第一章　聚类分析

1.1　简 介

1.1 简 介

- 当数据集有明确的标签时, 可利用标签信息进行监督学习, 发现数据内在影响关系, 然而, 在一些实际问题中, 标签数据有时较难获取或者无法获取,此时, 通常使用无监督学习对数据进行贴签, 聚类分析是无监督学习的典型代表.
- 所谓聚类是指根据某种规则, 将样本集中具有相似特征的样本进行分组的过程, 每个分组组成一个“簇”, 不同簇之间互不相交, 所有簇的并集构成整个样本集, 以簇中心表示簇内所有样本的特性. 聚类过程使得同一簇内样本的相似度尽可能高, 不同簇间样本的相似度尽可能低, 从而达到“物以类聚”的目的.
- 目前, 聚类算法被广泛应用于实际问题中, 例如：企业对客户的价值分析,由于数据量较大, 通常很难采用人工方式定义某一客户是重点客户或者一般客户, 此时可通过聚类算法将客户分为不同簇, 每个簇表示一个类别, 然后通过领域专家判断每个类别的特性.
- 本章将介绍几种较为常用的聚类算法, 在此之前, 首先介绍聚类算法中的重要概念——相似度.

1.2　相似度

1.2 相似度

- 相似度在聚类算法中至关重要, 是聚类算法的核心问题, 具体表现在两个层面：(1) 不同的聚类算法其相似度计算方法也不同; (2) 同一聚类算法采用不同的相似度计算将得到不同的聚类效果. 本节主要介绍数据对象间相似度和簇间相似度.

1.2.1 数据对象间相似度

- 聚类是对所有数据对象或样本进行一定的处理, 达到“物以类聚”的目标. 此过程涉及不同样本之间相似度的刻画, 选取不同的相似度将产生不同的聚类结果, 本节介绍几种样本之间相似度的度量方法.

- 每一个样本可以看作是一个向量, 所有样本则是向量空间中点的集合, 可以采用向量之间的距离度量刻画样本之间的相似度. 假设 $\boldsymbol{X}=(X_1,\cdots,X_p)^{\mathrm{T}}$、$\boldsymbol{Y}=(Y_1,\cdots,Y_p)^{\mathrm{T}}$、$\boldsymbol{Z}=(Z_1,\cdots,Z_p)^{\mathrm{T}}$ 表示三个样本, 其中 p 表示样本特征的维数, 给定函数 $d(\cdot,\cdot)$, 若其满足下面四个性质, 则称为距离度量:

  - (1) 非负性: $\mathrm{d}(\boldsymbol{X},\boldsymbol{Y})\geqslant 0$;
  - (2) 同一性: $\mathrm{d}(\boldsymbol{X},\boldsymbol{Y})=0$ 当且仅当 $\boldsymbol{X}=\boldsymbol{Y}$;
  - (3) 对称性: $\mathrm{d}(\boldsymbol{X},\boldsymbol{Y})=\mathrm{d}(\boldsymbol{Y},\boldsymbol{X})$;
  - (4) 直递性: $\mathrm{d}(\boldsymbol{X},\boldsymbol{Y})\leqslant \mathrm{d}(\boldsymbol{X},\boldsymbol{Z})+\mathrm{d}(\boldsymbol{Z},\boldsymbol{Y})$.

1.2.1 数据对象间相似度

1. 闵可夫斯基距离 (Minkowski distance)

- 闵可夫斯基距离是常用的距离度量, 其计算公式如下:

$$\mathrm{d}(\boldsymbol{X},\boldsymbol{Y})=\left(\sum_{k=1}^{p}|X_k-Y_k|^t\right)^{\frac{1}{t}}, \tag{1.2.1}$$

  - 上式中 $t \geqslant 1$. 下面介绍闵可夫斯基距离的三种特殊形式.
  - 当 $t=1$ 时, (1.2.1) 式称为**曼哈顿距离**, 定义为

$$\mathrm{d}(\boldsymbol{X},\boldsymbol{Y})=\sum_{k=1}^{p}|X_k-Y_k|. \tag{1.2.2}$$

  - 当 $t=2$ 时, (1.2.1) 式称为**欧氏距离**, 定义为

$$\mathrm{d}(\boldsymbol{X},\boldsymbol{Y})=\left(\sum_{k=1}^{p}|X_k-Y_k|^2\right)^{\frac{1}{2}}. \tag{1.2.3}$$

1.2.1 数据对象间相似度

- 当 $t=\infty$ 时, (1.2.1) 式称为**切比雪夫距离**, 定义为

$$\mathrm{d}(\boldsymbol{X},\boldsymbol{Y})=\max_{1\le k\le p}|X_k-Y_k|. \tag{1.2.4}$$

2. 马氏距离 (Mahalanobis distance)

- 马氏距离可视为欧氏距离的一种泛化形式, 一定程度上可以克服欧氏距离中由于不同特征量纲不一致导致的距离计算的问题.
- 若样本 $\boldsymbol{X}=(X_1,\cdots,X_p)^{\mathrm{T}}$, $\boldsymbol{Y}=(Y_1,\cdots,Y_p)^{\mathrm{T}}$ 服从同一类型分布且具有相同的协方差矩阵, 其协方差矩阵为 $\boldsymbol{\Sigma}$, 则马氏距离可以表示为

$$\mathrm{d}(\boldsymbol{X},\boldsymbol{Y})=\sqrt{(\boldsymbol{X}-\boldsymbol{Y})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}-\boldsymbol{Y})}. \tag{1.2.5}$$

- 上式中, 当协方差矩阵 $\boldsymbol{\Sigma}$ 是单位矩阵, 即各个维度独立同分布时, 马氏距离等同于欧氏距离.

1.2.1 数据对象间相似度

3. 夹角余弦

- 两个向量之间的夹角余弦可用于衡量两者之间的相似性, 当把样本看作向量时, 若两者之间的夹角余弦是 0, 即角度为 $\frac{\pi}{2}$, 则可理解为两者之间是不相关的; 若夹角余弦是 1, 说明两者指向相同方向; 若夹角余弦是 -1, 说明两者指向相反方向. 夹角余弦绝对值越靠近 1, 表明样本之间的相似度越高.

- 因此 $\boldsymbol{X}$ 与 $\boldsymbol{Y}$ 的夹角余弦定义为

$$\mathrm{d}(\boldsymbol{X},\boldsymbol{Y})=\left|\frac{\sum_{k=1}^{p}X_kY_k}{\left[\left(\sum_{k=1}^{p}X_k^2\right)\left(\sum_{k=1}^{p}Y_k^2\right)\right]^{\frac{1}{2}}}\right|. \tag{1.2.6}$$

1.2.1 数据对象间相似度

4. 相关系数

- 相关系数介于 $-1,1$ 之间, 常衡量变量之间线性相关的程度, 在样本聚类算法中也可用于样本间相似度的度量, 其绝对值越接近1, 样本之间的相似度越高; 越接近0, 样本之间相似度越低. 任取两个样本 $\boldsymbol{X}_i=(X_{i1},\cdots,X_{ip})^{\mathrm{T}}$、$\boldsymbol{Y}_j=(Y_{j1},\cdots,Y_{jp})^{\mathrm{T}}$, 两者之间的相关系数定义为

$$\mathrm{d}(\boldsymbol{X}_i,\boldsymbol{Y}_j)=\left|\frac{(\boldsymbol{X}_i-\overline{\boldsymbol{X}}_i)^{\mathrm{T}}(\boldsymbol{Y}_j-\overline{\boldsymbol{Y}}_j)^{\mathrm{T}}}{\|\boldsymbol{X}_i-\overline{\boldsymbol{X}}_i\|_2\|\boldsymbol{Y}_j-\overline{\boldsymbol{Y}}_i\|_2}\right|, \tag{1.2.7}$$

  - 其中 $\|*\|_2$ 表示 L_2 范数 (欧几里得范数), $\overline{\boldsymbol{X}}_i$ 和 $\overline{\boldsymbol{Y}}_i$ 分别表示样本 $\boldsymbol{X}_i$ 和 $\boldsymbol{Y}_j$ 的样本均值,

$$\overline{\boldsymbol{X}}_i=\frac{1}{p}\sum_{k=1}^{p}\boldsymbol{X}_{ik},\overline{\boldsymbol{Y}}_i=\frac{1}{p}\sum_{k=1}^{p}Y_{ik}.$$

  - 夹角余弦和相关系数常用于刻画变量相似性, 但是用于刻画样本相似性时, 其样本向量的分量的量纳要一致, 例如, X_{ik}, Y_{jk} 分别代表第 i 个样本和第 j 个样本在第 k 天的运动时长.

1.2.2 簇间相似度

- 聚类过程将样本集划分为互不相交的簇, 簇间相似度可通过计算簇间距离得到, 簇间距离可以通过样本间的距离定义. 假设 G_1、G_2 是两个不同的簇, 两者之间的距离可通过以下方式定义.

1. 最小距离

- 最小距离由簇间距离最近的两个样本决定, 定义为

$$D_{12} = \min_{X_i \in G_1, X_j \in G_2} d_{ij}, \tag{1.2.8}$$

  - 其中 d_{ij} 表示样本 $\boldsymbol{X}_i$ 与 $\boldsymbol{X}_j$ 之间的距离.

2. 最大距离

- 最大距离由簇间距离最远的两个样本决定, 定义为

$$D_{12} = \max_{X_i \in G_1, X_j \in G_2} d_{ij}. \tag{1.2.9}$$

1.2.2 簇间相似度

3. 平均距离

- 平均距离等于两簇之间所有样本间的距离的平均值, 定义为

$$D_{12}=\frac{1}{|G_1||G_2|}\sum_{X_i\in G_1,X_j\in G_2}d_{ij}, \tag{1.2.10}$$

  - 其中 $|G_1|$ 表示簇 G_1 中包含的样本个数, $|G_2|$ 表示簇 G_2 中包含的样本个数.

4. 中心距离

- 两个不同簇中心之间距离定义为中心距离, 假设 $\overline{X}_1$ 表示簇 G_1 的中心, $\overline{X}_2$表示簇 G_2 的中心, 则中心距离定义为

$$D_{12}=d_{\overline{X}_1\overline{X}_2}, \tag{1.2.11}$$

  - 其中,

$$\overline{X}_1=\frac{1}{|G_1|}\sum_{X_i\in G_1}X_i,\quad \overline{X}_2=\frac{1}{|G_2|}\sum_{X_j\in G_2}X_j.$$

1.3　K 均值聚类

1.3.1 原理

- K 均值也称为 K-means, 是一种原理简单、应用广泛的聚类算法. 其目标是将包含 n 个样本的数据集, 按照某种规则划分到 K 个互不相交的子集中,称为 K 个簇, 相同簇中样本的相似度较高, 不同簇之间样本的相似度较低.
- 假设样本集为 $\mathbf{X}=(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_n)^{\mathrm{T}}$, 其中 $\boldsymbol{X}_i$ 表示第 i 个样本. 令 $\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\cdots,\boldsymbol{\mu}_K$ 是 K 个向量, 表示 K 个簇中心, 其中 $\boldsymbol{\mu}_k$ 代表第 k 个簇的中心. K均值聚类采用欧氏距离计算样本与簇中心之间的距离, 然后将样本置入距离最近的簇中心所代表的簇中, 这样形成 K 个互不相交的集合(簇) $G1,G2,\cdots,G_K$, 当 $j=k$ 时满足 $G_j\cap G_k=\varnothing$, 并且 $\bigcup_{k=1}^{K}G_k=\mathbf{X}$.
- 基于欧氏距离, K 均值聚类算法的目标函数为

$$L(\Theta)=\sum_{k=1}^{K}\sum_{\boldsymbol{X}_i\in G_k}\left\|\boldsymbol{X}_i-\boldsymbol{\mu}_k\right\|^2, \tag{1.3.1}$$

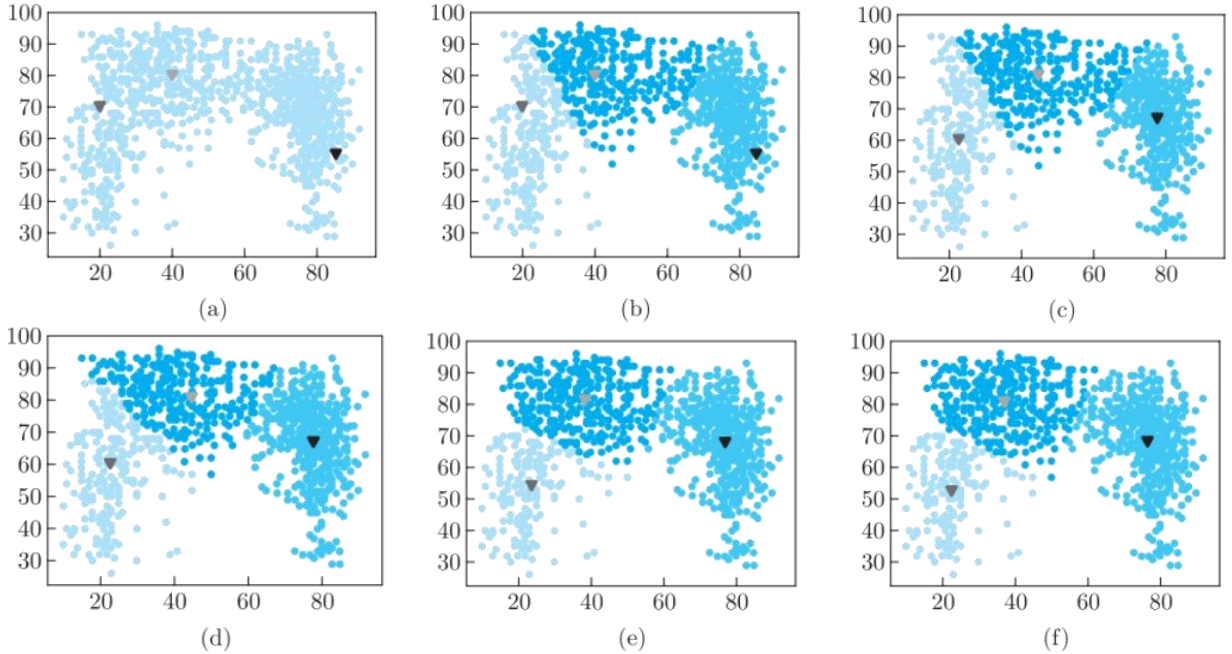
  - 其中, $\mu_k=\dfrac{1}{|G_k|}\sum_{\boldsymbol{X}_i\in G_k}\boldsymbol{X}_i$.

1.3.1 原理

- 计算上述目标函数的最优解并非易事, K 均值聚类算法采用迭代更新策略得到优化问题的近似解. 首先选取初始簇中心 $\boldsymbol{\mu}_k, k = 1, 2, \cdots, K$, 然后按照欧氏距离最小原则将样本划入不同簇中, 最后更新簇中心, 以便得到更小的 $L(\Theta)$ 值. 将此过程不断迭代下去, 直到聚类结果基本保持不变, 停止计算.K 均值聚类算法的流程如下:
  - (1) 从样本集 $\mathbf{X}$ 中随机选取 K 个样本向量 $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_K\}$ 作为初始簇中心.
  - (2) 计算每一个样本到 K 个簇中心的距离, 并将样本划入到距离最小的簇中心所代表的簇中.
  - (3) 重新计算每个簇的中心向量 $\{\boldsymbol{\mu}'_1, \boldsymbol{\mu}'_2, \cdots, \boldsymbol{\mu}'_K\}$. 若 $\boldsymbol{\mu}'_k \neq \boldsymbol{\mu}_k$, 则将 μ_k更新为 $\boldsymbol{\mu}'_k$.
  - (4) 不断迭代第 (2) 步和第 (3) 步, 直到所有的簇中心不再更新为止.
  - (5) 算法结束, 输出划分结果.

1.3.1 原理

- 图 1.1 展示了 K 均值聚类的整个过程, 每张图代表每一次的聚类结果.首先从原始数据集中随机选取三个样本作为初始聚类中心, 分别用不同颜色的三角形表示, 如图 1.1(a) 所示. 之后计算每个样本与三个聚类中心的距离,根据距离最近原则将样本划分为颜色各异的三个簇, 如图 1.1(b) 所示. 随后,重新计算每个簇的样本均值, 作为新的三个聚类中心, 如图 1.1(c) 所示, 此时簇中心已经发生改变, 计算样本到新的簇中心之间的距离, 得到新的簇划分.不断迭代此过程, 直至聚类中心不再变化.



1.3.2 特点

- K 均值聚类是机器学习中常用的无监督学习算法, 其优点是易于实现、拥有较好的性能, 并且相比于其他机器学习算法具有较少的超参数. 然而, K的取值以及初始簇中心的选择对算法有较大的影响. K 的取值关乎聚类输出的效果以及可解释性, 应当选取合适的 K 值作为聚类簇数. 初始簇中心的选择影响算法的迭代效果, 好的初始簇中心能够加快算法的收敛, 反之将影响收敛速度. 本节将从这两个方面对 K 均值算法进行分析.

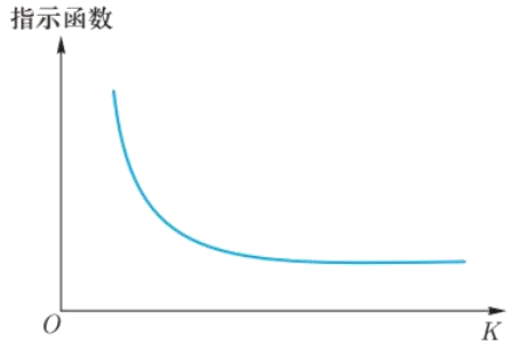
1. K 值的确定

- 在 K 均值算法中, 簇个数 K 的选择对算法最终的输出结果影响较大, 不同的 K 值将产生不同的聚类效果, 在实际应用中, K 的最优值通常较难获取,本节介绍三种 K 值确定方法.

  - (1) 根据数据的先验知识确定 K 值, 或者可通过简单数据分析得到 K 的可能取值.

1.3.2 特点

- (2) 指示函数确定法：也可称为肘部确定法. 定义一个指示函数, 如图1.2, 横轴表示 K 的取值, 纵轴表示指示函数的取值. 函数值随着 K 值的变化而变化, 对于选定的某个临界值, 当 K 大于该值时, 函数的变化趋于平缓, 反之小于该值时, 函数值的变化较为剧烈, 则可选择该值作为最优 K 值. 其中指示函数可以选择误差平方和 ((1.3.1) 式)、平均直径、平均半径等.



- (3) 平均轮廓系数法：轮廓系数的计算公式如下：

$$R(\boldsymbol{X}_i)=\frac{B(\boldsymbol{X}_i)-I(\boldsymbol{X}_i)}{\max\{B(\boldsymbol{X}_i),I(\boldsymbol{X}_i)\}}. \tag{1.3.2}$$

- 上式中, $\boldsymbol{X}_i$ 表示第 i 个样本向量, $I(\boldsymbol{X}_i)$ 表示样本 $\boldsymbol{X}_i$ 与其所在簇中其他样本之间的平均距离, $B(\boldsymbol{X}_i)$ 表示样本 $\boldsymbol{X}_i$ 与其他簇中所有样本之间的平均距离. 由式 (1.3.2) 可计算出每个样本的轮廓系数, 然后再求平均值, 得到平均轮廓系数. 一般情形下, 平均轮廓系数越接近 1, 聚类效果越好.

1.3.2 特点

2. 初始簇中心的选择

- 实际问题中, 合适的初始簇中心对算法至关重要. 本部分介绍四种选择初始簇中心的方法.

  - (1) 根据对数据集的先验知识, 由领域专家设定初始簇中心.
  - (2) 采用不同的簇中心, 多次运行算法, 选取最优的初始簇中心.
  - (3) 选择尽可能远离的 K 个点：首先从样本集合中选择一个样本, 然后选择距离此样本最远的样本作为第二个簇中心, 接着选择距离第一和第二簇中心的中点最远的样本作为第三个簇中心, 依次进行下去, 直到选出 K 个初始簇中心.
  - (4) 采用层次聚类 (1.6 节) 等算法进行初始聚类, 利用这些类的中心作为K 均值算法的初始簇中心.

1.4　模糊 C 均值聚类

1.4 模糊 C 均值聚类

- K 均值聚类算法将样本划分到不同簇, 其结果具有非黑即白的性质, 即非 0 即 1, 所有样本对象均被硬性划分到某一簇类别, 在很多实际问题中可以达到较好效果, 如是否下雨、是否旅行等; 然而, 这种硬聚类的方式不适用于类别界限不明确的问题, 如健康程度、冷热程度等. 模糊 C 均值聚类 (fuzzyC-means, FCM) 采用软聚类思想, 用隶属度描述样本与簇之间的关系, 隶属度表示样本属于某一簇的确定程度.

- 与 K 均值聚类相似, FCM 算法也是通过最小化目标函数, 不断迭代更新聚类簇中心, 当目标函数收敛时, 算法停止迭代; 不同的是, FCM 算法在 K 均值聚类算法目标函数的基础上, 增加了隶属度度量, 其表达式为

$$L(\Theta)=\sum_{i=1}^{n}\sum_{j=1}^{C}\omega_{i,j}^{m}\left\|\boldsymbol{X}_i-\boldsymbol{\mu}_j\right\|^2,\quad 1\le m<\infty, \tag{1.4.1}$$

  - 其中, n 为样本总个数, C 为聚类中心数, m 为隶属度参数, $\boldsymbol{X}_i$ 表示第 i 个样本, $\boldsymbol{\mu}_j$ 表示第 j 个聚类中心, $\omega_{i,j}{}^m$ 用于刻画隶属度并且满足

$$\sum_{j=1}^{C}\omega_{i,j}=1$$

1.4 模糊 C 均值聚类

- 通过拉格朗日乘子法可以得到：

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n} \omega_{i,j}^{m} \boldsymbol{X}_i}{\sum_{i=1}^{n} \omega_{i,j}^{m}}, \quad \omega_{i,j} = \frac{1}{\sum_{k=1}^{C} \frac{\left\| \boldsymbol{X}_i - \boldsymbol{\mu}_i \right\|^{\frac{2}{m-1}}}{\left\| \boldsymbol{X}_i - \boldsymbol{\mu}_k \right\|^{\frac{2}{m-1}}}}.$$

- FCM 算法通过不断迭代更新 $\omega_{i,j}{}^{m}$ 与 $\boldsymbol{\mu}_j$ 的取值, 最终得到聚类结果, 迭代终止条件可设置为

$$\max_{i,j} \left\{ \left| \omega_{i,j}^{m,s+1} - \omega_{i,j}^{m,s} \right| \right\} < \varepsilon,$$

  - 其中, s 表示迭代次数, ε 为给定的误差阈值. 其含义为, 当两次迭代之间的最大隶属度变化量小于给定阈值时, 说明继续迭代隶属度的变化量也会非常小,此时停止迭代, 得到最终的聚类结果.

1.4 模糊 C 均值聚类

- FCM 算法的算法流程如下：
  - (1) 超参数确定：选择合适的聚类簇数 C、隶属度参数 m 以及误差阈值ε;
  - (2) 初始化隶属度矩阵 $\boldsymbol{\omega}^{m,0}$;
  - (3) 计算并更新聚类中心 $\boldsymbol{\mu}_j$;
  - (4) 计算并更新隶属度矩阵 $\boldsymbol{\omega}^{m,s}$;
  - (5) 比较更新前后隶属度矩阵的最大变化量, 如果小于 ε, 算法停止, 否则返回第 (3) 步.

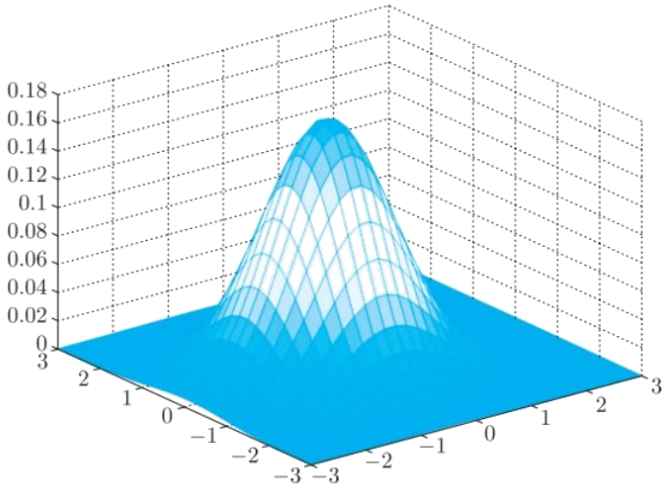
1.5　高斯混合聚类

1.5 高斯混合聚类

- 高斯混合聚类是在概率框架下实施聚类过程, 采用软聚类的思想计算样本被划分到不同簇的概率, 选取概率最大的簇作为最终的划分结果.
- 在阐述高斯混合聚类算法之前, 首先给出多变量高斯分布概率密度函数的定义：若样本空间中的 p 维随机向量 $\boldsymbol{X}=(X_1,\cdots,X_p)^{\mathrm{T}}$ 服从高斯分布,则其概率密度函数为

$$P(\boldsymbol{X}|\boldsymbol{\mu},\boldsymbol{\Sigma})=\frac{1}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}e^{-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^{T}\boldsymbol{\Sigma}(\boldsymbol{X}-\boldsymbol{\mu})}. \tag{1.5.1}$$

  - 其中, $\boldsymbol{\mu}$ 表示均值向量, $\boldsymbol{\Sigma}$ 表示协方差矩阵, 显然其概率密度函数由 $\boldsymbol{\mu}$ 和 $\boldsymbol{\Sigma}$唯一确定. 下面以二维高斯分布为例, 给出其概率密度函数图像 (如图 1.3).



1.5 高斯混合聚类

- 高斯混合模型 (Gaussian mixture model, GMM) 的聚类算法可分为三个环节：
  - (1) 假设样本集是由多个高斯分布生成, 每个分布的概率密度函数为式(1.5.1), 不同高斯分布的 $\boldsymbol{\mu}$ 和 $\boldsymbol{\Sigma}$ 值也不同;
  - (2) 采用 EM 算法不断迭代更新 $\boldsymbol{\mu}$ 和 $\boldsymbol{\Sigma}$ 的值拟合样本数据, 直到算法收敛或者达到最大迭代次数为止;
  - (3) 利用所计算出的概率密度函数对样本集进行聚类.
- 其中第 (2) 步是整个算法的核心, 下面将详细推导参数的更新策略.
- 其中, EM(expectation-maximization) 算法是一种用于估计含有潜在变量 (latent variables) 的概率模型参数的迭代优化算法, 广泛应用于高斯混合模型的参数估计.
- EM 算法是一种迭代优化算法, 用于估计概率模型参数, 特别适用于包含潜在变量的模型. 其中, E 步：使用当前参数估计计算潜在变量的后验概率(在给定观测数据的条件下潜在变量的概率分布); M 步：最大化完全数据 (包括观测数据和潜在变量) 的对数似然函数, 更新模型参数.

1.5 高斯混合聚类

- 高斯混合模型和 EM 算法的结合允许数据以不同的概率分布组合来表示, 更灵活地适应各种数据分布. 总体来说, EM 算法在高斯混合模型中的应用允许通过迭代优化来估计模型参数, 从而实现对复杂数据分布的建模.EM 算法的有效性在于其对包含潜在变量的概率模型参数估计的鲁棒性和收敛性.
- 假设高斯混合聚类模型由 K 个高斯分布组成, 每个高斯分布称为一个成分, 每个成分对应一个簇划分, 对高斯分布进行线性组合得到高斯混合聚类的概率密度函数为

$$P(\boldsymbol{X})=\sum_{k=1}^{K}\lambda_k P(\boldsymbol{X}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k), \tag{1.5.2}$$

  - 其中, $\boldsymbol{\mu}_k$ 和 $\boldsymbol{\Sigma}_k$ 分别表示第 k 个混合成分的均值向量与协方差矩阵. λ_k 是第k 个混合成分系数 (权重), 并且满足

$$\sum_{k=1}^{K}\lambda_k=1.$$

1.5 高斯混合聚类

- 可以采用最大似然估计求解模型参数 λ_k, $\boldsymbol{\mu}_k$ 以及 $\boldsymbol{\Sigma}_k$. 假设样本集 $\mathbf{X}=(\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n)^{\mathrm{T}}$, 对数似然函数可以定义为

$$L(\Theta)=\ln\left(\prod_{i=1}^{n} P(\boldsymbol{X}_i)\right)=\sum_{i=1}^{n}\ln\sum_{k=1}^{K}\lambda_k P(\boldsymbol{X}_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k). \tag{1.5.3}$$

  - 此时的目标转化为求使得式 (1.5.3) 最大化的 λ_k、$\boldsymbol{\mu}_k$ 和 $\boldsymbol{\Sigma}_k$, 其中 $1 \leqslant k \leqslant K$.对于 $\boldsymbol{\mu}_k$ 和 $\boldsymbol{\Sigma}_k$ 的求解, 可分别对其求导, 并令导数等于 0, 即

$$\frac{\partial L(\Theta)}{\partial \boldsymbol{\mu}_k}=0, \tag{1.5.4}$$

$$\frac{\partial L(\Theta)}{\partial \boldsymbol{\Sigma}_k}=0. \tag{1.5.5}$$

  - 求解式 (1.5.4), 可以得到,

$$\boldsymbol{\mu}_k=\frac{\sum_{i=1}^{n}\theta_{ik}\boldsymbol{X}_i}{\sum_{i=1}^{n}\theta_{ik}}, \tag{1.5.6}$$

1.5 高斯混合聚类

- 其中,

$$\theta_{ik}=\frac{\lambda_k P\left(\boldsymbol{X}_i\middle|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)}{\sum_{k=1}^{K}\lambda_k P\left(\boldsymbol{X}_i\middle|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)}.$$

- θ_{ik} 表示第 i 个样本 $\boldsymbol{X}_i$ 被划入第 k 个混合成分的概率, 高斯混合聚类选取$\max\{\theta_{ik}\}$, $k=1,2,\cdots,K$ 作为 $\boldsymbol{X}_i$ 的簇划分. 同理求解式 (1.5.5), 可以得到

$$\boldsymbol{\Sigma}_k=\frac{\sum_{i=1}^{n}\theta_{ik}\left(\boldsymbol{X}_i-\boldsymbol{\mu}_k\right)\left(\boldsymbol{X}_i-\boldsymbol{\mu}_k\right)^T}{\sum_{i=1}^{n}\theta_{ik}}. \tag{1.5.7}$$

- 下面考虑 λ_k 的求解方法. 由于 λ_k 满足条件

$$\sum_{k=1}^{K}\lambda_k=1,\lambda_k\geq 0,$$

1.5 高斯混合聚类

- 因此需求解带约束的极值优化问题, 采用拉格朗日乘子法容易得到

$$\lambda_k = \frac{1}{n}\sum_{i=1}^{n}\theta_{ik}. \tag{1.5.8}$$

- 通过上述推导过程可以看出, 高斯混合聚类可以采用 EM 算法对参数进行更新, 具体可分为两个步骤：(1) 根据前一个迭代步 (或初始值) 的参数值计算 θ_{ik}; (2) 利用第 (1) 步求得的 θi_k 更新 λ_k、$\boldsymbol{\mu}_k$ 和 $\boldsymbol{\Sigma}_k$.
- 高斯混合聚类算法的流程可以概括为：
  - (1) 参数初始化, 包括 λ_k、$\boldsymbol{\mu}_k$、$\boldsymbol{\Sigma}_k$ 以及混合成分个数 K.
  - (2) 根据前一个迭代步 (或初始值) 的参数值计算 θ_{ik}.
  - (3) 利用式 (1.5.6)、(1.5.7)、(1.5.8) 更新模型参数 $\boldsymbol{\mu}_k$、$\boldsymbol{\Sigma}_k$、λ_k. 若满足停止条件, 则停止迭代; 否则返回第 (2) 步, 继续迭代计算.
  - (4) 根据所求得到 θ_{ik}, 将 $\boldsymbol{X}_i$ 划入到相应簇划分 C_k, $C_k = C_k \cup \boldsymbol{X}_i$.
  - (5) 得到最终的簇划分 $C = \{C_1, C_2, \cdots, C_K\}$, 算法结束.

1.6　层次聚类

1.6 层次聚类

- 顾名思义, 层次聚类是一个逐层进行聚类的过程, 通过计算相似度形成一种树形结构. 按照形成树形结构方式的不同可以分为聚合聚类和分裂聚类.
- 聚合聚类是自下而上进行聚类, 初始状态将每个样本当作一个簇, 然后按照簇间距离最近原则合并两个簇, 组成一个新簇, 不断迭代此过程, 直到满足条件为止; 分裂聚类是自上而下进行聚类, 初始状态是将样本集看作一个簇, 然后按照簇间距离最远的原则将样本划分到两个新的簇, 不断迭代此过程, 直到满足条件为止.
- 本节以聚合聚类为例, 介绍层次聚类的算法思想. 假设样本集为 $\mathbf{X}=(\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n)^{\mathrm{T}}$, 样本个数为 n, 特征维数为 p, 将样本集划分为 K 个簇, 则聚合聚类的算法流程为:
  - (1) 将 n 个样本分为 n 个簇, 即每个簇中只包含一个样本.
  - (2) 对于给定数据集, 通过计算各簇之间的距离得到距离矩阵$\boldsymbol{D}=(d_{ij})_{n\times n}$. 其中$d_{ij}$表示第i个簇与第$j$个簇之间的距离.
  - (3) 将距离最近的两个簇合并为一个新簇.
  - (4) 计算新簇与其他簇之间的距离, 并更新距离矩阵.
  - (5) 不断迭代 (3)、(4) 两个过程, 直到满足停止条件.

1.6 层次聚类

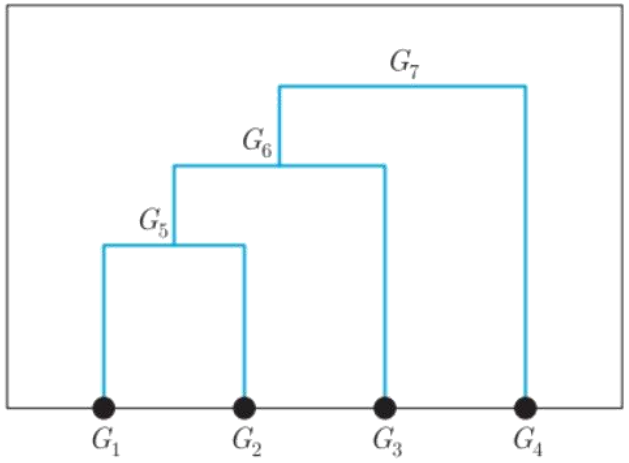
- 在聚合聚类算法中, 样本之间的距离 (相似度) 计算可以采用 1.2.1节中的方法, 簇间距离的计算可采用 1.2.2 节中的方法. 通常根据距离最小的原则将相似簇进行合并, 算法的停止条件可以是达到设定的簇的个数 K 或者达到某个指示函数的阈值.
- 下面通过例 1.1 阐述聚合聚类的算法流程.
- **例 1.1** 假设某数据集中有 4 个样本, 通过计算样本之间的欧氏距离得到如下距离矩阵 $\boldsymbol{D}$, 请利用聚合聚类算法将 4 个样本进行聚类.

$$\boldsymbol{D}=\begin{pmatrix}0 & 2 & 5 & 4\\ 2 & 0 & 3 & 5\\ 5 & 3 & 0 & 6\\ 4 & 5 & 6 & 0\end{pmatrix}.$$

1.6 层次聚类

解

- (1) 将 4 个样本分为 4 个簇 G_1, G_2, G_3, G_4, 每个簇中只有一个样本,则簇间距离矩阵为 $\boldsymbol{D}$.
- (2) 根据 $\boldsymbol{D}$, 挑选出距离最近的两个簇 G_1, G_2, 将这两个簇合并为一个新簇 $G_5 = \{G_1, G_2\}$, 此时已划分为 3 个簇 G_3, G_4, G_5.
- (3) 分别计算新簇 G_5 与 G_3, G_4 之间的最短距离, 得到最短距离大小是$D_{35} = 3, D_{45} = 4$.
- (4) 可以看出, 距离最近的 2 个簇是 G_3, G_5, 合并这两个簇形成一个新簇$G_6 = \{G_3, G_5\} = \{G_1, G_2, G_3\}$. 此时已划分为 2 个簇 G_4, G_6.
- (5) 将 G_4, G_6 合并为新簇 $G_7 = \{G_4, G_6\} = \{G_1, G_2, G_3, G_4\}$, 算法结束.聚类过程可以表示为图 1.4.



1.6 层次聚类

- 层次聚类可以达到“分层”的效果, 通过选取不同的聚类簇数, 得到不同层次的聚类结果. 然而算法实现过程的复杂度较高, 计算量通常高于 K 均值算法.

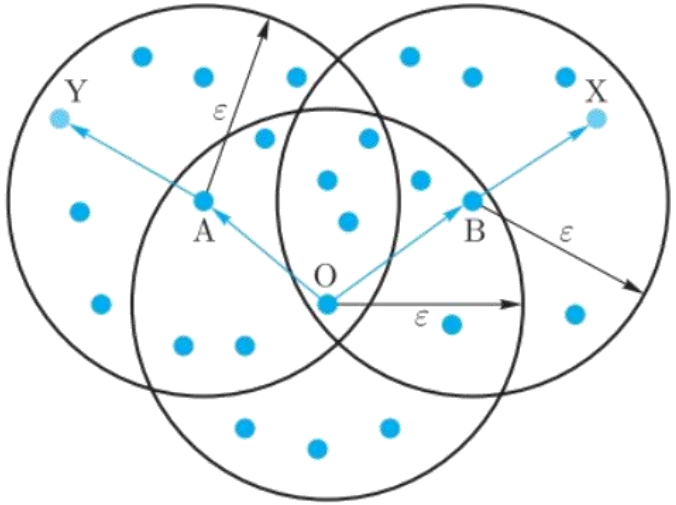
1.7　DBSCAN 聚类

1.7 DBSCAN 聚类

- 通常情形下, K 均值聚类一般适用于样本集具有凸形簇状结构 (形似“椭球”的簇结构), 对非凸样本集效果不够理想. 本节介绍一种既可用于凸样本集, 也可用于非凸样本集的 DBSCAN (density-based spatial clustering of applications with noise) 方法.
- DBSCAN 是一种基于密度的聚类方法, 算法假设样本的类别划分可由样本之间分布的紧密程度决定, 紧密程度较高的样本被划分到相同簇, 反之, 被划分到不同簇, 最终, 通过样本之间的紧密程度得到聚类结果. 由此可见, 紧密程度的刻画是 DBSCAN 算法的核心.
- DBSCAN 算法采用邻域描述样本集分布的紧密程度, 邻域由 (ε, M) 表示, 其含义是某一样本 ε 邻域内的样本个数不超过 M. 下面给出与描述样本集 $\mathbf{X} = (\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n)^{\mathrm{T}}$ 紧密程度相关的一些概念:
  - ε-邻域: 对于 $\mathbf{X}$ 中的任一样本 $\boldsymbol{X}_j$, 其 ε 邻域 $N_\varepsilon(\boldsymbol{X}_j)$ 表示与该样本距离不大于 ε 的所有样本组成的集合, 即 $N_\varepsilon(\boldsymbol{X}_j) = \{\boldsymbol{X}_i \in \mathbf{X} | \mathrm{d}(\boldsymbol{X}_i, \boldsymbol{X}_j) \leqslant \varepsilon\}$, 其中 $\mathrm{d}(\boldsymbol{X}_i, \boldsymbol{X}_j)$ 表示 $\boldsymbol{X}_i$ 与 $\boldsymbol{X}_j$ 之间的距离, 例如闵可夫斯基距离.
  - 核心对象: 对于 $\mathbf{X}$ 中的任一样本 $\boldsymbol{X}_j$, 若其所对应的 $N_\varepsilon(\boldsymbol{X}_j)$ 中样本个数不小于 M, 则称 $\boldsymbol{X}_j$ 是核心对象.
  - 密度直达: 若 $\mathbf{X}_i \in N_\varepsilon(\boldsymbol{X}_j)$, 并且 $\boldsymbol{X}_j$ 是一个核心对象, 则称样本 $\boldsymbol{X}_i$ 由样本 $\boldsymbol{X}_j$ 密度直达.

1.7 DBSCAN 聚类

- 密度可达: 对于 **X** 中的任意两个样本 $\mathbf{X}_i$ 和 $\boldsymbol{X}_j$, 若存在样本序列 $q_1,q_2,\cdots,q_t$ 满足 $q_1=\boldsymbol{X}_j$, $q_t=\mathbf{X}_i$, 并且 q_{l+1} 由 q_l 密度直达, 其中 $1 \leqslant 1 \leqslant t-1$,则称样本 $\mathbf{X}_i$ 由样本 $\boldsymbol{X}_j$ 密度可达.
- 密度相连: 对于 **X** 中的任意两个样本 $\mathbf{X}_i$ 和 $\boldsymbol{X}_j$, 若存在一个样本 $\boldsymbol{X}_l$ 使得 $\mathbf{X}_i$和 $\boldsymbol{X}_j$ 均可由 Xl 密度可达, 则称 $\mathbf{X}_i$ 和 $\boldsymbol{X}_j$ 密度相连.
- 图 1.5 描述了上述基本概念, 如图所示, 样本 A、B 和 O 均是核心对象,Y 由 A 密度直达, A 和 B 由 O 密度直达, X 由 B 密度直达; X 和 Y 均由 O密度可达; X 与 Y 密度相连.



1.7 DBSCAN 聚类

- 基于上述基本概念, 可以给出 DBSCAN 算法中簇的定义：**由密度可达关系导出的最大密度相连的样本集合, 即为聚类结果的一个簇.**
- DBSCAN 算法的目标是从样本集中找出所有满足簇定义的簇划分, 由此可见 DBSCAN 不需要预先给定簇划分的个数. 在寻找簇划分时, DBSCAN从样本集中选择一个没有被划分的核心对象作为“种子”, 寻找由该核心对象密度可达的所有样本构成一个样本子集, 即为一个簇划分; 然后选择另外一个没有被划分的核心对象作为“种子”, 依次进行下去, 直到所有核心对象均被划分为止.
- 给定样本集 $\mathbf{X} = (\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n)^{\mathrm{T}}$, DBSCAN 算法的流程可以描述为：
  - (1) 初始化 ε、M、核心对象集 $\Omega = \varnothing$、聚类簇数 k=0 以及未被访问的样本集 $\Phi = \mathbf{X}$;
  - (2) 遍历样本集中的所有样本, 找出核心对象, 将其添加到 Ω 中, 此时 Ω存储了所有的核心对象;
  - (3) 若 $\Omega = \varnothing$, 则算法结束, 否则进行第 (4) 步;
  - (4) 记录当前未被访问的样本集合: $\Phi^* = \Phi$, 在 Ω 中随机挑选一个核心对象 o, 初始化队列 $\Omega^* = <o>$, 更新未被访问的样本集 $\Phi = \Phi - \{o\}$;

1.7 DBSCAN 聚类

- (5) 在队列 Ω^* 中随机取出一个核心对象 o, 此时 $\Omega^* = \Omega^* - <o>$, 并找出其 ε 邻域 $N_\varepsilon(o)$, 令 $\Delta = N_\varepsilon(o) \cap \Phi$, 更新 $\Phi = \Phi - \Delta$, 更新 $\Omega^* = \Omega^* \cup (\Delta \cap \Omega)$;
- (6) 若 $\Omega^* = \varnothing$, 则进行第 (7) 步, 否则返回第 (5) 步;
- (7) 令 $k = k + 1$, 更新当前簇 $C_k = \Phi^* - \Phi$, 更新 $\Omega = \Omega - C_k$, 返回第 (3)步;
- (8) 输出最终的簇划分 $C = \{C_1, C_2, \cdots, C_k\}$.
- 注：若队列 Ω^* 中的元素在此前循环中被访问过, 则 Ω^* 中将不再包含此元素.

- 合聚类等算法相比, DBSCAN 算法不需要提前确定聚类簇数, 并且可应用于非凸数据集的聚类, 在得到聚类结果的同时还可以识别出异常点. 然而, 当样本集规模较大时, 算法收敛时间较长; 并且算法调优涉及 ε 和 M 两个参数,不同参数组合对聚类的结果影响较大, 调参工作量较为复杂.

1.8　其他类型聚类方法

1.8 其他类型聚类方法

- 聚类方法多种多样, 针对不同的应用场景可能会产生比传统聚类方法更好的算法, 因此在使用聚类方法解决实际问题时, 应当根据问题本身的特点选择合适的聚类方法. 本节将介绍其他几种常用的聚类方法.

1.8.1 混合型数据聚类方法

- 在实际应用中, 数据集通常由数值型和分类型特征组成, 尤其是当数据集由多源数据集成而得时, 如医疗数据 (包括患者个人信息：姓名、学历、职业、收入情况等; 医疗监测数据：血常规、血脂等)、客户数据 (包括客户的年龄、性别、种族、消费记录、消费频次、消费金额等) 等. 针对这种混合型数据进行聚类分析时需进行特殊处理, 本节介绍一种常用的混合型数据聚类方法：K-原型.
- K-原型是基于 K 均值方法的一种可以处理混合数据的聚类方法. 与 K均值聚类方法相比, 主要存在两点不同：(1) 簇中心的选取; (2) 距离的计算.下面针对这两点进行详细阐述.
- 在 K 均值聚类中, 每个簇选取簇内样本的均值作为当前簇中心, 这种计算方式仅适用于特征都是数值型数据, 当某些特征是分类型时将无法计算.因此 K-原型方法在计算簇中心时将所有特征分为两部分：数值型、分类型.对于数值型特征依然计算均值, 对于分类型特征选择众数, 然后将两部分合并起来组成当前簇的中心. 令样本集为$\boldsymbol{X}=(\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n)^{\mathrm{T}}$, 不失一般性, 假设任意一个样本 $\boldsymbol{X}_i = (\boldsymbol{X}_{i1}, \cdots, \boldsymbol{X}_{ip})^{\mathrm{T}}$ 包含 p 个特征, 其中 q 个数值型特征、$p-q$ 个分类型特征. 令 $\{\mu_1, \mu_2, \cdots, \mu_K\}$ 表示 K 个簇中心.

1.8.1 混合型数据聚类方法

- K 均值常采用欧氏距离计算样本与簇中心之间的距离, 在 K-原型中数值型特征依然采用欧氏距离, 分类型特征采用汉明距离, 然后得到样本点与簇中心的距离. K-原型的算法流程可以描述为:
  - (1) 初始化聚类簇数 K 以及簇中心 $\{\mu_1, \mu_2, \cdots, \mu_K\}$;
  - (2) 计算数据集中样本点 $\boldsymbol{X}_i$ 到 K 个簇中心的距离, 将所有样本划入距离最近的簇中;
  - (3) 重新计算每个簇的中心;
  - (4) 重复第 (2) 步、第 (3) 步, 直到满足停止条件 (簇中心变化很小或者达到迭代的最大次数);
  - 注: 对于两个数字来说, 汉明距离就是转成二进制后, 对应的位置值不相同的个数. 例如, 假设有两个十进制数 a =93 和 b=73, 如果将这两个数用二进制表示的话, 有 a =1011101、b=1001001, 可以看出, 二者的从右往左数的第 3 位、第 5 位不同 (从 1 开始数), 因此, a 和 b 的汉明距离是 2.

1.8.2 双向聚类方法

- 传统的聚类方法可大致分为两种模式：一种是对样本进行聚类 (如根据销售数据对客户类型聚类分析); 另一种是对特征或者变量进行聚类 (如分析影响青少年成长的因素, 分为积极因素、消极因素等). 这两种模式均是对单一方向 (样本或者特征) 进行聚类, 通常被称为单向聚类.
- 尽管单向聚类可以解决多数实际问题, 但仍然存在一些特殊场景并不适合采用单向聚类进行分析, 例如利用患某种疾病的患者基因数据对患者进行划分, 通常只有少量基因对该病产生影响, 并且不同的基因组合对疾病的影响也不尽相同, 由于需要同时考虑样本和特征之间的关系, 捕捉局部性质, 单向聚类在利用所有基因进行聚类分析时效果不佳, 因此利用双向聚类方法解决此类问题得到较多的研究. 近年来双向聚类方法被广泛应用于基因表达分析、文本数据挖掘等领域, 下面介绍一种双向聚类方法——稀疏双向 K 均值聚类.
- 稀疏双向聚类的目标是从原始数据集中根据矩阵元素的相似性提取具有特定结构的子矩阵, 下面介绍几种常见的类型:

1.8.2 双向聚类方法

- 全局常数型: 子矩阵中的所有元素是同一常数, 即对于子矩阵 $\boldsymbol{B} = (b_{tr})$,其中 $1 \leqslant t \leqslant T, 1 \leqslant r \leqslant R$, 满足 $b_{tr} = c$, 其中 c 是常数, 如图 1.6(a). 实际场景中由于噪声的存在, 通常要求子矩阵在阈值 ε 范围内满足元素为常数.
- 行 (列)常数型：子矩阵的每一行 (或列) 的元素为常数. 以行常数为例,对于子矩阵 $\boldsymbol{B} = (b_{tr})$, 其中 $1 \leqslant t \leqslant T, 1 \leqslant r \leqslant R$, 满足 $b_{t1} = bt_2 = \cdots = b_{tR} = c_t$, 其中 c_t 是常数, 如图 1.6(b).
- 线性关系型：子矩阵的每行 (列) 的元素是其他任意一行 (列) 的线性组合. 以行满足线性关系为例, 对于子矩阵 B = (btr), 其中 $1 \leqslant t \leqslant T, 1 \leqslant r \leqslant R$, 满足 $\boldsymbol{B}t_1 = c_{t2} \times \boldsymbol{B}_{t2} + \theta_{t2}$, 其中 c_{t2}, θ_{t2} 是常数, $\boldsymbol{B}_{t1}$ 表示矩阵第t_1 行. 列满足线性关系同理可得. 当 $c_{t2} = 1$ 时, 被称为加法模型, 如图 1.6(c);当 $\theta_{t2} = 0$ 时, 被称为乘法模型, 如图 1.6(d).

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

(a) 全局常数型

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

(b) 行常数型

1	2	3	4
2	3	4	5
3	4	5	6
4	5	6	7

(c) 加法模型

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

(d) 乘法模型

1.8.2 双向聚类方法

- 稀疏双向 K 均值聚类是将K均值聚类拓展到稀疏双向聚类中的一种算法, 求解过程与K均值算法类似. 现给定原始数据集矩阵为$\mathbf{X}=(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_n)^{\mathrm{T}}$, 其中$n$表示样本数, p表示特征数且 $\boldsymbol{X}_i=(\boldsymbol{X}_{i1},\cdots,\boldsymbol{X}_{ip})^{\mathrm{T}}$. 假设$n$个样本可以划分为 K_1 个类别, 记为$G_1,G_2,\cdots,G_{K1}$, p 个特征可以划分为K_2 个类别, 记为$G^*_{\ 1},G^*_{\ 2},\cdots,G^*_{\ \mathrm{K2}}$. 并且假设样本 $\boldsymbol{X}_i$ 中的元素 X_{ij} 的期望$\mathrm{E}(X_{ij})=\mu_{ls}$, 其中 μ_{ls} 表示 X_{ij} 属于第 l 个样本类别、第 s 个特征类别所对应子矩阵中的元素均值. 双向聚类的目标函数为

$$\min\sum_{l=1}^{K_1}\sum_{s=1}^{K_2}\sum_{i\in G_l}\sum_{j\in G_s}\left(X_{ij}-\mu_{ls}\right)^2. \tag{1.8.1}$$

  - 由式 (1.8.1) 可以看出, 当 $K_1=1$ 时, 转化为关于特征的 K 均值聚类, 当$K_2=1$ 时, 转化为关于样本的 K 均值聚类.

- 为了增强聚类结果的可解释性, 并且减少类别方差, 可对式 (1.8.1) 中的μ_{ls} 施加 Lasso 惩罚, 此时目标函数可变为

$$\min\left\{\frac{1}{2}\sum_{l=1}^{K_1}\sum_{s=1}^{K_2}\sum_{i\in G_l}\sum_{j\in G_s}\left(X_{ij}-\mu_{ls}\right)^2+\lambda\sum_{l=1}^{K_1}\sum_{s=1}^{K_2}\left|\mu_{ls}\right|\right\}. \tag{1.8.2}$$

  - 其中 λ 是非负参数.

1.8.2 双向聚类方法

- 显然稀疏双向 K 均值聚类是 K 均值聚类的一种泛化形式, 在利用稀疏双向 K 均值聚类算法时, 通常先将原始数据集矩阵 $\mathbf{X}$ 进行中心化处理, 然后再进入算法流程. 算法参数的求解过程与 K 均值算法相似, 可以利用迭代求解的思想不断更新参数的取值, 直到参数在某个阈值 ε 范围内不再变化或者达到最大迭代次数时停止计算.

1.9　聚类实践

实践代码