机器学习常见知识汇总之无监督学习

田晓彬
xiaobin9652@163.com

聚类算法的评估

相对于监督学习，无监督学习通常没有标注数据，算法的设计直接影响最终的输出和模型的性能。首先我们先介绍常见数据簇的特点。

1. **以中心定义的数据簇**：这类数据集合倾向于球状分布，即数据簇中的所有数据的平均值为簇的中心。簇中的数据到中心的距离比到其它簇中心的距离更近。
2. **以密度定义的数据簇**：这类数据簇呈现出和其他数据簇不同的密度，或稠密或稀疏。当数据簇不规则或者相互盘绕，并且有噪声和离群点时，常常使用基于密度的簇定义。
3. **以连通定义的数据簇**：这类数据的数据点和数据点之间有联通关系，整个数据簇表现为图结构。通常为不规则形状或者缠绕的数据簇。
4. **以概念定义的数据簇**：这类数据集合中的所有数据点具有某种共同性质。
   由于数据以及需求的多样性，没有一种算法可以适用于所有的数据类型、数据簇或应用场景。每种情况可能都需要一种不同的评估方法，在很多情况下，判断聚类算法结果的好坏强烈依赖主观解释。但聚类算法的评估还是非常必要的。

聚类的评估任务是估计在数据集上进行聚类的可行性，以及聚类方法产生结果的质量。这一过程通常分为三部分：

1. **估计聚类趋势**。这一步骤检测数据分布中是否存在随机的簇结构。如果数据的随机分布的，那对数据行行聚类是毫无意义的。我们可以观察聚类误差是否随着聚类类数的增加而单调变化。如果数据分布是随机的，那么聚类误差的变化幅度应该不大。还可以使用**霍普金斯统计量**来判断数据的随机性。
2. **判定数据簇数**。确定聚类趋势之后，我们需要找到与真实数据分布最为吻合的簇数，由此来判定聚类结果的质量，如**手肘法**和**GapStatistic**。用于评估的最佳数据簇数可能与算法输出的簇数是不同的。
3. **确定聚类效果**。在无监督的情况下，我们可以通过考察簇之间的分离情况和簇内部的紧凑情况来评估聚类的效果。为了更合理的评估不同聚类算法的性能，通常还需要认为地构造不同类型的数据集，以观察聚类算法在这些数据集上的效果。

聚类算法通常分为原型聚类、密度聚类和层次聚类。K均值聚类、高斯混合聚类都为原型聚类。

K均值聚类

1.K均值聚类的具体步骤

原型聚类算法通常先对原型进行初始化，然后对原型进行迭代更新求解。采用不同的原型表示、不同的求解方式，将产生不同的算法。K均值聚类是一种原型聚类算法，它的基本思想是通过迭代的方式求解K个簇划分，使得聚类结果对应的代价函数最小。算法的具体步骤如下：

1. 数据预处理，如归一化、离群点处理等。
2. 随机选取K个簇中心，记为 $\mu_1^{(0)}, \mu_2^{(0)}, ..., \mu_K^{(0)}$ 。
3. 定义代价函数： $J(c, \mu) = \min\limits_{\mu} \min\limits_{c} \sum_{k=1}^{K} \sum_{x_i \in c_k} ||x_i - \mu_k||^2$。
4. 令 $t = 0, 1, 2, ...$ 为迭代步数，重复下面过程直到 J 收敛：
   对于每一个样本 x_i，将其分配到距离最近的簇：

$$c_i^{(t)} = \arg\min_k ||x_i - \mu_k^{(t)}||^2;$$

   对于每一个类簇 k，重新计算该类簇的中心：

$$\mu_k^{(t+1)} = \frac{1}{|c_k|} \sum_{x_i \in c_k^{(t)}} x_i;$$

   K均值算法的 c_i 和 μ_k 交替更新，直到 μ_k 不再更新为止，即 μ_k收敛。当 μ_k 收敛时，c_i 收敛，同时 J 也递减到最小值。

2.k均值聚类的优缺点

由于K均值聚类初始值的影响，会出现聚类结果不稳定、结果通常是局部最优而不是全局最优的情况。并且无法很好的解决数据簇分布差别比较大的情况，不太适用于离散分类。但由于k均值聚类的时间复杂度 $O(NKt)$ 接近线性,所以对于大数据集，k均值聚类很高效。

K均值聚类本质上是一种基于欧氏距离的数据划分方法，均值和方差的维度都会对最后的聚类结果产生很大的影响，所以未做归一化处理的数据是无法参与运算和比较的。同时离群点和噪声数据会对均值造成较大的影响，导致数据簇中心的偏移，因此通常在做k均值聚类之前通常需要对数据做预处理。

由于K均值聚类的初始值会对算法的效果有较大的影响，所以**K值的选择**是K均值聚类最大的问题之一。通产来说，K值的选择一般基于经验和多次实验结果（手肘法、GapStatistic）。

K均值算法在本质上假设各个数据簇呈现球型或者高维球形分布。当数据簇呈现非凸的分布形状时，可以引入**核函数**来进行优化。核函数可以将数据点映射到高维特征空间，在高维特征空间中，数据线性可分的概率大大增加。

3.改进k均值聚类

1. **K-means++算法**
   原始K均值算法最开始随机选取数据集中 K 个点作为聚类中心，而K-means++算法按照如下思想选取聚类中心。首先在选取第一个聚类中心时通过随机的方法，随后的聚类中心较大概率的选择与之前的聚类中心较远的点。随后的操作和原始K均值算法相同。
2. **ISODATA算法**
   ISODATA的全称是迭代自组织数据分析法。ISODATA算法针对K均值聚类算法的 K 值选择进行优化。当属于某个类别的样本数过少时，算法将该类去除；当属于某个类的样本数过多、分散程度较大时，把该类别分成两个子类别。但ISODATA算法需要指定过多参数：参考聚类数量 k_0、每个类所要求的最少样本数量 N_{min}、最大方差 σ以及两个聚类中心所允许的最小距离 D_{min}。

高斯混合聚类（高斯混合模型）

1.高斯混合模型

高斯混合模型的核心思想是，假设数据可以看作是从多个高斯分布中生成出来的。在该假设下，每个单独的分模型都是标准的高斯模型，模型的均值 μ_i 和方差 Σ_i 都是待估计的参数。每一个分模型还有一个参数 α_i 代表该分模型的权重或者是生成数据的概率。则高斯混合模型的分布为

$$p(x) = \sum_{i=1}^{k} \alpha_i p(x|\mu_i, \Sigma_i). \tag{1}$$

其中 $\alpha_i > 0$ 并且 $\sum_{i=1}^{k} \alpha_i = 1$。该分布共有 k 个混合成分组成，每个混合成分对应一个高斯分布。

若训练集符合高斯混合模型的假设，令随机变量 $\theta_j \in \{1, 2, ..., k\}$ 表示生成样本 x_j 的高斯混合成分。显然 分模型生成数据的概率 α_i 即为 θ_j 的先验概率 $p(\theta_j = i)$。根据贝叶斯定理，α_i 的后验分布为：

$$p(\theta_j = i|x_j) = \frac{p(\theta_j = i)p(x_j|\theta_j = i)}{p(x_j)} \quad = \frac{\alpha_i p(x_j|\mu_i, \Sigma_i)}{\sum_{l=1}^{k} p(x_j|\mu_l, \Sigma_l)} \tag{2}$$

$p(\theta_j = i|x_j)$ 定义为样本 x_j 由第 i 个高斯混合成分生成的后验概率。我们将其定义为 γ_{ji}。

当高斯混合成分已知时，高斯混合聚类将把样本集 D 划分成 k 个簇，每个样本 x_j 的簇标记 λ_j 确定如下：

$$\lambda_j = \underset{i \in \{1,2,...,k\}}{\arg\max}\ \gamma_{ji}. \tag{3}$$

由此可以看出，高斯混合模型是采用概率模型对数据簇原型进行刻画，为簇划分则由原型对应的后验概率确定。

对于模型参数，我们使用最大似然估计进行求解。

$$L(D) = ln(\prod_{j=1}^{n} p(x_j)) = \sum_{j=1}^{n} \ln\left(\sum_{i=1}^{k} \alpha_i p(x_j|\mu_i, \Sigma_i)\right) \tag{4}$$

对上述公式进行求偏导,并限制 $\sum_{i=1}^{k} \alpha_i = 1$，可以得到

$$\mu_i = \frac{\sum_{j=1}^{n} \gamma_{ji} x_j}{\sum_{j=1}^{n} \gamma_{ji}},$$

$$\Sigma_i = \frac{\sum_{j=1}^{n} \gamma_{ji}(x_j - \mu_i)(x_j - \mu_i)^T}{\sum_{j=1}^{n} \gamma_{ji}},$$

$$\alpha_i = \frac{1}{m} \sum_{j=1}^{n} \gamma_{ji}. \tag{5}$$

接下来使用EM算法进行迭代优化求解三个参数：在每步迭代中，现根据当前参数计算每个样本属于每个高斯成分的后验概率 γ_{ji}，再根据公式(5)更新模型参数。一直重复该迭代过程直到满足停止的条件（达到最大迭代轮数、似然函数 $L(D)$ 增长很少或者不在增长）为止。当高斯混合模型的所有分模型参数确定后，就可以根据公式(3)来进行簇划分。

2.高斯混合模型和k均值算法

高斯混合模型和k均值算法的相同点是，它们都是聚类算法；都需要指定 K 值；都使用EM算法来求解；通常只能收敛到局部最优。

高斯混合模型相对于K均值算法的优点是，可以给出一个样本属于某个数据簇的概率；不仅可以用于聚类，还可以用于概率密度估计；可以用于生成新的样本点。

DBSCAN

基于密度的聚类方法假设密度结构能通过样本分布的紧密程度确定，DBSCAN就是一种基于密度的方法，它基于一组邻域参数来刻画样本分布的紧密程度。

给定数据集 $D=\{x_1,x2,....x_n\}$，定义一下几个概念：

$\epsilon-$邻域：对于$x_j\in D$，x_j 的 $\epsilon-$ 邻域表示为 $N_\epsilon(\boldsymbol{x}_j)=\{\boldsymbol{x}_j\in D|\text{dist}(\boldsymbol{x}_i,\boldsymbol{x}_j\leq\epsilon)\}$;

核心对象：若 $\boldsymbol{x}_j$ 的 $\epsilon-$邻域至少包含 m 个样本，则 $\boldsymbol{x}_j$ 是一个核心对象；

密度直达：若 $\boldsymbol{x}_j$ 位于 $\boldsymbol{x}_i$ 的 $\epsilon-$ 邻域中，且 $\boldsymbol{x}_i$ 是一个核心对象，则称 $\boldsymbol{x}_j$ 由 $\boldsymbol{x}_i$ 密度直达；

密度可达：对 $\boldsymbol{x}_j$ 与 $\boldsymbol{x}_i$，若存在样本序列 $\boldsymbol{p}_1,\boldsymbol{p}_2,....\boldsymbol{p}_n$，其中 $\boldsymbol{x}_i=\boldsymbol{p}_1$，$\boldsymbol{x}_j=\boldsymbol{p}_n$，且 $\boldsymbol{p}_i+1$由$\boldsymbol{p}_i$密度直达，则称 $\boldsymbol{x}_j$ 由 $\boldsymbol{x}_i$ 密度可达；

密度相连：对 $\boldsymbol{x}_j$ 与 $\boldsymbol{x}_i$，若存在 $\boldsymbol{x}_k$ 使得 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 均有 $\boldsymbol{x}_k$ 密度可达，则称 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 密度相连。

DBSCAN将簇定义为：由密度可达关系导出的最大的密度相连样本集合。

AGNES

层次聚类试图在不同层次对数据集进行划分，从而形成树型的聚类结构。AGNES是一种采用自底向上聚类策略的层次聚类算法。它先将数据集中的每个样本看作一个初始聚类簇，然后在算法运行的每一步中找出距离最近的两个聚类簇进行合并，该过程不断重复，直至达到预设的聚类簇个数。两个聚类簇的距离可由如下三种方式定义，给定聚类簇 C_i 和 C_j，可通过下面的式子计算距离：

$$\text{最小距离：}d_min(C_i,C_j)=\min_{\boldsymbol{x}\in C_i,\boldsymbol{z}\in C_j}dist(\boldsymbol{x},\boldsymbol{z}),$$

$$\text{最大距离：}d_max(C_i,C_j)=\max_{\boldsymbol{x}\in C_i,\boldsymbol{z}\in C_j}dist(\boldsymbol{x},\boldsymbol{z}),$$

$$\text{平均距离：}d_avg(C_i,C_j)=\frac{1}{|C_i||C_j|}\sum_{\boldsymbol{x}\in C_i}\sum_{\boldsymbol{z}\in C_j}dist(\boldsymbol{x},\boldsymbol{z}),$$

当聚类簇距离由 d_min、d_max 或 d_avg 计算时，AGNES算法被相应地称为“单链接”、“全链接”或“均链接”算法。