藏 202

分类专栏: 机器学习 文章标签: 谱聚类 图切 机器学习

 机器学习 专栏收录该内容

4 订阅 12 篇文章 订阅专栏

一. 前言

本来想写关于聚类系列算法的介绍，但是聚类系列的其它几个算法原理比较简单，网上有大量的教程可以查阅。这里主要是介绍一下谱聚类算法，做一个学习笔记，同时也希望对想要了解该算法的朋友有一个帮助。关于聚类的其他系列算法，这里推荐一个写的很不错的[博客](#)。

谱聚类在最近几年变得受欢迎起来，主要原因就是它实现简单，聚类效果经常优于传统的聚类算法（如 **K-Means** 算法）。刚开始学习谱聚类的时候，给人的感觉就是这个算法看上去很难，但是当真正的深入了解这个算法的时候，其实它的原理并不难，但是理解该算法还是需要一定的数学基础的。如果掌握了谱聚类算法，会对矩阵分析，图论和降维中的主成分分析等有更加深入的理解。

本文首先会简单介绍一下谱聚类的简单过程，然后再一步步的分解这些过程中的细节以及再这些过程中谱聚类是如何实现的，接着总结一下谱聚类的几个算法，再接着介绍谱聚类算法是如何用图论知识来描述，最后对本文做一个总结。下面我们就来介绍一下谱聚类的基本原理。

二. 谱聚类的基本原理介绍

2.1 谱和谱聚类

2.1.1 谱

方阵作为线性算子，它的所有特征值的全体统称为方阵的谱。方阵的谱半径为最大的特征值。矩阵A的谱半径是矩阵 $A^T A$ 的最大特征值。

2.1.2 谱聚类

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。谱聚类可以理解为将高维空间的数据映射到低维，然后在低维空间用其它聚类算法（如KMeans）进行聚类。

2.2 谱聚类算法简单描述

输入：n个样本点 $X = \{x_1, x_2, \dots, x_n\}$ 和聚类簇的数目k;

输出：聚类簇 A_1, A_2, \dots, A_k

(1) 使用下面公式计算 $n * n$ 的相似度矩阵W;

$$s_{ij} = s(x_i, x_j) = \sum_{i=1, j=1}^n \frac{\exp(-||x_i - x_j||^2)}{2\sigma^2}$$

W为 s_{ij} 组成的相似度矩阵。

(2) 使用下面公式计算度矩阵D;

$$d_i = \sum_{j=1}^n w_{ij}, \text{ 即相似度矩阵W的每一行元素之和}$$

D为 d_i 组成的 $n * n$ 对角矩阵。

(3) 计算拉普拉斯矩阵 $L = D - W$;

(4) 计算L的特征值，将特征值从小到大排序，取前k个特征值，并计算前k个特征值的特征向量 u_1, u_2, \dots, u_k ;

(5) 将上面的k个列向量组成矩阵 $U = \{u_1, u_2, \dots, u_k\} \quad U \in R^{n * k}$.

(6) 令 $y_i \in R^k$ 是第i行的向量，



KDQG

关注

32



202

4



专栏

(7) 使用k-means算法将新样本点 $Y = \{y_1, y_2, \dots, y_n\}$ 聚类成簇 C_1, C_2, \dots, C_k ;

(8) 输出簇 A_1, A_2, \dots, A_k , 其中, $A_i = \{j | y_j \in C_i\}$.

上面就是未标准化的谱聚类算法的描述。也就是先根据样本点计算相似度矩阵, 然后计算度矩阵和拉普拉斯矩阵, 接着计算拉普拉斯矩阵前k个特征值对应的特征向量, 最后将这k个特征值对应的特征向量组成 $n * k$ 的矩阵U, U的每一行成为一个新生成的样本点, 对这些新生成的样本点进行k-means聚类, 聚成k类, 最后输出聚类的结果。这就是谱聚类算法的基本思想。相比较PCA降维中取前k大的特征值对应的特征向量, 这里取得是前k小的特征值对应的特征向量。但是上述的谱聚类算法并不是最优的, 接下来我们一步一步的分解上面的步骤, 总结一下在此基础上进行优化的谱聚类的版本。

2.3 谱聚类算法中的重要属性

2.3.1 相似度矩阵介绍

相似度矩阵就是样本点中的任意两个点之间的距离度量, 在聚类算法中可以表示为距离近的点它们之间的相似度比较高, 而距离较远的点它们的相似度比较低, 甚至可以忽略。这里用三种方式表示相似度矩阵: 一是 ϵ -近邻法 (ϵ -neighborhood graph), 二是k近邻法 (k-nearest neighbor graph), 三是全连接法 (fully connected graph)。下面我们来介绍这三种方法。

(1) ϵ -neighborhood graph:

$$s_{ij} = \|x_i - x_j\|^2, \text{ 表示样本点中任意两点之间的欧式距离}$$

用此方法构造的相似度矩阵表示如下:

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

该相似度矩阵由于距离近的地点的距离表示为 ϵ , 距离远的点距离表示为0, 矩阵中没有携带关于数据集的太多的信息, 所以该方法一般很少使用, 在sklearn中也没有使用该方法。

(2) k-nearest neighbor graph:

由于每个样本点的k个近邻可能不是完全相同的, 所以用此方法构造的相似度矩阵并不是对称的。因此, 这里使用两种方式表示对称的knn相似度矩阵, 第一种方式是如果 v_i 在 v_j 的k个领域中或者 v_j 在 v_i 的k个领域中, 则 $w_{ij} = w_{ji}$ 为 v_i 与 v_j 之间的距离, 否则为 $w_{ij} = w_{ji} = 0$; 第二种方式是如果 v_i 在 v_j 的k个领域中并且 v_j 在 v_i 的k个领域中, 则 $w_{ij} = w_{ji}$ 为 v_i 与 v_j 之间的距离, 否则为 $w_{ij} = w_{ji} = 0$ 。很显然第二种方式比第一种方式生成的相似度矩阵要稀疏。这两种方式用公式表达如下:

第一种方式:

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

第二种方式:

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

(3) fully connected graph:

该方法就是在算法描述中的高斯相似度方法, 公式如下:

$$W_{ij} = W_{ji} = \sum_{i=1, j=1}^n \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$$

该方法也是最常用的方法, 在sklearn中默认的也是该方法, 表示任意两个样本点都有相似度, 但是距离较远的样本点之间相似度较低, 甚至可以忽略。这里面的参数控制着样本点的邻域宽度, 即越大表示样本点与距离较远的样本点的相似度越大, 反之亦然。



KDQG

关注

32



202

4



专栏

2.3.2 拉普拉斯矩阵介绍

对于谱聚类来说最重要的工具就是拉普拉斯矩阵了，下面我们来介绍拉普拉斯矩阵的三种表示方法。

(1) 未标准化的拉普拉斯矩阵：

未标准化的拉普拉斯矩阵定义如下：

$$L = D - W$$

其中W是上节所说的相似度矩阵，D是度矩阵，在算法描述中有介绍。很显然，W与D都是对称矩阵。

未标准化的拉普拉斯矩阵L满足下面几个性质：

(a) 对任意一个向量 $f(f \in R^n)$ 都有：

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

证明如下：

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

(b) L是对称的和半正定的，证明如下：

因为 $w_{ij} \geq 0$ ，所以 $f^T L f \geq 0$ ，所以为半正定矩阵。由于W和D都是对称矩阵，所以L为对称矩阵。

(c) L最小的特征值为0，且特征值0所对应的特征向量为全1向量，证明如下：

令 $\bar{1}$ 表示 $n \times 1$ 的全1向量，则

$$L \cdot \bar{1} = (D - W) \cdot \bar{1} = D \cdot \bar{1} - W \cdot \bar{1} = 0 \cdot \bar{1}$$

由D和W的定义可以得出上式。

(d) L有n个非负的实数特征值： $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$

(2) 标准化拉普拉斯矩阵

标准化拉普拉斯矩阵有两种表示方法，一是基于随机游走 (Random Walk) 的标准化拉普拉斯矩阵 L_{rw} 和对称标准化拉普拉斯矩阵 L_{sym} ，定义如下：

$$L_{rw} = D^{-1} L = I - D^{-1} W$$

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

标准化的拉普拉斯矩阵满足如下性质：

(a) 对任意一个向量 $f(f \in R^n)$ 都有：

$$f^T L_{rw} f = f^T L_{sym} f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$$

(b) 当且仅当 λ 是 L_{sym} 的特征值，对应的特征向量为 $w = D^{1/2} u$ 时，则 λ 是 L_{rw} 的特征值，对应的特征向量为 u ；

(c) 当且仅当 $Lu = \lambda Du$ 时， λ 是 L_{rw} 的特征值，对应的特征向量为 u ；

(d) 0是 L_{rw} 的特征值，对应的特征向量为 $\bar{1}$ ， $\bar{1}$ 为 $n \times 1$ 的全1向量；0也是 L_{sym} 的特征值，对应的特征向量为 $D^{1/2} \bar{1}$ ；

(e) L_{sym} 和 L_{rw} 是半正定矩阵并且有非负实数特征值： $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$

关于各个版本的谱聚类算法的不同之处，就是在于相似度矩阵的计算方式不同和拉普拉斯矩阵的表示方法不同，其它步骤基本相同。下面就来介绍关于谱聚类的两个比较流行的标准化算法。



KDQG

关注

32



202

4



专栏

2.4 标准化谱聚类算法介绍

2.4.1 随机游走拉普拉斯矩阵的谱聚类算法描述

输入: n 个样本点 $X = \{x_1, x_2, \dots, x_n\}$ 和聚类簇的数目 k ;

输出: 聚类簇 A_1, A_2, \dots, A_k

- (1) 计算 $n * n$ 的相似度矩阵 W ;
- (2) 计算度矩阵 D ;
- (3) 计算拉普拉斯矩阵 $L_{rw} = D^{-1}L = D^{-1}(D - W)$;
- (4) 计算 L_{rw} 的特征值, 将特征值从小到大排序, 取前 k 个特征值, 并计算前 k 个特征值的特征向量 u_1, u_2, \dots, u_k ;
- (5) 将上面的 k 个列向量组成矩阵 $U = \{u_1, u_2, \dots, u_k\}$, $U \in R^{n * k}$;
- (6) 令 $y_i \in R^k$ 是 U 的第 i 行的向量, 其中 $i = 1, 2, \dots, n$;
- (7) 使用 k -means 算法将新样本点 $Y = \{y_1, y_2, \dots, y_n\}$ 聚类成簇 C_1, C_2, \dots, C_k ;
- (8) 输出簇 A_1, A_2, \dots, A_k , 其中, $A_i = \{j | y_j \in C_i\}$

2.4.2 对称拉普拉斯矩阵的谱聚类算法描述

输入: n 个样本点 $X = \{x_1, x_2, \dots, x_n\}$ 和聚类簇的数目 k ;

输出: 聚类簇 A_1, A_2, \dots, A_k

- (1) 计算 $n * n$ 的相似度矩阵 W ;
- (2) 计算度矩阵 D ;
- (3) 计算拉普拉斯矩阵 $L_{sym} = D^{-1/2}LD^{-1/2} = D^{-1/2}(D - W)D^{-1/2}$;
- (4) 计算 L_{sym} 的特征值, 将特征值从小到大排序, 取前 k 个特征值, 并计算前 k 个特征值的特征向量 u_1, u_2, \dots, u_k ;
- (5) 将上面的 k 个列向量组成矩阵 $U = \{u_1, u_2, \dots, u_k\}$, $U \in R^{n * k}$;
- (6) 令 $y_i \in R^k$ 是 U 的第 i 行的向量, 其中 $i = 1, 2, \dots, n$;
- (7) 对于 $i = 1, 2, \dots, n$, 将 $y_i \in R^k$ 依次单位化, 使得 $|y_i| = 1$;
- (8) 使用 k -means 算法将新样本点 $Y = \{y_1, y_2, \dots, y_n\}$ 聚类成簇 C_1, C_2, \dots, C_k ;
- (9) 输出簇 A_1, A_2, \dots, A_k , 其中 $A_i = \{j | y_j \in C_i\}$.

上面两个标准化拉普拉斯算法加上未标准化拉普拉斯算法这三个算法中, 主要用到的技巧是将原始样本点 x 转化为新的样本点 y_i , 然后再对新样本点使用其它的聚类算法进行聚类, 在这里最后一步用到的聚类算法不一定非要是 KMeans 算法, 也可以是其它的聚类算法, 具体根据实际情况而定。在 sklearn 中默认是使用 KMeans 算法, 但是由于 KMeans 聚类对初始聚类中心的选择比较敏感, 从而导致 KMeans 算法不稳定, 进而导致谱聚类算法不稳定, 所以在 sklearn 中有另外一个可选项是 'discretize', 该算法对初始聚类中心的选择不敏感。

三. 用切图的观点来解释谱聚类

聚类算法给我们的直观上的感觉就是根据样本点的相似性将他们划分成不同的组, 使得在相同组内的数据点是相似的, 不同组之间的数据点是不相似的。对于给定的样本点计算相似度, 形成相似度图, 因而谱聚类的问题可以被重新描述如下: 我们想要找到图形的一个分区, 使得不同分区之间的边具有非常低的权重 (这意味着不同分区中的点彼此不相似) 并且分区内的边具有高权重 (这意味着其中的点彼此相似)。在这个小节我们将讨论如何推导谱聚类为近似的图分区的问题。

3.1 最小切 (mincut)

对于无向图 G , 我们的目标是将图 $G(V, E)$ 切成互相没有连接的子图, 每个子图点的集合为 A_1, A_2, \dots, A_k , 其中 $A_i \cap A_j = \emptyset$ 且 $A_1 \cup A_2 \cup \dots \cup A_k = V$.



KDQG

关注

32



202

4



专栏

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之间的权重切图:

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

对于k个子图集合 A_1, A_2, \dots, A_k , 我们定义切图cut为:

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

其中, \bar{A} 为A的补集。

我们可以想到的切图方法就是最小化上式, 也就是使各个组之间的权重尽可能的小, 但是在许多情况下mincut只是简单的将图中的一个定点与其余的顶点分开, 在并不是我们想要的结果, 合理的切分结果应该是组内的样本点尽可能的多。所以mincut在实际中并不常用。下面我们介绍另外两个切图方式RatioCut和Ncut。

3.2 RatioCut切图

在RatioCut切图中, 不仅要考虑使不同组之间的权重最小化, 也考虑了使每个组中的样本点尽量多。

定义 $|A_i|$ 为子集A中的顶点个数, RatioCut的表达式如下:

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

将图中顶点V分成k个分区 A_1, A_2, \dots, A_k , 我们定义指示向量为:

$h_j = (h_{1,j}, \dots, h_{n,j})^T$, 其中:

$$h_{i,j} = \begin{cases} \frac{1}{\sqrt{|A_j|}} & \text{if } v_i \in A_j \\ 0 & \text{otherwise} \end{cases} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, k$$

我们令 $H \in R^{n \times k}$ 为包含k个指示向量作为列向量的矩阵, 注意到H的列向量彼此正交, 即 $H^T H = I$, 然后我们计算下式:

$$h_i^T L h_i = \frac{cut(A_i, \bar{A}_i)}{|A_i|} = (H^T L H)_{ii} \quad , \text{证明如下:}$$

$$\begin{aligned} h_i^T L h_i &= \frac{1}{2} \sum_{m=1}^n \sum_{n=1}^k w_{mn} (h_{im} - h_{in})^2 \\ &= \frac{1}{2} \left(\sum_{m \in A_i, n \notin A_i} w_{mn} \left(\frac{1}{\sqrt{|A_i|}} - 0 \right)^2 + \sum_{m \notin A_i, n \in A_i} w_{mn} \left(0 - \frac{1}{\sqrt{|A_i|}} \right)^2 \right) \\ &= \frac{1}{2} \left(\sum_{m \in A_i, n \notin A_i} w_{mn} \frac{1}{|A_i|} + \sum_{m \notin A_i, n \in A_i} w_{mn} \frac{1}{|A_i|} \right) \\ &= \frac{1}{2} \left(\frac{cut(A_i, \bar{A}_i)}{|A_i|} + \frac{cut(\bar{A}_i, A_i)}{|A_i|} \right) \\ &= \frac{cut(A_i, \bar{A}_i)}{|A_i|} = RatioCut(A_1, \dots, A_k) \end{aligned}$$

<https://blog.csdn.net/cxk207017>

结合上面可以得到:

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

其中 $Tr(A)$ 表示矩阵A的迹, 也就是矩阵A的对角线之和。

所以我们此时的目标函数成为:

$$\min_{A_1, \dots, A_k} Tr(H^T L H)$$

, 终



KDQG

关注

32



202

4



专栏

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 $Tr(H^T LH)$ 中每一个优化子目标 $h_i^T L h_i$ ，其中 $h_i^T L h_i$ 单位正交基，L为对称矩阵。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

$Tr(H^T LH) = \sum_{i=1}^k h_i^T L h_i$ ，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 $n \times k$ 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

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

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 $n \times k$ 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。

3.3 Ncut切图

Ncut在最小化损失函数之外，还考虑了子图之间的权重大小。Ncut切图与RatioCut类似，只是把RatioCut分母中 $|A_i|$ 的替换成了 $vol(A_i)$ ，其中：

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

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。所以Ncut的目标函数如下：

$$vol(A_i) = \sum_{i \in A} d_i, \text{ 证明同上}$$

在Ncut中使用子图权重 $\frac{1}{\sqrt{vol(A_j)}}$ 来表示指示向量h，定义如下：

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

我们的优化目标函数是：

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

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

$$H^T D H = \sum_{i=1}^k h_{ij}^2 d_j = \frac{1}{vol(A_i)} \sum_{v_j \in A_i} w_{v_j} = \frac{1}{vol(A_i)} vol(A_i) = 1$$

也就是说，此时我们的优化目标最终为：

$$\min_{A_1, \dots, A_k} Tr(H^T L H) \quad \text{约束条件: } H^T D H = I$$

令 $T = D^{-1/2} H$ ，则问题变成如下形式：

$$\min_{T \in R^{n \times k}} Tr(T^T D^{-1/2} L D^{-1/2} T) \quad \text{约束条件为: } T^T T = I$$



KDQG

关注

32



202

4



专栏

可以发现这个式子和RatioCut基本一致，只是中间的L变成了 $D^{-1/2}LD^{-1/2}$ 。这样我们就可以继续按照RatioCut的思想，求出 $D^{-1/2}LD^{-1/2}$ 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T,最后对T进行一次传统的聚类（比如K-Means）即可。

一般来说， $D^{-1/2}LD^{-1/2}$ 相当于对拉普拉斯矩阵L做了一次标准化，即 $\frac{L_{ij}}{\sqrt{d_i * d_j}}$ ，所以Ncut会产生标准化的谱聚类，而RatioCut会产生未标准化的谱聚类。

四. 谱聚类算法的优缺点

4.1 优点

- (1) 当聚类的类别个数较小时，谱聚类的效果会很好，但是当聚类的类别个数较大的时候，则不建议使用谱聚类；
- (2) 谱聚类算法使用了降维的技术，所以更加适用于高维数据的聚类；
- (3) 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
- (4) 谱聚类算法建立在谱图理论上，与传统的聚类算法相比，它具有能在任意形状的样本空间上聚类且收敛于全局最优解

4.2 缺点

- (1) 谱聚类对相似度图的变化和聚类参数的选择非常的敏感；
- (2) 谱聚类适用于均衡分类问题，即各簇之间点的个数相差不大，对于簇之间点个数相差悬殊的聚类问题，谱聚类则不适用；

五. 参考文献/博文

- (1) [A Tutorial on Spectral Clustering](#)
- (2) [比较不错的博客](#)


原文：https://blog.csdn.net/qq_24519677/article/details/82291867

文章知识点与官方知识档案匹配，可进一步学习相关知识

算法技能树 首页 概览 48833 人正在系统学习中

SpectralClustering.zip_SpectralClustering_kmeans聚类matlab_twomoon 07-14
一种快速有效的谱聚类算法，很好的结合谱聚类和kmeans算法。

SpectralClustering:谱聚类的简单实现 06-16
光谱聚类 这段代码或多或少地实现了频谱聚类，如所述。该代码使用机器学习库进行矩阵操...

4 条评论  陌路人389 热评 我想问一下，6维的数据可以用谱聚类吗 写评论

...Clustering)原理及Python实现_谱聚类算法原理与实现_张博208的博客... 6-16
用matlab对图像进行频谱分析代码PyTorch光谱集群 [开发中]-使用PyTorch和Matlab等效代码...

光谱分类算法 matlab,Matlab K-means聚类算法对多光谱遥感图像进行分 ... 7-3
MatlabK-means聚类算法对多光谱遥感图像进行分类 作者: 白艺亭 测试了下matlab自带kmean...

几个聚类算法 Tc. zyh的博客 1892
文章目录一、什么是聚类二、聚类算法2.1聚类数据集2.2亲和传播2.3聚合聚类2.4K均值2.5...

谱聚类-----spectral clustering 冷月无声的博客 917
这篇论文介绍了谱聚类方法，即利用相似矩阵的光谱（特征值）来对数据降维。【论文链接...

光谱双聚算法——SpectralBiclustering_yuanzhoulvpi的博客 7-2
data, rows, columns = make_checkerboard(shape=(300,300), n_clusters=n_clusters, noise=...

聚类_光谱聚类_u200710的博客
光谱聚类Spectral Clustering SpectralC



KDQG 关注

32 202 4 专栏

聚类算法-K-means 和 DBSCAN 【机器学习笔记简摘】 简介 决策树，随机森林，PCA和逻辑回归，他们虽然有着不同的功能，但却都属于“有监督学...	YPS的博客	4601
谱聚类 (spectral clustering) 理解 谱聚类spectral clustering，构图和切图，拉普拉斯矩阵		01-30
Spectral clustering 谱聚类讲解及实现_光谱聚类_肥宅_Sean的博客-C... In multivariate statistics and the clustering of data, spectral clustering techniques make use o...		6-25
无监督显著目标检测论文阅读(一):光谱聚类投票方法 Unsupervised Salient... 首先,通过对多个 k 值的不同特征反复应用光谱聚类形成候选 mask 集。然后,将空间范围等于...		6-21
介绍谱聚类 (spectral clustering) 1、谱聚类概述 谱聚类演化于图论，后由于其表现出优秀的性能被广泛应用于聚类中，对比其...	gf1136010396的博客	1万+
非常全面的谱聚类算法原理总结 谱聚类算法是目前最流行的聚类算法之一，其性能及适用场景优于传统的聚类算法如k-均值算...	wuyy0224的博客	4703
Matlab多光谱kmeans聚类分割_Larrisaaaaa的博客 Matlab多光谱kmeans聚类分割 参考代码:https://blog.csdn.net/ma7856728/article/details/848...		7-4
聚类算法概述 聚类算法的意义 功夫大笨鲨的博客 需要注意的一件重要事情是,该模块中实现的算法可以采用不同类型的矩阵作为输入。所有方...		6-13
【Python实例第19讲】图像分割的谱聚类 机器学习训练营——机器学习爱好者的自由交流空间 (qq 群号: 696721295) 在数据的多变...	wong2016的博客	5277
【图像分割】基于谱聚类算法实现图像分割matlab源码.md 【图像分割】基于谱聚类算法实现图像分割matlab源码.md		08-18
谱聚类算法对数据点进行分类 使用谱聚类算法对matlab产生的随机数据点进行分类，共有三个案例，每个案例都可以直接运行		04-01
一种谱聚类算法 spectral 用谱聚类来分类数据。		11-28
谱聚类的分析及算法 (spectral clustering) 该论文详细介绍了谱聚类的原理、实现算法以及算法分析。		04-21
机器学习之谱聚类算法 参考https://www.cnblogs.com/pinard/p/6221564.html 谱聚类 (spectral clustering) ...	liuy9803的博客	9831
从拉普拉斯矩阵说到谱聚类 热门推荐 从谱聚类说到拉普拉斯矩阵 0 引言 11月1日上午，机器学习班第7次课，邹博讲聚类 (PPT) ...	结构之法 算法之道	8万+
scikit-learn(sklearn)学习笔记七聚类算法 1.聚类 聚类算法是监督学习中的一个典型的代表 典型的聚类算法有: K-Means、DBSCAN、...	weixin_52703681的博客	880
谱聚类 谱聚类 (spectral clustering) 原理总结: 谱聚类 (spectral clustering) 是广泛使用的聚类算...	qq_34755941的博客	334
谱聚类算法python代码 最新发布 以下是Python中的谱聚类算法代码: ```python import numpy as np from sklearn.cluster impo...		03-28

“相关推荐”对你有帮助么？

-
-
-
-
-

关于我

招贤纳士

商务合作

寻求报道

400-660-0108

kefu@csdn.net

在线客服

工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息
北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范
版权与免责声明 版权申诉 出版物许可证 营业执照 ©1999-2023北京创新乐知网络技术有限公司



KDQG

码龄8年

暂无认证

17

7万+

1万+

28万+

原创

周排名

总排名

访问

等级



KDQG

关注

32



202

4



专栏

积分

粉丝

获赞

评论

收藏

私信

关注

搜博主文章

Q

热门文章

sfc /scannow是什么意思，有什么作用？

40763

初识合成孔径雷达SAR

27444

MATLAB中白噪声的WGN和AWGN函数的使用

21503

python获取电脑屏幕分辨率

19609

谱聚类 (Spectral Clustering) 算法介绍

19025

分类专栏

刷题

1篇

MFC

1篇

合成孔径雷达

mysql

1篇

单片机java学习笔记一

5篇

最新评论

谱聚类 (Spectral Clustering) 算法介绍

陌路人389: 我想问一下，6维的数据可以用谱聚类吗

二次规划问题和MATLAB函数quadprog...

nagisajiang462: 请问是否设置初值对结果有影响吗

python文件开头# coding=gbk的用途

人间十年: 截止到2022年11月01日，还没有办法。我今天编码的时候也遇到了。

谱聚类 (Spectral Clustering) 算法介绍

weixin_47233021: 这句话讲得好：谱聚类可以理解为将高维空间的数据映射到低维

python文件开头# coding=gbk的用途

Monster....: 有没有什么方法可以不加这行注释

您愿意向朋友推荐“博客详情页”吗？

强烈不推荐

不推荐

一般般

推荐

强烈推荐

最新文章

GO 多版本管理工具之smart-go-dl

python 读取文件，因为编码问题不能遍历

KDQG

关注

322024

专栏

2023年	3篇	2020年	3篇
2019年	24篇	2018年	5篇
2017年	8篇		



目录

一. 前言

二. 谱聚类的基本原理介绍

2.1 谱和谱聚类

2.2 谱聚类算法简单描述

2.3 谱聚类算法中的重要属性 2.3.1 ...

2.3.2 拉普拉斯矩阵介绍

2.4 标准化谱聚类算法介绍 2.4.1 随...

2.4.2 对称拉普拉斯矩阵的谱聚类算...

三. 用切图的观点来解释谱聚类

3.1 最小切 (mincut)

3.2 RatioCut切图

3.3 Ncut切图

四. 谱聚类算法的优缺点

4.1 优点