聚类模型与算法

张翼鹏

November 16, 2019

目录

0 聚类概述 **3**

0.1 聚类的定义 3

0.2 聚类的类别 3

1 划分聚类 **3**

1.1 ***K-Means*** 3

1.1.1 原始***K-Means*** 3

1.1.2 ***K-Means++*** 4

1.1.3 ***K-Median*** 4

1.1.4 ***K-Mediod*** 4

1.2 ***LVQ***学习向量量化 4

2 层次聚类 **6**

2.1 凝聚型层次聚类 6

3 密度聚类 **6**

3.1 ***DBSCAN*** 6

4 图论聚类 **7**

4.1 谱聚类 7

4.1.1 代数步骤 7
4.1.2 几何解释 8
4.1.3 补充 9

5 聚类模型评价 **11**

5.1 肘部法则 11
5.2 CH指标 11
5.3 轮廓系数 11
5.4 兰德系数 12

0 聚类概述

0.1 聚类的定义

将物理或抽象对象的集合分成由类似的对象组成的多个类的过程被称为聚类。按照某一个特定的标准（比如距离），把一个数据集分割成不同的类或簇，使得同一个簇内的数据对象的相似性尽可能大，同时不再同一个簇内的数据对象的差异性也尽可能的大。

聚类与分类的不同在于，聚类所要求划分的类是未知的。聚类是无监督学习，分类是监督学习。

0.2 聚类的类别

- 划分聚类：$K\text{-}Means$、LVG
- 层次聚类：凝聚型层次聚类、$BIRCH$
- 密度聚类：$DBSCAN$
- 图论聚类：谱聚类

1 划分聚类

1.1 $\boldsymbol{K\text{-}Means}$

1.1.1 原始$\boldsymbol{K\text{-}Means}$

思想：以k为参数，把n个对象分成k个簇，使簇内具有较高的相似度，而簇间的相似度较低。

步骤：1、随机选择k个点作为初始的聚类中心。

2、对于剩下的点，根据其与聚类中心的距离，将其归入最近的簇。

3、对每个簇，计算所有点的均值作为新的聚类中心。

4、重复2、3直到聚类中心不再发生改变。

优点：易于理解和实现，计算速度快；反映了簇内样本围绕质心的紧密程度。

缺点：很难预测到准确的簇的数目（k值难以确定）；该算法主要发现圆形或者球形簇，对其他形状和密度的簇效果不好；对噪声非常敏感。

1.1.2 K-$Means$++

介绍：k个初始化的质心的位置对最后的聚类结果和运行时间都有很大的影响，若仅仅是完全随机选择，可能导致算法收敛很慢。K-$Means$++算法就是对K-$Means$随机初始化质心的方法的优化。

步骤：1、随机选择一个点作为第一个聚类中心。

2、对每个样本点，计算它与已选择的聚类中心中最近聚类中心的距离。

3、选择距离最大的样本点作为新的聚类中心。

4、重复2和3直到选择出k个聚类质心。

1.1.3 K-$Median$

介绍：K-$Median$算法是K-$Means$算法的改进，对噪声不敏感，鲁棒性更好。

区别：1、距离的计算方式：K-$Median$ 算法采用曼哈顿距离（即$L1$-范数）。

2、中心点的选取方式：K-$Median$算法使用中位数来选取中心点（取数据点每个分量的中位数），而不是平均值。

1.1.4 K-$Mediod$

介绍：K-$Mediod$算法是K-$Means$算法的改进。K-$Means$的聚类中心由簇内所有点的均值算出，很容易被异常值带偏；但K-$Mediod$的聚类中心一定是原有数据点，所以对噪声相对不敏感，鲁棒性更好。

区别：将K-$Means$算法的步骤3改为“在每一个簇中，计算每个点到其他点的距离的平方和，将平方和最小的点作为新的聚类中心”。

1.2 LVQ学习向量量化

思想：找到一组原型向量来代表聚类的中心，每个聚类中心定义了一个区域，区域内的样本点与本区域聚类中心的距离不大于与其他区域聚类中心的距离。LVG假设数据样本带有类别标签，学习过程中利用这些样本的监督信息来辅助聚类。

步骤：1、我们拥有带标签的数据集D、类别个数k以及学习率参数$\eta \in (0,1)$，最终要得到k个原型向量$q_1, q_2, \ldots, q_k$。

2、初始化原型向量，可以选择某个标签为i的样本作为第i个原型向量的初始值。

3、任取一个样本x_j，找到与此样本距离最近的原型向量，假设为q_i。

4、更新q_i：

$$q_i' = q_i + \eta(x_j - q_i) \quad 若y_j = i$$

$$q_i' = q_i - \eta(x_j - q_i) \quad 若y_j \neq i$$

即若x_j与最近的原型向量q_i具有相同的类别标记，则令q_i向x_j的方向靠拢，且

$$dist(q_i', x_j) = (1 - \eta) \cdot dist(q_i, x_j)$$

否则令q_i远离x_j，且

$$dist(q_i', x_j) = (1 + \eta) \cdot dist(q_i, x_j)$$

5、判断是否达到最大迭代次数或原型向量更新幅度小于某个阈值。若是，则停止迭代，输出原型向量；否则，转至步骤3。

6、得到原型向量后，即可实现对数据集D的$Voronoi$（一组由连接两邻点线段的垂直平分线组成的连续多边形）划分，每个原型向量对应着一片区域，此区域内的样本点就隶属于此原型向量所代表的聚类簇。

优点：结构简单，只通过内部单元的相互作用就能完成复杂的聚类处理。

缺点：原型向量的初始化可能影响聚类效果。

LVQ作为监督学习算法，既然我们已经明确了簇的数量k以及每个样本的标签值，那此处它为什么属于聚类算法而不是分类算法呢。

我认为原因在于，在分类中，我们不仅知道类别的数目，还知道每种类型的具体含义；但在聚类中，我们并不知道每个簇所代表的含义，即使知道聚类数目，赋予每个簇一个标签（如$1, 2, \cdots, k$），这也只是一个代号，这个值本身对我们没有任何帮助。

换句话说，我们建立分类模型的目的是为了进行预测，获得新的样本的标签值，这个标签是我们最终需要的；但在聚类中，标签值本身不带任何信息，我们得到的信息，是样本之间或紧密或疏远的联系。

所以在LVQ中，这个标签值只是一个我们自己定义的伪标签，它依然属于聚类。在$K\text{-}Means$和LVQ模型中中，我们最终都能得到一个“分类器”，可以对新样本进行预测，但我们只能预测它们落到哪个类，与哪些样本关系紧密，但我们不明白这个类代表什么，不明白样本落入这个类后意味着它自身可能具有怎样的属性。

按照这种理论，如果我们面对一个真正的分类问题，是否可以使用LVQ算法呢。我认为可以，那么我们最终将得到一个真正意义上的分类器。同时我认为这就是互联网中同时包含LVQ聚类和LVQ分类的资料的原因。

但是，LVQ与其他聚类方法相比确实有一个很大的不同，即大部分聚类模型是没有训练这个过程的，最初拿到的样本即是我们进行聚类行为的目标样本；但LVQ作为监督学习，需要根据已知标签的样本计算出聚类中心，然后作为”分类器“进行使用。

按照这种理论，监督学习和无监督学习并不是判断分类和聚类的标准，关键还要看我们的掌握的信息以及最终的目的。

2 层次聚类

2.1 凝聚型层次聚类

思想：通过计算不同类别数据点间的距离来创建一棵有层次的嵌套聚类树。

步骤：1、计算所有数据点两两之间的欧氏距离，组成距离矩阵。

2、将距离最近的点合并为一类。

3、重复1、2，直到所有数据点合并为一类。计算数据点与数据点组合或两个数据点组合之间的距离时一般采用两两数据点计算距离然后取均值的方式。

优点：不需要预先制定聚类数；可以得到任意形状的簇；可以在不同的尺度（层次）上展示数据集的聚类情况。

缺点：计算量较大，不适合大样本聚类；对噪声较敏感。

3 密度聚类

3.1 $DBSCAN$

思想：将具有足够密度的区域划分为簇，并在具有噪声的空间数据库中发现任意形状的簇。它将簇定义为密度相连的点的最大集合。

步骤：1、确定两个参数：半径Eps和邻域内最少点数$MinPts$

2、将所有数据点划分为三类：

- 核心点：在半径Eps内含有不少于$MinPts$数目的点。
- 边界点：不是核心点，但是落在核心点的邻域内的点。
- 噪声点：既不是核心点也不是边界点的点。

3、将所有点标记为核心点、边界点或噪声点，删除噪声点。

4、为距离在Eps之内的所有核心点两两之间赋予一条边，每组连通的核心点形

成一个簇。

5、将每个边界点指派到其邻域内某一个核心点的簇中。

优点：不需要预先制定聚类数；可以得到任意形状的簇；可以识别噪声点。

缺点：对于簇密度变化较大的情况，聚类效果不好。

4 图论聚类

4.1 谱聚类

4.1.1 代数步骤

思想：矩阵的全部特征值的集合称为谱，谱聚类通过计算样本数据的拉普拉斯矩阵的特征向量，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K-$Means$）进行聚类。

步骤：1、给定n个样本$X = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_n}\}$和簇数$k$。

2、计算相似度矩阵（邻接矩阵）W，一般使用高斯核函数：

$$w_{ij} = \exp(\frac{-||\boldsymbol{x_i} - \boldsymbol{x_j}||^2}{2\sigma^2}) \quad i, j = 1, 2, \cdots, n$$

其中σ控制着样本点的邻域宽度，σ越大表示距离较远的样本点之间相似度越大。

3、计算度矩阵D：

$$d_i = \sum_{j=1}^{n} w_{ij} \quad i = 1, 2, \cdots, n$$

d_i即相似度矩阵W每一行元素之和，D为d_i组成的对角矩阵。

4、计算拉普拉斯矩阵$L = D - W$。

5、计算L的特征值，将特征值从小到大排序，计算前k个特征值的特征向量$u_1, u_2, \cdots, u_k$，将其组成矩阵U。

6、计U的第i个行向量为y_i，$i = 1, 2, \cdots, n$。

7、用K-$Means$算法将新样本$Y = \{y_1, y_2, \cdots, y_n\}$聚类成$k$ 簇。

优点：(1) 使用了降维技术，适用于高维数据的聚类。

(2) 只需要数据之间的相似度矩阵，对于处理稀疏数据的聚类（使计算量减小）或数据维度不一样的聚类（如时间序列聚类）很有效。

(3) 建立在图论基础上，能在任意形状的样本空间上聚类且收敛于全局最优解。

（K-$Means$采用欧式距离作为分类依据，只能处理线性可分的聚类问题；谱聚类采用核映射方法，定义样本间相似度时用内积来代替欧式距离，能够处理非线性可分的聚类问题。）

缺点：对k值和构造相似矩阵的尺度参数非常敏感；只适用于各簇之间点的个数相差不大的均衡聚类问题；谱聚类的松弛条件只是原问题的一个近似，但不能保证该近似是最合适的。

4.1.2 几何解释

所有样本点之间两两相连，组成一个带权无向图，权重由样本点之间的距离算出，表示样本点之间的相似度。谱聚类的目的是找到一种合理的分割图的方法，使得分割后的若干个子图中，连接不同子图的边的权重尽可能低，同子图内的边的权重尽可能高。

即我们要让被切断的边的权重之和尽量小。以$k=2$举例，设$\{A,\overline{A}\}$为图的两个互不相交的子集，目标函数如下：

$$\mathrm{cut}(A,\overline{A})=\mathrm{cut}(\overline{A},A)=\sum_{i\in A,j\in\overline{A}}w_{ij}$$

即分割时去掉的边的权值的总和。

上式被称为图的最小切，但在实际应用中往往效果并不好，很可能得到的结果是将图分为一个点和其他$n-1$个点。为了让每个类都有合理且均衡的大小，使用改进后的目标函数：

$$\mathrm{RatioCut}(A,\overline{A})=\frac{\mathrm{cut}(A,\overline{A})}{|A|}+\frac{\mathrm{cut}(A,\overline{A})}{|\overline{A}|}$$

其中$|A|$表示集合A中包含的顶点数目。

为了最小化目标函数，我们要用到拉普拉斯矩阵一个性质：对于任意n维实向量f，有

$$\begin{aligned}f'Lf&=f'Df-f'Wf\\&=\sum_{i=1}^{n}d_if_i^2-\sum_{i,j=1}^{n}f_if_jw_{ij}\\&=\frac{1}{2}(\sum_{i=1}^{n}d_if_i^2-2\sum_{i,j=1}^{n}f_if_jw_{ij}+\sum_{j=1}^{n}d_jf_j^2)\\&=\frac{1}{2}\sum_{i,j=1}^{n}w_{ij}(f_i-f_j)^2\end{aligned}$$

此处以$k=2$为例，定义n维实向量f：

$$f_i=\begin{cases}\sqrt{|\overline{A}|/|A|} & \text{if } v_i\in A\\ -\sqrt{|A|/|\overline{A}|} & \text{if } v_i\in\overline{A}\end{cases}\quad i=1,2,\cdots,n$$

我们将推出一个有趣的结论:

$$
\begin{aligned}
f'Lf &= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \\
&= \frac{1}{2}\left(\sum_{i\in A, j\in \overline{A}} w_{ij}\left(\sqrt{\frac{|\overline{A}|}{|A|}} + \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 + \sum_{i\in \overline{A}, j\in A} w_{ij}\left(-\sqrt{\frac{|\overline{A}|}{|A|}} - \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2\right) \\
&= \mathrm{cut}(A,\overline{A})\left(\frac{|\overline{A}|}{|A|} + \frac{|A|}{|\overline{A}|} + 2\right) \\
&= \mathrm{cut}(A,\overline{A})\left(\frac{|A|+|\overline{A}|}{|A|} + \frac{|A|+|\overline{A}|}{|\overline{A}|}\right) \\
&= |V| \cdot \mathrm{RatioCut}(A,\overline{A})
\end{aligned}
$$

所以最小化RatioCut，等价于最小化$f'Lf$。此时优化问题变为：

$$
\begin{cases}
\min\limits_{f\in R^n} f'Lf \\
s.t.\ f\text{满足上述定义}
\end{cases}
$$

只要我们解出了f，也就知道了每个样本点的归属。但由于f的取值是离散的，这个优化问题是一个NP难的问题难以求解，于是我们放宽标准，让f 的取值可以连续。为了尽可能与原问题保持一致，注意到$\sum\limits_{i=1}^{n} f_i = 0$，$\sum\limits_{i=1}^{n} {f_i}^2 = n$，所以我们的优化问题变为：

$$
\begin{cases}
\min\limits_{f\in R^n} f'Lf \\
s.t.\ f\perp\vec{1},\ ||f_i||^2 = n
\end{cases}
$$

此时，利用Rayleigh quotient：

$$
R(A,x) = \frac{x'Ax}{x'x}
$$

$R(A,x)$的最小值等于A的最小特征值，当且仅当x为最小特征值对应的特征向量时取到。我们代入f和L，问题则变成了求解L的最小特征值（但L的最小特征值为0，对应特征向量为$\vec{1}$，不满足$f\perp\vec{1}$条件，所以改取L的第二最小特征值，但也正因为此，L的其他特征向量都满足与$\vec{1}$正交的条件）。推广到$k > 2$ 的情况时，即为求解L的前k个最小特征值对应的特征向量。

最后，由于我们上一步松驰了NP难问题，让f的取值范围连续，因此解出来的f不再具有原来的性质——元素值能指出哪个点属于哪一类。因此，我们使用$K\text{-}Means$对L的前k 个最小特征值组成的矩阵的n个行向量进行聚类作为谱聚类的最终结果。

4.1.3 补充

对于$k > 2$的情况有一个更严格的解释：

首先我们已知

$$\mathrm{RatioCut}(A_1, A_2, \cdots, A_k) = \sum_{p=1}^{k} \frac{\mathrm{cut}(A_p, \overline{A_p})}{|A_p|}$$

对图的分割$\{A_1, A_2, \cdots A_k\}$引入一个$n * k$维的指示矩阵$H = \{h_1, h_2, \cdots, h_k\}$，图的每个子集$A_p$对应一个指示向量$h_p$（$p = 1, 2, \cdots, k$）。定义$H$中的元素：

$$h_{ip} = \begin{cases} \dfrac{1}{\sqrt{|A_p|}} & \text{if } v_i \in A_p \\ 0 & \text{if } v_i \in \overline{A}_p \end{cases} \qquad i = 1, 2, \cdots, n \quad p = 1, 2, \cdots, k$$

可以看出H中的列向量均为单位向量且两两正交（直观上也很好理解，H中每一个列向量h_p暗含的信息是每个数据点是否属于第p个类，一个数据点不可能同时属于两类，所以H的列向量必然两两正交），即$H^{\mathrm{T}}H{=}I$。

根据拉普拉斯矩阵的性质，有：

$$\begin{aligned} h_p' L h_p &= \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (h_{ip} - h_{jp})^2 \\ &= \frac{1}{2} \left(\sum_{i \in A_p, j \in \overline{A_p}} w_{ij} \left(\frac{1}{\sqrt{|A_p|}} - 0\right)^2 + \sum_{i \in \overline{A_p}, j \in A_p} w_{ij} \left(0 - \frac{1}{\sqrt{|A_p|}}\right)^2 \right) \\ &= \frac{1}{2} \left(\sum_{i \in A_p, j \in \overline{A_p}} \frac{w_{ij}}{|A_p|} + \sum_{i \in \overline{A_p}, j \in A_p} \frac{w_{ij}}{|A_p|} \right) \\ &= \frac{1}{2} \left(\frac{\mathrm{cut}(A_p, \overline{A_p})}{|A_p|} + \frac{\mathrm{cut}(\overline{A_p}, A_p)}{|A_p|} \right) \\ &= \frac{\mathrm{cut}(A_p, \overline{A_p})}{|A_p|} \quad p = 1, 2, \cdots, k \end{aligned}$$

所以

$$\mathrm{RatioCut}(A_1, A_2, \cdots, A_k) = \sum_{p=1}^{k} \frac{\mathrm{cut}(A_p, \overline{A_p})}{|A_p|} = \sum_{p=1}^{k} h_p' L h_p = tr(H^{\mathrm{T}} L H)$$

再对H进行松弛化处理，最终优化问题变为：

$$\begin{cases} \min\limits_{H \in R^{n*k}} tr(H^{\mathrm{T}} L H) \\ s.t.\ H^{\mathrm{T}} H{=}I \end{cases}$$

（这里为了表述方便，写的是优化$tr(H^{\mathrm{T}}LH)$，但看成优化$\sum\limits_{p=1}^{k} h_p' L h_p$更便于理解。）

为了最小化$\sum\limits_{p=1}^{k} h_p' L h_p$，我们让它的每一项都尽量小。根据瑞利熵，取$L$的前$k$个最小特征值，将其对应的特征向量排列起来即得到最优H。

由于我们在使用维度规约时损失了少量信息，导致得到的指示矩阵H不能直观指示各样本的归属（但其中仍含有大量信息），因此还需要对H的每一行（最好按行做一次标准化，转化为单位向量）进行一次传统的聚类，比如$K\text{-}Means$聚类。

5 聚类模型评价

5.1 肘部法则

每个样本点到与其所在簇内质心的距离平方和称为簇内误差平方和（SSE）。它会随着类别的增加而降低，但对于有一定区分度的数据，在类别数达到某个临界点时SSE会得到极大改善，之后缓慢下降，这个临界点就可以被认为是聚类性能较好的点。

5.2 CH指标

CH指标通过计算分离度与紧密度的比值来衡量聚类好坏。CH值越大，聚类效果越好。

$$CH(k) = \frac{traceB/(k-1)}{traceW/(n-k)}$$

$$traceB = \sum_{i=1}^{k} n_i \times dist^2(z_i, z)\text{为类间离差阵的迹}$$

$$traceW = \sum_{i=1}^{k} \sum_{x \in C_i} dist^2(x, z_i)\text{为类内离差阵的迹}$$

$$z = \frac{1}{n}\sum_{i=1}^{n} x_i\text{为所有样本的均值} \qquad z_i = \frac{1}{n_i}\sum_{x \in C_i} x\text{为第i类中样本的均值}$$

k为簇数，n为样本量，n_i为第i类中的样本量。

5.3 轮廓系数

轮廓系数定义如下，对单个数据点来说：

$$s = \frac{b-a}{\max(a,b)}$$

其中a为该数据点距自身簇内其他所有数据点的距离的均值；b为该数据点距其他最近的一个簇内的所有数据点的距离的均值。所有数据点的轮廓系数的均值称为该聚类模型的整体轮廓系数。轮廓系数取值为$[-1,1]$，越接近1代表聚类效果越好。

5.4 兰德系数

给定真实类别信息U和聚类结果V，兰德系数定义如下：

$$RI = \frac{a+b}{C_n^2}$$

其中n为数据点总数；a为在U和V中都为同一类的数据点对数；b为在U和V中都不在同一类的数据点对数。RI取值范围为$[0,1]$，值越大意味着聚类结果与真实情况越吻合。