**聚类**

8

聚类是一种非监督学习任务，其目的是发现数据中隐含的结构。聚类算法按照某种标准，把数据集划分成不同簇，使得同一个簇内的数据尽可能相似，不同簇中的数据差异尽可能大。

本章我们将讨论聚类算法性能的评价指标和常用的样本间相似性度量/距离度量，然后学习一些常用的聚类算法，包括K均值、混合高斯模型、层次聚类、基于密度的噪声鲁棒的聚类、基于密度峰值等算法和谱聚类。最后我们通过案例，介绍Scilit-Learn中各种聚类算法的API。

“物以类聚、人以群分”，机器学习中的聚类（Clustering）算法发现数据中隐含的结构，按照某种标准（如距离）把数据集划分成不同的类或簇，使得同一个簇内的数据尽可能相似，不同簇中的数据差异尽可能大。簇内相似性越大，簇间差距越大，说明聚类效果越好。本章我们先讨论聚类算法性能的评价指标和常用的样本间相似性度量/距离度量，然后学习一些常用的聚类算法。

## **标题2** 8.1 聚类算法的评价指标

监督学习中，由于训练集中的样本/验证集的样本有标签，所以评价指标容易理解。聚类是非监督学习，相对而言，评价指标没那么直观。根据评价数据集是否有类别标注，聚类效果评价有两类指标衡量：一类是有参考分类结果作基准的外部评价指标，一类是无参考结果的内部评价指标。

### **8.1.1** 外部评价指标

对数据集，假设通过聚类算法将样本聚为簇：，参考结果给出的簇划分为，令和分别表示与对应的簇标记向量。我们计算聚类结果和参考类别结果的样本两两配对，定义如下：

|  |  |
| --- | --- |
|  | （8-1） |

其中集合包含在聚类结果中属于相同的簇并且在参考结果中也属于相同的簇的样本对，包含在中属于相同的簇但在中属于不同的簇的样本对，包含在中属于不同的簇但在中属于相同的簇的样本对，包含在中属于不同的簇，且在中属于相同的簇的样本对。每个样本对仅能出现在一个集合中，因此有。

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 参考结果 | |
| 相同簇 | 不同簇 |
| 聚类结果 | 相同簇 |  |  |
| 不同簇 |  |  |

基于以上定义，对聚类结果，我们定义如下的性能度量指标：

#### Jaccard系数（Jaccard Coefficent，JC）

|  |  |
| --- | --- |
|  | （8-2） |

刻画了所有属于同一簇的样本对（要么在中属于同一簇，要么在中属于同一簇），同时在中属于同一簇的样本对的比例*。*，越大，聚类效果与参考结果的一致性越高。

#### FM指数（Fowlkes and Mallows Index，FMI）

|  |  |
| --- | --- |
|  | （8-3） |

令在中属于同一簇的样本对中，同时属于的样本对的比例为精确度（precision）；在中属于同一簇的样本对中，同时属于的样本对的比例为召回率召回（recall），则FM指数是精确度和召回率的几何均值。，越大，聚类效果与参考结果的一致性越高。

#### 兰德指数（Rand Index，RI）

|  |  |
| --- | --- |
|  | （8-4） |

其中为所有可能的样本对的数目，表示聚类结果和参考结果吻合的样本对的数目（在两种结果中两个样本都同属于一个簇，或者都属于不同簇）。，越大，聚类效果与参考结果的一致性越高。

#### 调整兰德指数（Adjusted rand index，ADI）

使用时有个问题，对于随机聚类，不保证接近0（可能还很大）。而利用随机聚类情况下的（）来解决这个问题，使得在聚类结果随机产生的情况下指标应近零：

|  |  |
| --- | --- |
|  | （8-5） |

，值越大意味着聚类结果与参考结果越吻合。

#### V度量（V-measure）

介绍V-measure之前，先介绍两个指标：同质性（homogeneity）和完整性（completeness）。基于下面表格，可以使用条件熵来定义同质性和完整性。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| */* |  | … |  |  |
|  |  | … |  |  |
| … | … | … | … | … |
|  |  | … |  |  |
|  |  | … |  |  |

聚类结果的熵为：

其中

在给定分类结果条件下，聚类结果的条件熵为：

其中

在给定聚类结果条件下，分类结果的条件熵为：

那么我们定义同质性为：

|  |  |
| --- | --- |
|  | （8-6） |

如果一个簇中只包含一个类别的样本，则同质性好。

完整性定义为：

|  |  |
| --- | --- |
|  | （8-7） |

如果一个类的所有元素都分配给同一个簇，则完整性好。

V度量（V-measure）是同质性和完整性的调和平均：

|  |  |
| --- | --- |
|  | （8-8） |

，值越大，聚类结果和参考结果越一致，聚类效果越好。

### **8.1.2** 内部评价指标

绝大多数情况下，我们不知道每个样本真实类别标签，没有参考结果，这时我们只能用内部评价法评估聚类的性能。

对数据集，假设通过聚类算法将样本聚为类：，我们定义如下指标：

|  |  |
| --- | --- |
|  | （8-9） |
|  |
|  |
|  |

其中表示第个簇中的样本数目，表示两个样本之间的距离，表示簇的中心。所以表示簇中所有样本对之间的平均距离，表示中距离最远的两个样本之间的距离，为两个簇的最短距离，则是两个簇中心之间的距离。因此前两个指标表示一个簇内样本之间的距离，这个距离越小越好；后两个指标表示两个簇之间的距离，这个距离越大越好。

基于上述定义式，可有以下的内部指标。

#### 戴维森堡丁指数（Davies-Bouldin Index，DBI）

|  |  |
| --- | --- |
|  | （8-10） |

为簇内距离除以簇间距离，所以越小，聚类效果越好。

#### 邓恩指数（Dunn Validity Index，DVI）

|  |  |
| --- | --- |
|  | （8-11） |

计算任意两个簇的最短距离（类间距离）除以任意簇中的最大距离（类内距离），因此越大聚类效果越好。

#### Calinski-Harabaz指数（Calinski-Harabaz Index，CHI）

|  |  |
| --- | --- |
|  | （8-12） |

其中为簇间散度矩阵，为所有样本的中心，为簇间散度矩阵的迹；为簇内散度矩阵，簇内散度矩阵的迹。为簇内散度和与簇间散度和的比值，越大，代表着簇自身越紧密，簇与簇之间越分散，聚类结果越好。

#### 轮廓系数（Silhouette coefficient，SC）

对于每个样本点 ：

（1）计算：样本点到与其所属簇中其它点的平均距离，所以与簇内散度有关；

（2）计算：样本点到其他类内所有点的平均距离，与簇间的距离有关。

（3）样本点的轮廓系数为：

|  |  |
| --- | --- |
|  | （8-13） |

。，表示该样本点离邻近的簇很远； 表示样本非常靠近相邻的簇，样本在两个簇的边界上；，表示将该样本分配给错误的簇。

将所有点的轮廓系数求平均，就是该聚类结果总的轮廓系数：

|  |  |
| --- | --- |
|  | （8-14） |

需要注意的是轮廓系数的计算复杂度高（），当样本数很大时计算慢。

## **标题2** 8.2 相似性度量

聚类的过程是将相似的样本聚成一簇，从而使得同一簇中的样本越相似越好，不同簇之间的样本越不相似越好。所以样本之间的相似性度量对聚类结果很关键。因为距离和相似度相反，有些聚类算法基于距离度量进行聚类（如均值聚类）。一个合法的距离度量函数满足下列条件：

|  |  |
| --- | --- |
|  | （8-15） |

样本间的相似性度量和具体应用有关，常用的相似性/聚类度量有：

#### 欧氏距离

欧氏距离是最易于理解的一种距离计算方法，源自欧氏空间中两点间的直线距离公式：

|  |  |
| --- | --- |
|  | （8-16） |

#### 曼哈顿距离

曼哈顿距离是一种很形象的命名。想象我们在曼哈顿要从一个十字路口开车到另外一个十字路口，驾驶距离不是两点间的直线距离，因为我们不能穿越大楼，只能沿着街区路线驾驶，驾驶距离就是“曼哈顿距离”：

|  |  |
| --- | --- |
|  | （8-17） |

#### 切比雪夫距离

在二维棋盘格上，假设走一步能够移动到相邻的8个方格中的任意一个，那么从格子走到格子最少需要的步数是步，这个距离称为切比雪夫距离：

|  |  |
| --- | --- |
|  | （8-18） |

#### 闵可夫斯基距离

闵可夫斯基氏距离不是一种距离，而是一组距离的定义：

|  |  |
| --- | --- |
|  | （8-19） |

其中是一个变参数。当时，为曼哈顿距离；当时，是欧氏距离。当时，是切比雪夫距离（可用放缩法和夹逼法则证明）。

闵可夫斯基氏距离函数对特征的旋转和平移变换不敏感，但对数值的尺度敏感。如果样本不同特征的量纲不一致，需要将数据标准化。

#### 马氏距离（Mahalanobis Distance）

个样本向量，协方差矩阵记为，均值记为，则之间的马氏距离定义为：

|  |  |
| --- | --- |
|  | （8-20） |

若协方差矩阵是单位矩阵（各个特征之间独立同分布），则马氏距离就是欧氏距离了。若协方差矩阵是对角矩阵，相当于对每维特征做标准化，然后再计算欧氏距离（标准化的欧氏距离）。因此马氏距离的优缺点是与特征的量纲无关，排除了变量之间的相关性的干扰。

#### 汉明距离（Hamming Distance）

在一个码组集合中，任意两个码字之间的汉明距离定义为对应位上码元取值不同的位的数目：

|  |  |
| --- | --- |
|  | （8-21） |

其中表示异或操作。

两个等长字符串与之间的汉明距离定义为将其中一个变为另外一个所需要作的最小替换次数（字符串“1111”与“1001”之间的汉明距离为2）。

#### 夹角余弦

若将两个样本看作维空间的两个向量，则这两个向量间的夹角余弦可以表示这两个样本之间的相似度（不是距离） ：

|  |  |
| --- | --- |
|  | （8-22） |

夹角余弦取值范围为。夹角余弦越大表示两个向量的夹角越小，夹角余弦越小表示两向量的夹角越大。当两个向量的方向重合时夹角余弦取最大值1，当两个向量的方向完全相反夹角余弦取最小值-1。

#### 相关系数

相关系数亦被称为Pearson系数，定义为：

|  |  |
| --- | --- |
|  | （8-23） |

当对数据做中心化后，，此时相关系数等于夹角余弦，即。

#### 杰卡德相似系数（Jaccard similarity coefficient）

杰卡德相似系数是衡量两个集合的相似度一种指标。将一个样本看作是包含个元素的集合，则两个集合之间的杰卡德相似系数为这两个集合的杰卡德相似系数，即两个集合交集的元素在并集中两个集合并集元素所占的比例：

|  |
| --- |
|  |

假设是两个维向量，且所有维度的取值都是0或1。例如，。我们将样本看成集合，1表示集合包含该元素，0表示集合不包含该元素。

：两个向量中对应维度都是1的维度的数目，

：对应维度为1，而对应维度是0的维度的数目，

：对应维度为0，而对应维度是1的维度的数目，

：两个向量中对应维度都是0的维度的数目，

那么样本的杰卡德相似系数可以表示为：

|  |  |
| --- | --- |
|  | （8-24） |

此处分母之所以不加的原因在于：杰卡德相似系数处理的是非对称二元变量。非对称的意思是指状态的两个输出不是同等重要的，例如，疾病检查的阳性和阴性结果。按照惯例，我们将比较重要的输出结果，通常也是出现几率较小的结果编码为1（例如HIV阳性），而将另一种结果编码为0（例如HIV阴性）。给定两个非对称二元变量，两个都取1的情况（正匹配）认为比两个都取0的情况（负匹配）更有意义。负匹配的数量认为是不重要的，因此在计算时忽略。

杰卡德相似度算法没有考虑向量中数值的大小，而是简单的处理为0和1，因此计算效率高，但也损失了很多信息。

## **标题2** 8.3 ***K***均值（***K*Means**）聚类

K均值聚类是最常用的聚类算法。虽然其性能不一定好，但速度快，因为只需计算样本点和簇中心之间的距离，具有线性复杂度 。

K均值聚类的基本思想是将样本划分到离其最近的簇中，以迭代方式实现，如算法8.1所示。

**算法**8-1**：K均值聚类**

输入：训练样本：；

簇的数目：

1．选择个点作为初始质心：；

2．Repeat

2.1 将每个点指派到离其最近的质心，形成个簇：

，

2.2 重新计算每个簇的质心：；

Until簇不发生变化或达到最大迭代次数。

输出：每个簇中心，每个样本所属簇的标记矩阵、每个簇的样本集合

上述过程也可以看成是对下述目标函数取极小值：

|  |  |
| --- | --- |
|  | （8-25） |

即样本到其所属簇中心的距离和最小，其中为样本所属簇索引，表示第个样本属于第个簇，否则。

目标函数的参数包含两部分：每个簇的中心和每个样本所属簇归属指示矩阵。在优化时我们采用坐标轴下降法，即先固定其他参数（如簇中心）优化一个参数（如簇归属指示矩阵）；然后再交换参数进行迭代优化。

在给定簇的中心的情况下，将每个点指派到离其最近的质心会使得目标函数最小。在给定每个样本所属簇的情况下，计算目标函数对参数的梯度：

从而得到：

|  |  |
| --- | --- |
|  | （8-26） |

在给定每个样本所属簇的情况下，当距离度量取欧氏距离时，簇中心为该簇样本集合的均值，这也是K均值聚类算法名称的由来。由于均值计算对噪声比较敏感，可以将簇的质心由均值换成中值，得到K中值（Medians）聚类算法。换成中值的另一个好处是中值是一个真实存在的样本，而均值通常不是一个真实样本/原型。但对于较大的数据集，K中值聚类要慢得多，因为在计算中值时，每次迭代都需要进行排序。

K均值聚类是在目标函数进行坐标轴下降优化，所以目标函数会单调下降，算法会收敛。但由于目标函数非凸，K均值不能保证收敛到全局最小值。不同的初始值可能会产生不同的聚类结果。一种常见的做法是以不同的初始值，运行K均值算法多次，再从中选择最好的结果。Scikit-Learn中构造函数中可设置参数，默认值为10，即取不同的初始值运行10次均值，取目标函数值最小那次的结果作为最终结果。

K均值聚类中，不同初始值得到最后的聚类结果可能不同，因此我们在选择初始质心时需要仔细。一个解决方案是随机确定第一个质心，其他质心的位置尽量远离已有质心。Scikit-Learn中构造函数中可设置参数实现。

K均值聚类中，我们必须选择的值，即聚成多少个类，的选择不能以上述优化的目标函数为准，因为越大，目标函数的值越小。可能的解决方案有：

（1）画出训练集上目标函数值随的变化曲线，“肘部”位置的为最佳的，即随着的增加，目标函数的值不再显著下降；一个示例如图8-1所示。

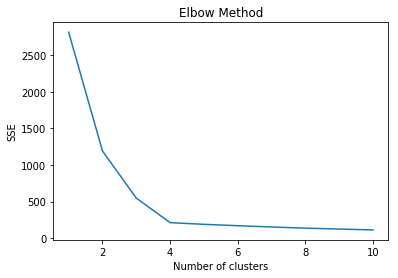


图8-1 用“肘部”法选择均值聚类中最佳的，这里最佳的

（2）根据8.1节中的评价指标，如，做为准则选择即最佳的K的值；

（3）若聚类是一个有监督学习任务的一部分，可以以监督学习任务的评价指标为准则选取。

K均值聚类简单快速，但也有很多缺点：

（1）K均值聚类需要指定簇的数目。如图8-2左上角图所示，K不正确时，聚类结果不好。

（2）K均值聚类根据样本点到簇中心的欧氏距离指派样本所属簇，相当于假设簇的形状为球形高斯分布。所以当数据集不满足这些假设时，K均值聚类的效果不好（如图8-2右上角图所示），我们需要考虑其他方法。

（3）K均值聚类直接用欧氏距离作为指标，相当于假设考虑各个簇的方差/散布程度相同。当数据不符合假设时，聚类效果不好（如图8-2左下角图所示）。

（4）在K均值聚类中，每个簇的地位相同，相当于假设每个簇的概率相等（每个簇的大小相等，密度相等），所以不能处理每个簇样本数差异很大的情况（如图8-2右下角图所示）。

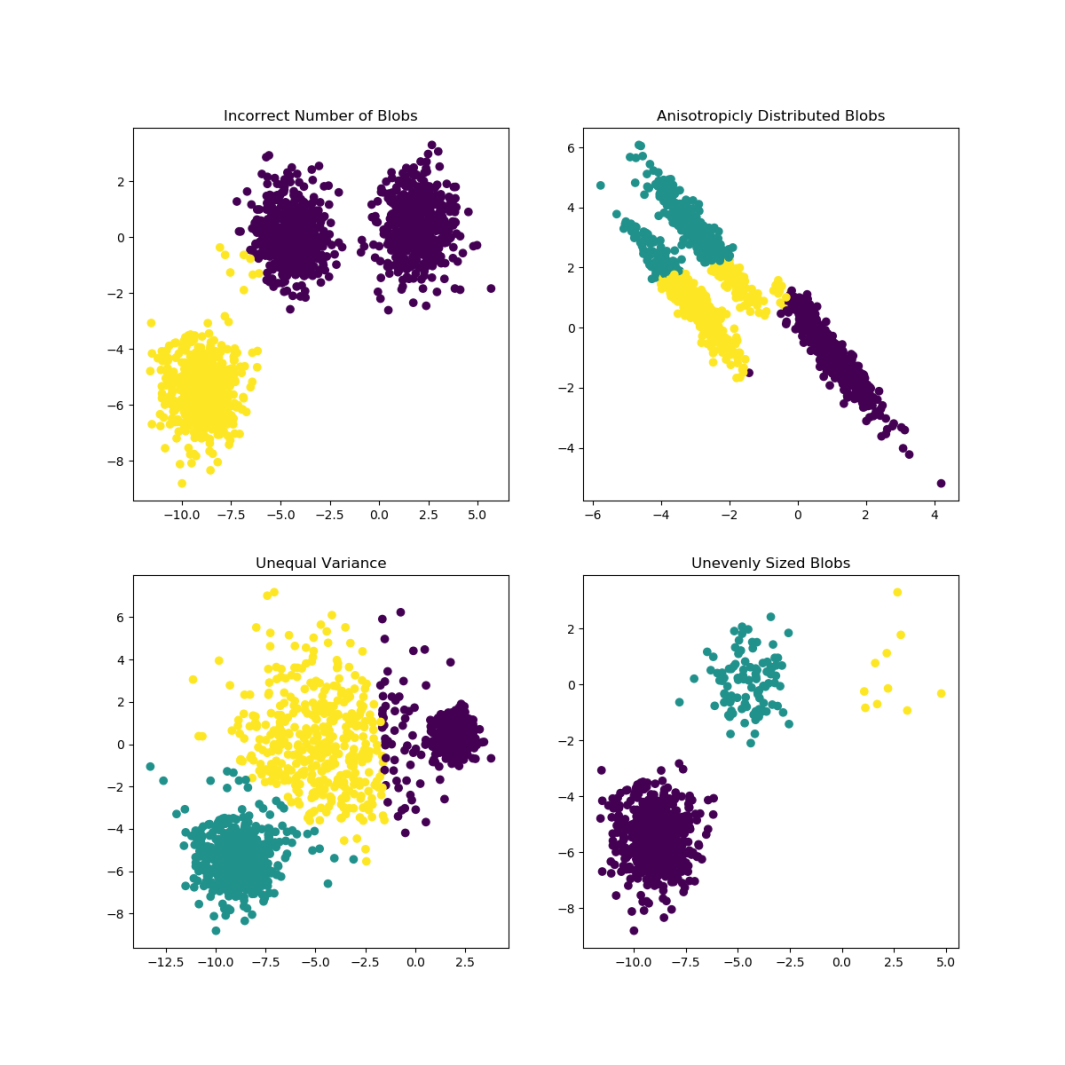


图8-2 均值聚类算法结果示例

## **标题2** 8.4 高斯混合模型

### **8.4.1** 高斯混合模型的概率分布

K均值聚类的缺点是它对于簇假设过于简单：每个样本只能属于一个簇，每个簇为一个球形高斯分布。高斯混合模型（Gaussian Mixed Models，GMM）是一种比K均值更灵活的聚类算法。

GMM中，我们假设每个簇的数据点服从高斯分布，用两个参数来描述簇：均值向量和协方差矩阵。由于有协方差参数，簇可以呈椭球形状，而不是被限制为球形。另外，我们不再要求每个样本只能属于一个簇，而是以一定的概率从属于每个簇。这样样本的从属度值，而不是只能取0或1两个值。

所以GMM模型对数据产生的假设为：假设有个簇，每一个簇服从高斯分布。首先以概率随机选择一个簇，并从该簇的分布中采样出一个样本点。所以GMM的概率密度函数为：

|  |  |
| --- | --- |
|  | （8-27） |

为每个样本点关联一个维的隐含变量，指示样本所属的簇（独热编码，所属簇的指示向量:表示样本属于第簇），其对应的随机向量用大写字母表示 （大写字母表示随机变量，小写字母表示随机变量的取值；大写表示随机事件发生的概率，小写的表示概率密度函数），则

如果已知的取值，如属于第簇，则：

所以

如果已知模型参数，根据贝叶斯公式，可计算样本从属簇的条件概率：

|  |  |
| --- | --- |
|  | （8-28） |

可以看做是对从属于第个簇的一种估计或者“解释”，可作为聚类结果。

模型参数估计可以采用极大似然法得到。我们写出对数似然函数为

|  |  |
| --- | --- |
|  | （8-29） |

其中为所有训练样本的集合。

原则上计算目标函数对各个参数的偏导数并置其等于0，可到使得似然函数最大。但目标函数中ln函数的输入是里有求和，所有参数耦合在一起，导致计算困难。

令目标函数对参数求偏导数并等于0，得到

得到：

|  |  |
| --- | --- |
|  | （8-30） |

类似的，可得到：

|  |  |
| --- | --- |
|  | （8-31） |

需要注意上面的结果并不是封闭解，因为依赖于参数。不过上述结论也提示了我们求解问题的方案，我们可以采用类似均值聚类的迭代求解过程：先初始化参数，计算；然后再根据上述结论，更新参数。（上述推导涉及向量矩阵求导，感兴趣的话可参考附件的矩阵求导推导）

事实上，上述迭代求解过程是期望最大化（Expectation Maximization, EM）的一个特例。

### **8.4.2** EM算法

期望最大算法是一种迭代算法，用于含有隐变量的概率参数模型的最大似然估计或极大后验概率估计。

假设完整的数据集由组成，但我们只能观测到。我们要求使得对数似然函数最大的参数向量。

如果我们知道完整数据，，可以很方便对计算完整数据的对数似然函数。然而隐含变量不可见，其信息只能从后验分布得到。通过最大化完整似然在该后验分布下的期望，来更新模型参数。

EM 算法分为两步：E步（Expectation 和 M步（Maximization）。

E步：通过观测到数据和当前模型参数，计算隐含变量后验分布；

M步：寻找完整似然函数的在隐含变量后验分布下的期望最大化对应的参数：

|  |  |
| --- | --- |
|  | （8-32） |

其中

|  |  |
| --- | --- |
|  |  |

下面我们讨论为什么通过这种启发式的方式推导出来的EM算法，但确实是在最大化似然函数。

定义函数

|  |  |
| --- | --- |
|  |  |

其中KL散度

|  |  |
| --- | --- |
|  |  |

我们采用坐标轴下降法求解：

1. 固定为，求使得 最大的 ；

由于，其中第一项与未知量无关，因此

因为当两个分布和相等时，KL散度取最小值0。

1. 固定为，求使得最大的 ；

由于，第二项与未知量无关，因此

其中。

可以证明，当最大时，也最大。

因为否则存在某个，使得 >，则 ，其中 ，这与假设最大相矛盾。后验分布未知，当变化时，取最大值对应的，也取最大值。

### **8.4.3** GMM的EM算法求解

我们将上述通用EM算法特化到GMM。

首先写出完整数据的似然函数为：

|  |  |
| --- | --- |
|  |  |

然后给定参数的当前估计值，计算隐含变量后验概率：

|  |  |
| --- | --- |
|  |  |

基于当前的隐含变量后验概率，计算完整数据似然在该分布下的期望：

|  |  |
| --- | --- |
|  |  |

再计算使得最大的参数。在函数中，参数与解耦，可以分开计算。

对参数，还有约束条件：，带约束的优化问题采用拉格朗日乘子法求极值。拉格朗日函数为

|  |  |
| --- | --- |
|  |  |

对拉格朗日函数分别对参数求偏导并设为0，得到

求解上述方程组，得到：

|  |  |
| --- | --- |
|  |  |

对参数，对目标函数分别对参数求偏导并设为0，得到

其中第3行到第4行利用链式求导和向量求导公式。

从而得到：

|  |  |
| --- | --- |
|  |  |

类似的，对参数，得到（目标函数分别对参数求偏导比较麻烦，需要运用矩阵微分，参考附件的矩阵求导）：

|  |  |
| --- | --- |
|  |  |

整理一下，得到**算法8-2。**

**算法**8-2**：GMM聚类**

1．初始化参数；

2．Repeat

2.1 E步：给定当前参数的估计值计算后验概率/从属度：

2.2 M步：基于当前从属度更新参数：

Until似然函数收敛或达到最大迭代次数。

GMM中，协方差矩阵的参数数目为，其中为特征的维度。对协方差矩阵施加限制，会使得模型从简单到复杂：

• 球形（对角线上元素值为1，其余元素为0）；

• 对角形（只有对角线上元素值非0）；

• 并列（所有簇的协方差矩阵相同）；

• 完全协方差。

模型越复杂，通常性能越好，但在小数据集上容易过拟合。在Scikit-learn中，可以通过设置类的参数实现。

GMM虽然比K均值聚类更加灵活，簇形状可以是椭球（如图8-3左图）；但GMM仍然假设簇的分布为高斯分布，所以当簇的形状不满足高斯分布时，聚类效果不佳（如图8-3右图）。后续我们将讨论能发现任意形状簇的聚类算法。

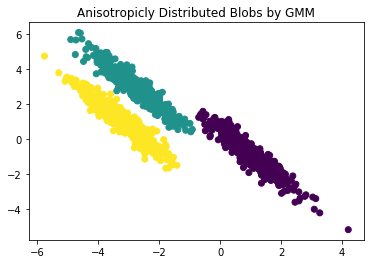
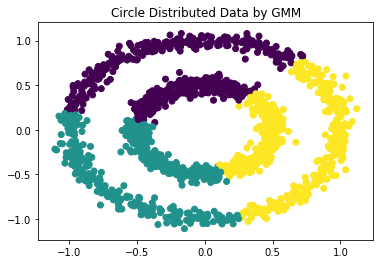
 

图8-3 GMM聚类算法结果示例

## **标题2** 8.5 层次聚类

层次聚类算法可分为两类：自上而下（分裂式）或自下而上（凝聚式）。自下而上的算法首先将每个数据点视为一个簇，然后连续聚合两个簇，直到所有的簇都聚合成一个包含所有数据点的簇。因此，自下而上层次聚类被称为凝聚式层次聚类。类似的，分裂式从一个包含所有样本的簇开始，然后不断分裂已有簇，直到每个簇只包含一个样本。层次聚类的结果可用用树状图表示，树的根结点是收集所有样本的大簇，叶子结点仅包含一个样本的簇。由于凝聚式层次聚类计算量更小，一般层次聚类采用自下而上方式进行，Scikit-learn中也只支持凝聚式层次聚类。

### **8.5.1** 凝聚式层次聚类

凝聚式层次聚类最开始每个样本视为一个簇，而后计算各簇之间的距离，将两个最相近的簇合并成一个新的簇。如此往复，最后就只剩下一个簇。

**算法**8-3**：凝聚式层次聚类**

1．初始化：每个样本为一个簇；

2．计算簇与簇之间的相似度，可存为相似度矩阵；

3．Repeat

3.1 合并最相似的两个簇；

3.2 更新簇之间的相似度矩阵；

Until只剩下一个簇。

层次聚类不需要指定簇的数量，甚至可以根据树状图选择一个合适簇的数量。与均值和GMM的线性复杂度不同，层次聚类的的效率较低，时间复杂度为。

虽然相对其他聚类算法而言，层次聚类对样本点之间距离度量标准的选择并不敏感，但簇之间的相似度/距离度量对层次聚类还是很重要。常用的簇之间的距离度量有：

• 最小距离(MIN/Single Linkage)：两个簇中距离最近的样本对之间的的距离

• 最大距离(MAX, Complete Linkage)：

• 平均距离(Group Average Linkage)：

• 中心点距离(Distance Between Centroids):

其中分别为簇的质心。

• 沃德（Wald）距离

Wald方法采用误差平方和（Sum of the Squared Error，SSE）采用衡量簇的质量。SSE计算公式如下：

Wald距离定义为如果合并两个簇，带来SSE的增加量：

我们可以看到在沃德方法中，既考虑了簇间的距离，同时也考虑了每个簇中的样本数目。所以当两簇间的距离相等时，沃德方法会选择簇更小的那组进行合并。不同簇距离度量的聚类结果如图8-4所示。

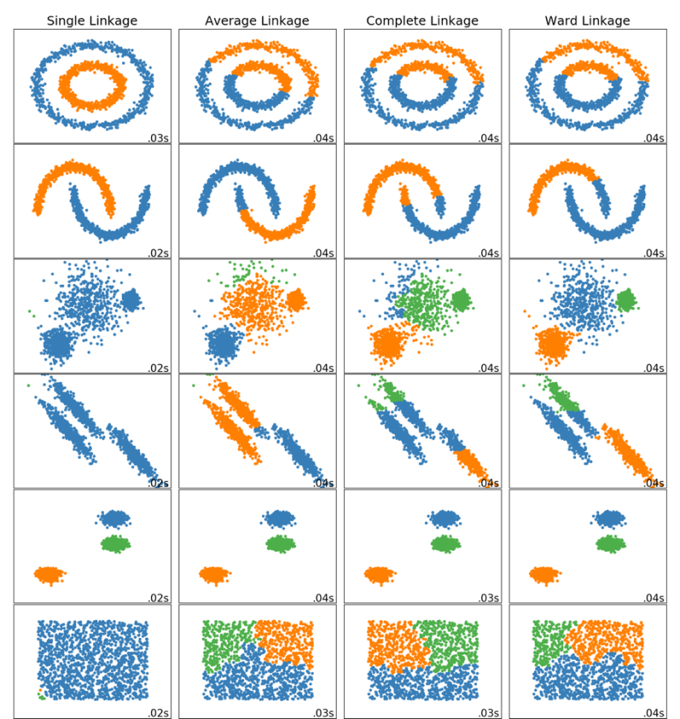


图8-4 凝聚式层次聚类算法结果示例

层次聚类算法较新的算法有：BRICH（Balanced Iterative Reducing and Clustering Using Hierarchies）适合数值特征大数据情况；ROCK (RObust Clustering using linKs)  用于对离散型特征的样本进行聚类；变色龙（Chameleon）算法根据近邻构造邻接图定义簇间相似性度量能自适应地合并簇。

### **8.5.2** 分裂式层次聚类

二分均值聚类算法是一种常用的分裂式层次聚类算法，是均值聚类算法的一个变体，主要是为了改进均值算法随机选择初始质心的随机性造成聚类结果不确定性的问题。

在二分均值聚类中，最开始时所有的样本都属于同一个簇，这个簇为树状图的根结点。接下来我们要挑选一个质量最不好的簇，将其分裂成两个新的簇。选择哪个簇进行二分的原则是能否使得尽可能小。同均值聚类算法类似，二分均值聚类算法也不适用于非球形簇的聚类，和不同尺寸和密度的类型的簇的聚类。

## **标题2** 8.6 **DBSCAN**

基于密度的噪声鲁棒的聚类（Density-Based Spatial Clustering of Applications with Noise，DBSCAN）是一种基于密度的聚类算法，可以处理不规则形状的簇，且对噪声数据的处理比较好。其核心思想是在数据空间中找到分散开的密集区域。

DBSCAN定义密度为给定半径圆圈内的样本数目，然后根据密度将样本分成不同类别（如图8-5所示）：

• 核心点（Core point）：指定半径内多于指定数量（）个点；

• 边界点（Border point）：半径内有少于个点，但在某个核心点的邻域内；

• 噪声点（Outliers）：核心点和边界点之外的点。

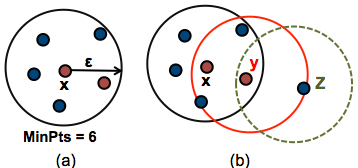


图8-5 DBSCAN中的基本概念。(b)中红色的点为核心点，  
圆圈中其他点为边界点，非圆圈中的点为噪声点。

在介绍DBSCAN的聚类过程之前，我们先做如下定义：

**密度可达**：如果连接两个点和点两个点的路径上所有的点都是核心点，则称点到点密度可达。如果是核心点，那么由它密度可达的点形成一个簇。

**密度相连**：如果存在点，从其密度可达点和点，则称点到点密度相连。

基于上述定义，簇满足以下两个性质:

1．连接性：簇内任意两点点是密度相连的；

2．最大性：如果一个点从一个簇中的任意一点密度可达，则该点属于该簇。

**算法**8-4**：**DBSCAN**聚类**

1．初始化核心点集合：；

2．#确定核心点

for

2.1 确定样本点的邻域内的样本集合

2.2 if ，

则将样本将加入核心点集合：

end if

end for

3．初始化簇的数目，未访问过的点集合；

4．while do #对所有核心点，确定和这个核心点相连的样本

4.1 记录当前未访问过的点的集合 ；

4.2 随机选择一个核心点，初始化队列；

4.3

4.4 while do

4.4.1 取出队列中的首个样本点；

4.4.2 if ，

为邻域中未访问过的点；

将中的点加入队列;

;

end if

end while

4.5 。

聚类结束后，不在任何簇内的点为噪声点（不是核心点，且不在任何核心点的邻域内），所以DBSCAN算法可以识别噪声，对噪声不敏感。DBSCAN算法也能很好地找到任意大小和任意形状的簇。

DBSCAN算法无需指定簇的数目，但需要设置两个邻域参数：。如果不变，取得值过大，会导致大多数点都聚到同一个簇中；过小，会导致一个簇的分裂。如果不变，的值取得过大，会导致更多样本点被标记为噪声点；过小，会导致发现大量的核心点。参数可以根据距离曲线和经验知识设置。

距离曲线的绘制过程如下：给定数据集，对于任意点，计算该点到集合中其他所有点的距离，这些距离按照从小到大排序，假设排序后的距离集合为，则就被称为距离，即点到其近邻的距离。对所有点的距离集合进行升序排序，得到距离集合，然后根据绘出距离变化曲线。曲线发生急剧变化的位置所对应的距离的值为半径的值（图8-6中），的值为（图8-6中）。

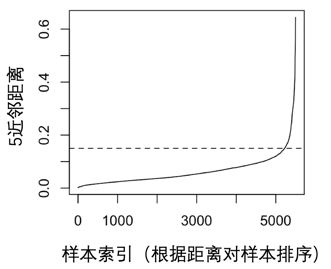


图8-6 距离曲线

由于这两个参数是全局的，当每个簇的密度不同时，DBSCAN 的表现不如其他聚类算法。因为当密度变化时，用于识别邻域点的距离阈值的设置将会随着簇而变化。这个缺点在非常高维度的数据尤为突出，因为距离阈值变得难以估计。DBSCAN的扩展OPTICS（Ordering Points To Identify Clustering Structure）通过优先对高密度进行搜索，然后根据高密度的特点设置参数，改进了DBSCAN。

## **标题2** 8.7 基于密度峰值的聚类

基于密度峰值的聚类算法假设簇中心周围都是局部密度低的点，并且与任何一个局部密度较高的点保持相对较远的距离。

对于每一个数据点，要计算两个量：点的局部密度、和该点到具有更高局部密度的点的距离，而这两个值都取决于数据点间的距离。

数据点的局部密度定义为：

|  |  |
| --- | --- |
|  | （8-40） |

其中为截断距离，

|  |  |
| --- | --- |
|  | （8-41） |

所以为到点的距离小于的点的数目。可以将所有点对相互距离从小到大排序，2%的位置距离数值设置为。

上述局部密度的定义是一种硬邻域定义（是邻域点或不是邻域点），我们也可以换成用高斯核定义的软邻域定义，根据距离来表示数据点与中心点的权重，离的越近，权重越高，离得越远，权重就越低：

|  |  |
| --- | --- |
|  | （8-42） |

点到高局部密度的点的距离定义为该点到其他有更高局部密度的点之间的最小距离：

|  |  |
| --- | --- |
|  | （8-43） |

所以那些具有较大距离 且同时具有较大局部密度的点定义为聚类中心。同时具有较高的距离但密度较小的数据点称为异常点。我们构造决策图和乘积曲线来寻找簇的数目和簇的中心。聚类中心确定之后，剩余点被分配给与其具有较高密度的最近邻居相同的簇。首先为每个簇定义一个边界区域，即划分给该簇但是距离其他簇的点的距离小于的点。然后为每个簇找到其边界区域的局部密度最大的点，令该最大局部密度为，则该簇中所有局部密度大于的点被认为是簇核心部分，其余点被认为是该簇的光晕（噪声点）。图8-7给出了示例数据集上的聚类结果。

|  |  |  |
| --- | --- | --- |
|  |  |  |
| （a）数据点的分布, 样本数目，不同簇用不同颜色表示，黑色点为背景噪声点，噪声点在20%； | （b）决策图，横轴为局部密度，纵轴为相对距离。取较大的5个样本点做为簇中心（图中彩色的大点）。 | （c）乘积曲线，纵轴为，横轴为，为对样本点降序排列后的索引。可以看出，在的地方陡降，即值最大的5个样点为簇中心。 |

图8-7 基于密度峰值的聚类

## **标题2** 8.8 谱聚类

谱聚类(Spectral Clustering)是一种基于图的聚类方法，其基本思想是将每个样本点看做无向图图中的一个结点，结点之间用边相连，边的权重与结点之间的相似度有关，聚类问题转化为图切分问题，即将图切分为多个子图，使得切分后的子图内部尽量相似，子图间距离尽量远。

### **8.8.1** 图论基础

**相似性图及其构造**

给定个数据点，将这些数据点视为带权无向图，结点集合，边的集合，其中结点表示数据点，两个结点和之间用边连接，其权重与数据对和之间的相似性有关，因此图也被称为相似性图。

一种常用的样本间的相似度为。边的构造根据样本间相似度可分为邻近法，近邻法和全连接法。

**-邻近法**

若样本间相似度大于 ，则用权重 连接两个样本；样本间的相似度小于，则连接两个样本的权重等于0。因此图的无向权重表达式为：

|  |  |
| --- | --- |
|  | （8-44） |

 近邻描述样本间的权重只有和0，丢失了很多信息 。

**近邻法**

近邻法只考虑离每个样本点的最近的个邻居，不在近邻范围的样本，权重为0。为了使相似度对称，我们假设样本点是样本点的近邻，且样本点也是样本点的近邻，则该样本间的权重不为0，数学表达式为：

|  |  |
| --- | --- |
|  | （8-45） |

其中表示样本点的近邻。

**全连接法**

全连接法直接用相似度衡量所有的样本间权重，即。

全连接法构建边和权重信息最全，通常效果最好。但其缺点是构建的图不稀疏，导致计算成本增加。

**邻接矩阵**

所有节点之间的权重值构成图的邻接矩阵，这是一个的对称矩阵，。

**度矩阵**

节点的度定义为与其相连的所有边的权重之和，即

|  |  |
| --- | --- |
|  | （8-46） |

所有节点的度构成的度矩阵**。**是对角矩阵，第行的主对角线元素值，为第个点的度数，即

|  |  |
| --- | --- |
|  | （8-47） |

**拉普拉斯(Laplace)矩阵**

拉普拉斯矩阵定义为

|  |  |
| --- | --- |
|  | （8-48） |

其中为度矩阵，为邻接矩阵。

**说明 拉普拉斯矩阵及其性质**

**拉普拉斯矩阵**

如果是欧式空间中的二阶可微实函数，那么是在欧氏空间中求其二阶微分（散度），其中是拉斯矩阵算子。拉普拉斯算子是维欧氏空间中的一个二阶微分算子定义为:

**一维空间**：

离散函数的一阶导数：；

离散函数的二阶导数：；

离散函数的二阶微分算子：，为左右邻居（，）与的函数值的差值和。

拓展到**二维空间**：

在二维空间中，某点的二阶微分算子（散度）为其上下左右邻居与该点处函数值的差值和。

我们现在将这个结论推广到**图**：

假设具有个节点的图，此时图中每个节点的自由度至多为，此时该图为完全图，即任意两个节点之间都有一条边连接。此时以上定义的函数不再是二维，而是维向量：  ，其中为函数在图中节点处的函数值。类比于在节点处的值。

 用 表示节点 的一阶邻域节点，拉普拉斯算子为一个点到其所有邻居的差异之和，考虑图中边的权值相等（简单说就是1）则有：

如果考虑边具有权重，则有：

其中为该点的度，表示邻接矩阵的第行。

对于所有的个节点有：

这里就是拉普拉斯矩阵 。

所以图拉普拉斯算子是作用在由图节点信息构成的向量上得到的结果，等于图拉普拉斯矩阵和向量的点积。拉普拉斯矩阵反映了当前节点对周围节点差异的累积，刻画图信号局部平滑度。

**拉普拉斯矩阵的性质**

拉普拉斯矩阵有一些很好的性质：

1. 拉普拉斯矩阵是对称矩阵。

由于为对称矩阵，为对角矩阵，也是对称矩阵，显然为对称矩阵。

2. 对于任意的向量，

|  |  |
| --- | --- |
|  | （8-49） |

证明：

3. 拉普拉斯矩阵是半正定的。

根据性质2，显然。

4. 拉普拉斯矩阵的最小特征值是0，且对应的特征向量为全1向量。

因为每一行的和均为0（根据定义），而对一个方阵求行和，相当于乘以一个维全1的向量， 所以。

我们将上述等式改写成。根据特征值和特征向量的定义，这个等式告诉我们：有一个特征值为0，且其对应的特征向量为（元素值全为1）。

5. 拉普拉斯矩阵有个实数特征值都大于等于0，即。

6. 拉普拉斯矩阵连通部件的数目：设是一个具有非负权的无向图，那么拉普拉斯矩阵多重0特征值的数目等于图连通部件的数目，特征值0的的特征空间由这些连通部件对应的指示指示向量张成。

我们从的情况开始，即图是全连通的，假设是特征值0的对应特征向量（），那么

由于权重是非负的，只有当所有项为0时，和才为0。由于图中任意两个节点是连接的（即），所以要求对任意的，相等，即在连通部件上是常数。当图是连通的情况下，特征值0的特征向量只有常数向量，这显然是连通分量的指示向量。

现在我们考虑有个连通部件的情况。不失一般性，我们假定结点是根据它们所属的连通部件排序的。在这种情况下，邻接矩阵具有块对角形式，并且矩阵也是块对角形式：

请注意，每个块本身都是一个合适的图拉普拉斯矩阵，即对应于第个连通子图的拉普拉斯矩阵。由于这对所有块对角矩阵都成立，则的谱由的谱的联合给出，相应的特征向量是的特征向量，其它块的位置用0填充。由于每个都是连通图的拉普拉斯矩阵，我们知道每个都有1个特征值0，对应的特征向量是第个连通分量上的常数向量。因此，矩阵的特征值0的数目与连通分量的数目相同，相应的特征向量为连通分量的指示向量。

**规范化的拉普拉斯矩阵**

文献中有两种矩阵称为规范化的图拉普拉斯矩阵，这两个矩阵彼此密切相关，定义为：

|  |  |
| --- | --- |
| 是对称矩阵，与随机游走密切相关。 | （8-50） |

和的性质如下：

1. 对于任意的向量，

|  |  |
| --- | --- |
|  | （8-51） |

2. 如果的特征值为，对应的特征向量为，则是的特征值，是对应的特征向量。

3.   是的特征值，对应的特征向量为**，**当且仅是广义特征值问题的解。

### **8.9.2** 图的切分

聚类的直觉是根据相似性将数据点分到不同簇中。对于以相似图形式给出的数据，数据集的聚类可以表述为：找到图的一个切分，使得不同簇之间的边的权重非常低（这意味着不同簇中的点彼此不同），而组内的边具有较高的权重（这意味着同一簇中的点彼此相似）。

**切图(cut)**

无向图的切分定义为将图切成互不连接的个子图，每个子图点的节点集合为，满足，且。

对任意两个子图节点的集合, 定义和之间的切分权重为

|  |  |
| --- | --- |
|  | （8-52） |

其中表示图的邻接矩阵的元素。

对个子图节点的集合，定义切图(cut)为

|  |  |
| --- | --- |
|  | （8-53） |

其中表示集合的补集。

最小化可得到一种图的切分，但这种切分标准只考虑了最小化簇间的相似度，并没有考虑簇内的相似度。例如将一个图切分为2个子集时，得到的切分结果是去掉权重最小的边，可能会导致其中一个簇的样本数为1。这显然不是我们所希望的，通常每个簇类包含的样本数应该比较多。

因此需要用每个子集的大小对上述最简单的切分方法进行规范化，得到规划化的切图(Normalized Cuts, NCut)，使得切出的几个子集大小尽量平均，对过大规模的簇进行惩罚。

**NCut**

定义子集的体积为图中与相连的边的权重，即

|  |  |
| --- | --- |
|  | （8-54） |

NCut最小化规划化的切图：

|  |  |
| --- | --- |
|  | （8-55） |

为了表示切分过程，我们引入一组指示向量 ，满足

|  |  |
| --- | --- |
|  |  |

即为一个长度为的向量，其中只有在第 个位置是 1，其余位置全是0， 表示样本  属于类。那么 的矩阵，表示将图切分成 个簇的一种方式（类似K均值聚类和GMM中的隐含向量/簇指示向量）。

最终我们将切图转化成一个优化问题：

|  |  |
| --- | --- |
|  | （8-56） |

**NCut的优化**

目标函数

|  |
| --- |
|  |

优化问题便转换为

|  |  |
| --- | --- |
|  | （8-57） |

其中，

。

而我们已知的只有 ，因此接下来尝试用它们表示矩阵。

对矩阵，

|  |  |
| --- | --- |
|  |  |

由于 为长度为 的向量，因此  的结果是  的矩阵。如果属于第簇，则只在第 个位置是 1，其余位置是0，因此 只在 处是1，其余位置是0。

因此，

|  |  |
| --- | --- |
|  |  |

即 也是  的矩阵，在 处为第簇包含数据点的数目是1，其余位置是0。我们发现，与矩阵极其相似，只是  与  的区别。显然：

|  |  |
| --- | --- |
|  | （8-58） |

其中为度矩阵。

对矩阵，，所以

|  |
| --- |
|  |

其中。

接下来我们讨论矩阵的表示。

可以看出，矩阵 为  对角线上的元素。由于要求的优化问题是几个对角矩阵相乘再求迹，那么如果用 代替 ，也不会影响最后的结果。

优化问题便转换为

|  |  |
| --- | --- |
|  | （8-59） |

这个优化目标是一个典型的瑞丽商（Rayleigh Quotient）的形式。

令，则，，且。

优化问题变成

|  |  |
| --- | --- |
|  | （8-60） |

我们要求的是迹（ trace），可以拆成 个子问题，让矩阵对角线上每个值都最小，即

|  |  |
| --- | --- |
|  | （8-61） |

其中 ，所以分母去掉，变成带约束的优化问题

|  |  |
| --- | --- |
|  | （8-62） |

利用拉格朗日乘子法，得到

|  |  |
| --- | --- |
|  |  |
|  |  |

从而，

|  |  |
| --- | --- |
|  |  |

即对规范化的拉普拉斯矩阵进行特征值分解。使目标函数最小的为最小特征值对应的特征向量，是将图切一刀的结果。矩阵的特征值和特征向量计算相当于将矩阵分解为一系列谱的组合，因此基于规范化切图的聚类亦被称为谱聚类。

因此我们得到基于规范化切图谱聚类的过程如下。

**算法**8-**5：规范化切图谱聚类**

输入：相似度矩阵、 簇的数目

1. 根据相似度矩阵构建邻接矩阵和度矩阵（如8.9.1节所述）；

2. 计算规范化的拉普拉斯矩阵；

3. 计算矩阵的前个最小特征值对应的特征向量：；

4. 将作为矩阵的每列，构造矩阵；

5. 并将每行规范化（模长为1），得到矩阵：；

6. 对中的每一行作为一个样本，共个样本作为输入进行K均值聚类。

输出 ：簇划分 .

上述算法使用是规范化的拉普拉斯矩阵。由前面的和之间的关系，亦可对矩阵进行特征分解实现。事实上，采用计算更方便，因为的特征向量就是聚类指示向量，而的特征向量还需额外乘以，这可能会导致意外的计算误差。由于使用没有任何计算优势，因此我们提倡使用。

这样算法8-6的第2步改为计算，同时无需第5步对的规范化，即。

上述算法也相当于先将每个样本的原始表示变成低维表示，再基于的欧氏距离采用K均值聚类，将表示变成表示会使得聚类更容易。因此谱聚类可视为将样本转换到谱空间再聚类。

**说明 谱聚类与随机游走**

谱聚类的另一个解释是基于相似图上的随机游动。图上的随机游动是从一个结点到另一个结点随机跳跃的随机过程。谱聚类可以解释为试图找到图的一个划分，使得随机游动很长时间保持在同一个簇，很少在簇之间跳跃。从直觉上看，图切小的划分意味着随机游动不太可能在簇之间跳跃。形式化表示一个一个结点到另一个结点跳跃的转移概率与边的权重成成正比，即。

因此，随机游动的转移矩阵定义如下：

|  |  |
| --- | --- |
|  | （8-63） |

如果图是连通且非二部的，那么随机游动总是具有唯一的平稳分布，其中。

显然，和之间存在紧密的关系：。因此是的特征值，且特征向量为，当且仅当是具有特征向量的特征向量。众所周知，图的许多性质可以用矩阵表示。因此，从随机游走的观点来看，最大的特征向量和最小的特征向量可以用来描述图的聚类属性。可以证明，Ncut和随机游动转移概率之间的形式上是等价的：。

在scikit-learn中，sklearn.cluster.SpectralClustering实现了基于Ncut的谱聚类。其中相似矩阵的建立只实现了基于K邻近法和全连接法的方式，没有基于ϵ-邻近法。最后一步的聚类方法提供了K-Means算法和 discretize算法两种算法。对于SpectralClustering的参数，需要调参的主要是邻接矩阵建立相关的参数和聚类类别数目。

## **标题2** 8.9 基于深度学习的聚类

基于深度学习的聚类将深度网络的学习和聚类结合起来，同时学习网络参数和对网络输出的特征进行聚类。我们亦可将深度网络部分视为从原始数据空间提取特征，然后在该特征空间中优化聚类。基于深度学习的聚类包含3部分：深度网络、网络学习目标函数和聚类目标函数。其中深度网络用于学习数据的低维非线性表示，自编码器、CNN、变分自编码器和GAN等模型均可用于聚类。深度聚类的目标函数通常是网络学习目标函数和聚类目标函数的线性组合：

|  |  |
| --- | --- |
|  | （8-63） |

其中表示聚类损失，表示将样本归到某簇的代价，如K均值聚类中的SSE；表示深度模型的学习的目标函数，如自编码器的重构误差和稀疏约束，超参数控制二者之间的折中。有些模型在深度网络学习中再加入局部保持约束。一种典型的模型如图8-8所示。

网络目标函数对于深度神经网络的初始化至关重要。 通常在训练几轮之后，通过改变参数的值引入聚类损失。也有一些模型完全放弃网络学习目标函数，而只使用聚类目标函数来指导表示网络学习和聚类，此时需要仔细设计聚类目标函数，同时网络参数的初始化也很重要。

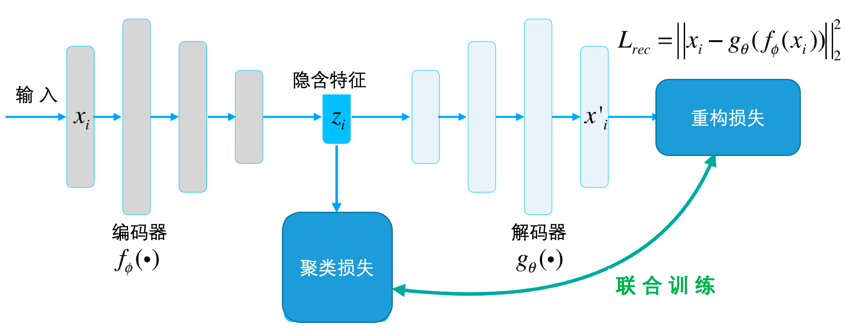


图8-8 基于深度学习的聚类

## **标题2** 8.10 聚类案例分析—**MNIST**数据集聚类

我们在MNIST数据集进行聚类练习。MNIST数据集是一个手写数字图像数据集，数据集介绍请见5.3.3节。数据集本身带有标签，方便我们比较各种聚类技术。我们从训练集中随机选择了20%（共8400个样本）参与聚类试验。

我们在Scikit-Learn框架下实现了均值聚类、GMM、均值漂移、DBSCAN。由于数字的图像中大部分为黑色的背景，且偏白色的数字笔划大多连续且灰度值相似，所以MNIST数据集中，样本的各个维度之间的相关性很强，因此我们首先采用PCA降维，保留85%方差，得到主成分数目为59。后续聚类我们用手写数字图像PCA降维后的特征表示。这里我们用每个簇中出现次数最多的数字作为该簇样本的预测标签，计算预测的正确率。各聚类算法的性能如表8-1所示。

对K均值聚类，我们采用“肘部法”选择最佳的K值。不同K对应的SSE如图8-9（a）所示。不过图中没有明显的肘部位置，相对而言稍好些。GMM中也取。对层次聚类，无需指定簇的数目，不过簇的数目也设为100。

DBSCAN算法的超参数为邻域大小eps和邻域内最小样本数。超参数可以通过观察距离曲线确定，如图8-9（b）所示。不同对应的距离曲线形状大致相同，图中没有看出距离显著增大的地方，因此不好确定DBSCAN的超参数eps，这也意味着DBSCAN聚类效果不会太好。在750-800之间近邻距离有较大的变化，因此取，，对应的聚类效果也非常差，其他对应的聚类效果也很差。再适当减少，以发现更多核心点。 的结果如表8-1所示。需要注意的是虽然预测正确率还可以，但还有6851个样本点没有参加聚类（被认为是噪声）。

表**11-1** 聚类算法在**MNIST**数据集上的结果

|  |  |  |
| --- | --- | --- |
| 聚类算法 | 簇的数目 | 预测正确率 |
| 均值聚类 | 100 | 0.832619 |
| GMM | 100 | 0.848929 |
| 层次聚类 | 100 | 0.845357 |
| DBSCAN | 24 | 0.728212 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| （a）K均值 | （b）层次聚类 | （c）DBSCAN |

图8-9 MNIST数据集上聚类算法超参数确定

## **标题2** 8.10 聚类算法小结

本章介绍了一些常用的聚类算法，其中K均值聚类及其改进算法属于基于划分的聚类算法，K均值聚类简单快速被经常使用，虽然在很多情况下性能并不是很好。层次聚类我们讨论了凝聚和分裂两种形式。DBSCAN和基于密度峰值的聚类都属于基于密度的聚类算法。常用的聚类算法还有基于网格的聚类算法STING、基于图的聚类算法（谱聚类）等。自组织网络（Self-Organized Maps，SMO）亦可实现聚类，深度学习也可以用于聚类任务。

我们在选择聚类算法需要的考虑的因素包括：数据规模和维度、簇的形状、数据的噪声水平，以及是否需要预先知道簇的数目等。

**附：矩阵求导**

一元微积分中，导数与微分之间的关系：。

多元微积分中，梯度与微分之间的关系：

矩阵导数与微分也可建立联系：

其代表迹（trace），是方阵对角线元素之和。

**矩阵微分的运算法则**

加减法：= ;

矩阵乘法：=;

转置：;

迹：;

逆：。

行列式：，其中表示的伴随矩阵，在可逆时又可以写作。

通过矩阵导数与微分的联系，在求出左侧的微分后，可以利用如下一些迹技巧（trace trick）写成右侧的形式并得到导数。

**矩阵求导的运算法则**

当为实对称矩阵时， ， ；

。

**多元正态分布参数的极大似然估计**

多元正态分布的概率密度函数为：

给定样本集合，参数的对数似然函数为：

对参数**，**

从而得到：

|  |  |
| --- | --- |
|  |  |

对参数：

为实对称矩阵时，根据矩阵求导规则， ，上式中第一项；

第一项：；

第二项：

​

再套上迹，

其中第一个等号先交换了与，第二个等号将右边式子交换到左边，第三个等号再一次交换与。

合并第一项和第二项，得到

对照导数与微分的联系有：

令得到：

|  |  |
| --- | --- |
|  |  |