Cluster Algorithm

2022-07-10

Note: 该文章基本只是对几篇文章的整合，只有细微的细改，原文章在参考下。

参考：

相似性衡量，聚类算法以及 data reduction 的整体介绍
距离计算的详细内容，每种聚类方法的实现过程，example
聚类结果的内部价指标
聚类结果的外部评价指标

聚类分析：根据样本特征计算样本距离。需要考虑的点，聚类算法，相似度的衡量。

相似度衡量 (similarity)

相似系数，核函数 $K(x, y)$ 以及 DTW

相似系数，包括夹角余弦和相关系数。其主要优势在于不受原线性变换的影响，可以轻松转化为距离，但其运算速度比距离法慢的多，当维数很高的时候。

核函数 $K(x, y)$，定义在 $R^d * R^d$ 的二元函数，本质也是距离。核函数的功能是把数据从低维到高维进行投影。

DTW(dynamic time wraping)。可以用于计算两个不同长度向量的距离，也可以对两对向量中不同时间段的数据做匹配。主要用于时间序列分析。

距离

数值变量

- Minkowski 距离：其实就是 L_p norm。考虑两个向量，$X = (x_1, x_2, ..., x_p)$, $Y = (y_1, y_2, ..., y_p)$.

$$d(X,Y) = \sqrt[q]{|(x_1 - y_1)|^q + |(x_2 - y_2)|^q + ... + |(x_p - y_p)|^q}$$

- Manhattan 距离：Minkowski，q = 1 的特例。

$$d(X,Y) = |(x_1 - y_1)| + |(x_2 - y_2)| + ... + |(x_p - y_p)|$$

- Euclidean 距离：Minkowski, q = 2 的特例。

$$d(X,Y) = \sqrt[2]{|(x_1 - y_1)|^2 + |(x_2 - y_2)|^2 + ... + |(x_p - y_p)|^2}$$

- supremum 距离, 也叫切比雪夫距离，$q \to +\infty$

Mahalanobis 距离：权重向量 $W = (\omega_1, \omega_2, ..., \omega_p)$，主要用于 *Gaussian Mixture Model(GMM)*

$$d(X,Y) = \sqrt[q]{\omega_1 * |(x_1 - y_1)|^q + \omega_2 * |(x_2 - y_2)|^q + ... + \omega_p * |(x_p - y_p)|^q}$$

Note: 考虑到不同特征的尺度不一致，对特征做标准化处理。$Z_f = \frac{X_f - mean_f}{S_f}$

二元变量

$$d(X,Y) = \frac{unpaired\ features}{unpaired\ feature + paired\ positive}$$

Note: 为什么分母没有考虑 *paired negativae*, 因为 *paired negative* 说明是两者都没有的属性，那这样的属性可以说是无穷多的，计算上也没什么意义，所以不考虑，该说法由 *Jaccard* 提出，所以该距离称为 *Jaccard distance*。

分类变量

- 简单匹配

$$d(X,Y) = \frac{paired\ feature}{category\ number}$$

- 分类变量二值化，即将多类归为两类。

有序变量

考虑 $Level \in low, middle, high, ...$

1. 用 $1, 2, ..., N$ 定义 *level* 排序。
2. 对 *level* 进行 *z score* 标准化。
3. 计算 *level* 的 Minkowski 距离。

聚类算法

Hierarchical methods

主要有两种路径：agglomerative 和 divisive, 也可以理解为自下而上法 (bottom-up) 和自上而下法 (top-down) 。自下而上法，就是一开始每个个体 (object) 都是一个类，然后根据巧 linkage 寻找同类，最后形成一个” 类” 。自上而下法就是反过来，一开始所有个体都属于一个” 类，然后根据 linkage 排除异已，最后每个个体都成为一个” 类” 。这两种路径本质上没有优劣之分，只是在实际应用的时候要根据数据特点以及你想要的” 类” 的个数，来考虑是自上而下更快还是自下而上更快。至于根据 Linkage 判断” 类” 的方法就是楼上所提到的最短距离法、最长距离法、中间距离法、类平均法等等 (其中类平均往往被认为是最常用也最好用的方法，一方面因为其良好的单调性，另一方面因为其空间扩张/ 浓缩的程度适中) ，HierarchicaI methods 中比较新的算法有 BIRCH (BaIanced lterative Reducing and clustering Using Hierarchies) 主要是在数据体量很大的时候使用，而且数据类型是 numerical; ROCK (A HierarchicaI CIustering Algorithm for CategoricaI Attributes) 主用在 categorical 的数据类型上; ChameIeon (A HierarchicaI CIustering Algorithm Using Dynamic ModeIing) 里用到的 linkage 是 kNN (k-nearest-neighbor) 算法，并以此构建一个 graph。Chameleon 的聚类效果被认为非常强大，比 BIRCH 好用，但运算复杂度很高。

- example

把每一个单个的观测都视为一个类，而后计算各类之间的距离，选取最相近的两个类，将它们合并为一个类。新的这些类再继续计算距离，合并到最近的两个类。如此往复，最后就只有一个类。然后用树状图记录这个过程，这个树状图就包含了我们所需要的信息。类的数量取决于你从树状图哪里剪。

1. 计算类与类之间的距离，用邻近度矩阵记录。
2. 将最近的两个类合并为一个新的类。
3. 根据新的类，更新邻近度矩阵。
4. 重复 2. 3。
5. 到只只剩下一个类的时候，停止。

Partition-based methods

其原理简单来说就是，想象你有一堆散点需要聚类，想要的聚类效果就是" 类内的点都足够近，类间的点都足够远" 。首先你要确定这堆散点最后聚成几类，然后挑选几个点作为初始中心点，再然后依据预先定好的启发式算法 (heuristic algorithms) 给数据点点做迭代重置（iterative relocation), 直到最后到达" 类内的点都足够近, 类间的点都足够远" 的目标效果。也正是根据所渭的" 启发式算法", 形成了 k-means 算法及其变体包括 k-medoids 、k-modes 、k-medians、kernel k-means 等算法。k-means 对初始值的设置很敏感，所以有了 k-means 十十、intelligent k-means 、genetic k-means; k-means 对噪声和离群值非常敏感，所以有了 k-medoids 和 k-medians; k-means 只用于 numerical 类型数据，不适用于 categorical 类型数据，所以 k-modes; k-means 不能解决非凸 (non-convex) 数据，所以有了 kernel k-meanso 另外，很多教程都告诉我们 Partition-based methods 聚类多适用于中等体量的数据集，但我们也不知道中等' 到底有多' 中" ，所以不妨理解成，数据集越大，越有可能陷入局部最小。

- example k-means

1. 选择 K 个初始质心，初始质心随机选择即可，每一个质心为一个类。
2. 把每个观测指派到离它最近的质心，与质心形成新的类。
3. 重新计算每个类的质心，所谓质心就是一个类中的所有观测的平均向量（这里称为向量，是因为每一个观测都包含很多变量，所以我们把一个观测视为一个多维向量，维数由变量数决定)。
4. 重复 2. 和 3。
5. 直到质心不在发生变化时或者到达最大迭代次数时。

Density-based methods

k-means 解决不了不规则形状的聚类。于是就有了 Density-based methods 来系统解决这个问题。该方法同时也对噪声数据的处理比较好。其原理简单说画圈，其中要定义两个参数，一个是圈的最大半径，一个是圈里最少应容纳几个点。最后在一个圈里的，就是一个类。DBSCAN (Density-Based SpatiaI Clustering of Applications with Noise) 就是其中的典型，可惜参数设置也是个问题，对这两个参数的设置非常敏感。DBSCAN 的扩展叫 OPTICS (Ordering Points To ldentify Clustering Structure) 通过优先对高密度 (high density) 进行搜索，然后根据高密度的特点设置参数，改善了 DBSCAN 的不足。

- example

其核心思想是在数据空间中找到分散开的密集区域，简单来说就是画圈，其中要定义两个参数，一个是圈的最大半径，一个是一个圈里面最少应该容纳多少个点。

1. 从数据集中随机选择核心点。

2. 以核心点为圆心，做半径为 V 的圆，圆内圈入点的个数满足密度阈值的核心点称为核心对象，每一个核心对象的对应的圈都是一个簇。

3. 合并这些相互重合的簇。

Grid-based method

- example

根据网格的聚类其原理是将数据空间划分为网格单元，将数据对象映射到网格单元中，并计算每个单元的密度。根据预设阈值来判断每个网格单元是不是高密度单元，由邻近的稠密单元组成 “类”。

1. 将数据空间划分为网格单元。
2. 依照设置的阈值，判定网格单元是否稠密。
3. 合并相邻稠密的网格单元为一类。

Model-based methods

这一类方去主要是指基于概率模型的方法和基于神经网络模型的方法，尤其以基于概率模型的方法居多。这里的概率模型主要指概率生成模型 (generative Model)，同一" 类" 的数据属于同一种概率分布。这种方法的优点就是对" 类" 的划分不那么" 坚硬"，而是以概率形式表现，每一类的特征也可以用参数来表达；但缺点就是执行效率不高，特别是分布数量很多并且数据量很少的时候。其中最典型、也最常用的方法就是高斯混合模型 (GMM, Gaussian Mixture Models)。基于神经网络模型的方法主要就是指 SO(SeIfOrganized Maps) 了。

Data reduction

数据简化 (data reduction)，这个环节 optional。其实第二部分提到的有些算法就是对数据做了简化，才得以具备处理大规模数据的能力，比如 BIRCHO。但其实你可以任意组合，所以理论上把数据简化的方法和上面提到的十几种聚类算法结合使用，可以有上百个算法了。

- 变换 (Data Transformation)：离散傅里叶变换 (Discrete Fourier Transformation) 可以提取数据的频域 (frequency domain) 信息，离散小波变换 (Discrete Wavelet Transformation) 除了频域之外，还可以提取到时域 (temporal domain) 信息。
- 降维 (Dimensionality Reduction)：在降维的方法中，PCA (Principle Component Analysis) 和 SVD (Singular Value Decomposition) 作为线性方法，受到最广泛的应用。还有像 MDS (Multi-DimensionaI Scaling) 什么的，不过只是作为 PCA 的一个扩展，给我的感觉是中看不中用。这几个方法局限肯定是无法处理非线特征明显的数据。处理非线性降维的去主要是流形学习 (Manifold Learning)，这又是一大块内容，里面集中常见的算法包括 ISOMAP 、LLE (LocaIIy Linear Embedding) 、MVU (Maximum variance unfolding)Laplacian eigenmaps 、Hessian eigenmaps 、Kernel PCA 、ProbabiIistic PCA 等等。关于降维在聚类中的应用，最著名的应该就是谱聚类 (Spectral Clustering) 就是先用 LapIacian eigenmaps 对数据降维 (简单地说，就是先将数据转换成邻接矩阵或相似性矩阵，再转换成 LapIacian 矩阵，再对 LapIacian 矩阵进特征分解，把最小的 K 个特征向量列在一起)，然后再使用 k-means 完成聚类。谱聚类是个很好的方法，效果通常比 k-means 好，计算复杂度还低，这都要归功于降维的作用。
- 抽样 (Sampling)：最常用的就是随机抽样 (Random Sampling) ，如果你的数据集特别大，随机抽样就越能显示出它的低复杂性所带来的好处。比如 CLARA (CIustering LARge AppIications) 就是因为 k-medids 应对不了大规模的数据集，所以采用 sampling 的方法。

result asessment

对于聚类结果的评价方法一般可以分为内部评估法（internal evaluation）与外部评估方法（external evaluation）。

所谓外部评估方法是指在知道真实标签（ground truth ）的情况下来评估聚类结果的好坏，一般来说在做论文，或者是有少量的标注数据时，都可以用外部评估法选择一个相对最优的聚类模型，然后再应用到其它未被标记的数据中。

所谓内部评估法是不借助于外部信息，仅仅只是根据聚类结果来进行评估，常见的有轮廓系数（Silhouette Coefficient）、Calinski-Harabasz Index 等。一般来说，在完全没有标记数据的情况下可以通过这种方式来评估聚类结果的好坏。

外部评价指标

- 聚类纯度 (purity) 与准确率异曲同工，其总体思想就是用聚类正确的数目除以总的样本数，因此常被称为准确率。

$$P = \sum_k \frac{max\ groud\ truth\ number\ in\ cluster\ k}{total\ number}$$

- Rand Index

定义簇 (cluster，cluster result), 类 (classification, groud truth)。

precision, 你发现的阳性有多少是真的阳性。
Recall, 放出去的阳性，你找回了多少。

TP = 同簇同类的数目
TN = 不同簇不同类的数目
FP = 同簇不同类的数目
FN = 不同簇同类的数目

$$RI = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_\beta = \frac{(\beta^2 + 1)}{\beta^2} \frac{(Precision * Recall)}{Precision + Recall}$$

在这里 RI 和 F_β 的取值范围均为 $[0, 1]$, 越大表示聚类效果越好。一般用得较多的是 $F1$，这里 $\beta = 1$

- 调整兰德系数 (adjusted Rand Index)

$$ARI = \frac{\sum_i \sum_j \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_b \binom{b_j}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2} * \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}$$

考虑 A 在三个 cluster 中数量分别为 5, 1, 2. B 在三个 cluster 中数量分别为 1, 4, 0. C 在三个 cluster 中的数量分别为 0, 1, 3.

$$\sum_i \sum_j \binom{n_{ij}}{2} = \binom{5}{2} + \binom{2}{2} + \binom{4}{2} + \binom{3}{2} = 20$$

$$\sum_i \binom{a_i}{2} = \binom{8}{2} + \binom{5}{2} + \binom{4}{2} = 44$$

$$\sum_j \binom{b_j}{2} = \binom{6}{2} + \binom{6}{2} + \binom{5}{2} = 40$$

$$ARI = \frac{20 - 44 * 40/136}{0.5 * (44 + 40) - 44 * 40/136} = 0.24$$

内部评价指标

内部评价指标基本都基于簇内距离，与簇间距离的比值，只是这些距离的含义不同。

- 轮廓系数 (Sihouette Coefficient Index)

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

$$s = \frac{1}{n}\sum_{i}^{n} s_i$$

i 代表一个样本点，其中 $a(i)$ 是该样本点在簇类与其它点的均值距离，$b(i)$ 是该样本点与其最近的簇的样本点的距离均值，这里所谓的距离最近的簇是该点与其它簇的中心点的距离最近的簇。可以看出 s 的取值范围为 $[-1, 1]$

- Calinski-Harabasz Index(方差比准则)

$$W = \sum_{k=1}^{k} \omega_k = \sum_{k=1}^{K} \sum_{x \in C_k} (x - c_k)^2$$

$$B = \sum_{k=1}^{k} b_k = \sum_{k=1}^{k} n_k (c_k - c)^2$$

$$s = \frac{B}{K-1} \Big/ \frac{W}{n-K} = \frac{B}{W} \cdot \frac{n-K}{k-1}$$

其中 ω_k 是簇 k 中所有点与该簇中心点的距离和，b_k 是簇 k 的中心点与所有样本中心点距离乘以簇 k 的样本点数目。其中 c_k 为簇 k 的簇中心，c 为所有样本点的中心。K 为聚类得到的簇总数，n 为样本数目，n_k 为簇 k 的样本数目。

- Davies-Bouldin Index

簇内直径与簇间距离的比值。
首先定义簇内直径 s_i 等于簇 i 中所有点与簇中心点的距离均值，簇 i 与簇 j 之间的距离为 d_{ij}, 等于簇 i 与簇 j 中心点之间的距离。

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

$$DB = \frac{1}{K}\sum_{i,j=1}^{K} max_{i \neq j} R_{ij}$$

DB 指数的取值范围在 $[0, +\infty]$，结果越小聚类效果越好。