聚类算法学习总结

汤淳

15140220024

一、简要介绍

1、聚类概念

聚类就是按照某个特定标准(如距离准则)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，同时不在同一个簇中的数据对象的差异性也尽可能地大。即聚类后同一类的数据尽可能聚集到一起，不同数据尽量分离。

2、聚类和分类的区别

聚类技术通常又被称为无监督学习，因为与监督学习不同，在聚类中那些表示数据类别的分类或者分组信息是没有的。

Clustering (聚类)，简单地说就是把相似的东西分到一组，聚类的时候，我们并不关心某一类是什么，我们需要实现的目标只是把相似的东西聚到一起。因此，一个聚类算法通常只需要知道如何计算相似度就可以开始工作了，因此 clustering 通常并不需要使用训练数据进行学习，这在 Machine Learning 中被称作 unsupervised learning (无监督学习)。

Classification (分类)，对于一个 classifier，通常需要你告诉它“这个东西被分为某某类”这样一些例子，理想情况下，一个 classifier 会从它得到的训练集中进行“学习”，从而具备对未知数据进行分类的能力，这种提供训练数据的过程通常叫做 supervised learning (监督学习)。

3、衡量聚类算法优劣的标准

不同聚类算法有不同的优劣和不同的适用条件。大致上从跟数据的属性（是否序列输入、维度），算法模型的预设，模型的处理能力上看。具体如下：

1、算法的处理能力：处理大的数据集的能力（即算法复杂度）；处理数据噪声的能力；处理任意形状，包括有间隙的嵌套的数据的能力；

2、算法是否需要预设条件：是否需要预先知道聚类个数，是否需要用户给出领域知识；

3、算法的数据输入属性：算法处理的结果与数据输入的顺序是否相关，也就是说算法是否独立于数据输入顺序；算法处理有很多属性数据的能力，也就是对数据维数是否敏感，对数据的类型有无要求。

4、聚类算法有哪些类

被广泛运用的聚类算法有以下几类：

- 基于**连通模型**（connectivity-based）的聚类算法： 即**层次聚类**算法，其核心思想是按照对象之间的距离来聚类，两个离的近的对象要比两个离的远的对象更有可能属于同一 cluster。
- 基于**中心点模型**（centroid-based）的聚类算法： 在此类算法中，每个 cluster 都维持一个中心点（centorid），该中心点不一定属于给定的数据集。当预先指定聚类数目 k 时，此类算法需要解决一个优化问题，目标函数为所有的对象距其所属的 cluster 的中心点的距离的平方和，优化变量为每个 cluster 的中心点以及每个对象属于哪个 cluster；此优化问题被证明是 NP-hard 的，但有一些迭代算法可以找到近似解， k-means 算法 即是其中的一种。
- 基于**分布模型**（distribution-based）的聚类算法： 此类算法认为数据集中的数据是由一种混合概率模型所采样得到的，因而只要将可能属于同一概率分布所产生的数据归为同一 cluster 即可，最常被使用的此类算法为 高斯混合模型（GMM）聚类。
- 基于**密度**（density-based）的聚类算法： 在此类算法中，密度高的区域被归为一个 cluster，cluster 之间由密度低的区域隔开，密度低的区域中的点被认为是噪声（noise），常用的密度聚类算法为 DBSCAN 和 OPTICS。

二、算法介绍

1、基于**连通模型**（connectivity-based）的聚类算法

1.1 基本思想

层次聚类主要有两种类型：合并的层次聚类和分裂的层次聚类。前者是一种自底向上的层次聚类算法，从最底层开始，每一次通过合并最相似的聚类来形成上一层次中的聚类，整个当全部数据点都合并到一个聚类的时候停止或者达到某个终止条件而结束，大部分层次聚类都是采用这种方法处理。后者是采用自顶向下的方法，从一个包含全部数据点的聚类开始，然后把根节点分裂为一些子聚类，每个子聚类再递归地继续往下分裂，直到出现只包含一个数据点的单节点聚类出现，即每个聚类中仅包含一个数据点。

1.2 算法流程

以下流程以自下向上为例。

1. 将每个对象看作一类，计算两两之间的最小距离；

2. 将距离最小的两个类合并成一个新类；

3. 重新计算新类与所有类之间的距离；

4. 重复 2、3，直到所有类最后合并成一类

1.3 算法优缺点

优点：可解释性好（如当需要创建一种分类法时）；还有些研究表明这些算法能产生高质量的聚类，也会应用在上面说的先取 K 比较大的 K-means 后的合并阶段；还有对于 K-means 不能解决的非球形族就可以解决了。

缺点：时间复杂度高啊，o(m^3)，改进后的算法也有 o(m^2lgm)，m 为点的个数；贪心算法的缺点，一步错步步错；同 K-means，difficulty handling different sized clusters and convex shapes。

1.4 常见的算法及改进

该聚类算法因为计算复杂度比较大适用于小数量级，如对中国省会城市聚类。改进的算法有 BIRCH（Balanced Iterative 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。

Chameleon（A Hierarchical Clustering Algorithm Using Dynamic Modeling）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，但运算复杂还是很高，O(n^2)。

2、基于**中心点模型**（centroid-based）的聚类算法

2.1 基本思想

基于划分的方法：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。也正是根据所谓的“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。

2.2 算法流程

经典 K-means 算法流程：

1. 随机地选择 k 个对象，每个对象初始地代表了一个簇的中心；

2. 对剩余的每个对象，根据其与各簇中心的距离，将它赋给最近的簇；

3. 重新计算每个簇的平均值，更新为新的簇中心；

4. 不断重复 2、3，直到准则函数收敛。

2.3 算法优缺点

优点：对于大型数据集也是简单高效、时间复杂度、空间复杂度低。

缺点：最重要是数据集大时结果容易局部最优；需要预先设定 K 值，对最先的 K 个点选取很敏感；对噪声和离群值非常敏感；只用于 numerical 类型数据；不能解决非凸（non-convex）数据。

2.4 常见的算法及改进

k-means 对初始值的设置很敏感，所以有了 k-means++、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-means。

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。

3、基于模型的方法（Model-based methods）

3.1 基本思想

基于模型的方法：为每簇假定了一个模型，寻找数据对给定模型的最佳拟合，这一类方法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，即假设数据是根据潜在的概率分布生成的。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 SOM（Self Organized Maps）了，也是我所知的唯一一个非监督学习的神经网络了。

3.2 算法流程

【以 SOM 为例】SOM 神经网络是由芬兰神经网络专家 Kohonen 教授提出的，该算法假设在输入对象中存在一些拓扑结构或顺序，可以实现从输入空间(n 维)到输出平面(2 维)的降维映射，其映射具有拓扑特征保持性质,与实际的大脑处理有很强的理论联系。

SOM 网络包含输入层和输出层。输入层对应一个高维的输入向量，输出层由一系列组织在 2 维网格上的有序节点构成，输入节点与输出节点通过权重向量连接。学习过程中，找到与之距离最短的输出层单元，即获胜单元，对其更新。同时，将邻近区域的权值更新，使输出节点保持输入向量的拓扑特征。

算法流程：

1、 网络初始化，对输出层每个节点权重赋初值；

2、 将输入样本中随机选取输入向量，找到与输入向量距离最小的权重向量；

3、定义获胜单元，在获胜单元的邻近区域调整权重使其向输入向量靠拢；

4、 提供新样本、进行训练；

5、收缩邻域半径、减小学习率、重复，直到小于允许值，输出聚类结果。

3.3 算法优缺点

优点：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。

缺点：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。

3.4 常见的算法及改进

基于概率模型的最典型、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 SOM（Self Organized Maps）了，也是我所知的唯一一个非监督学习的神经网络了。

4、基于密度的方法(Density-based methods)

4.1 基本思想

基于密度的方法：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。其原理简单说画圈儿，其中要定义两个参数，一个是圈儿的最大半径，一个是一个圈儿里最少应容纳几个点。只要邻近区域的密度（对象或数据点的数目）超过某个阈值，就继续聚类,最后在一个圈里的，就是一个类。DBSCAN（Density-Based Spatial Clustering of Applications with Noise）就是其中的典型.

4.2 算法流程

DBSCAN 流程：

1. 从任一对象点 p 开始；

2. 寻找并合并核心 p 对象直接密度可达（eps）的对象；

3. 如果 p 是一个核心点，则找到了一个聚类，如果 p 是一个边界点（即从 p 没有密度可达的点）则寻找下一个对象点；

4. 重复 2、3，直到所有点都被处理

DBSCAN 聚类算法原理的基本要点：确定半径 eps 的值

①DBSCAN 算法需要选择一种距离度量，对于待聚类的数据集中，任意两个点之间的距离，反映了点之间的密度，说明了点与点是否能够聚到同一类中。由于 DBSCAN 算法对高维数据定义密度很困难，所以对于二维空间中的点，可以使用欧几里德距离来进行度量。

②DBSCAN 算法需要用户输入 2 个参数：一个是半径（Eps），表示以给定点 P 为中心的圆形邻域的范围；另一个是以点 P 为中心的邻域内最少点的数量（MinPts）。如果满足：以点 P 为中心、半径为 Eps 的邻域内的点的个数不少于 MinPts，则称点 P 为核心点。

③DBSCAN 聚类使用到一个 k-距离的概念，k-距离是指：给定数据集 P={p(i);i=0,1,…n}，对于任意点 P(i)，计算点 P(i)到集合 D 的子集 S={p(1)，p(2)，…，p(i-1)，p(i+1)，…，p(n)}中所有点之间的距离，距离按照从小到大的顺序排序，假设排序后的距离集合为 D={d(1)，d(2)，…，d(k-1)，d(k)，d(k+1),…,d(n)}，则 d(k)就被称为 k-距离。也就是说，k-距离是点 p(i)到所有点（除了 p(i)点）之间距离第 k 近的距离。对待聚类集合中每个点 p(i)都计算 k-距离，最后得到所有点的 k-距离集合 E={e(1)，e(2)，…，e(n)}。

④根据经验计算半径 Eps：根据得到的所有点的 k-距离集合 E，对集合 E 进行升序排序后得到 k-距离集合 E’，需要拟合一条排序后的 E’集合中 k-距离的变化曲线图，然后绘出曲线，通过观察，将急剧发生变化的位置所对应的 k-距离的值，确定为半径 Eps 的值。

⑤根据经验计算最少点的数量 MinPts：确定 MinPts 的大小，实际上也是确定 k-距离中 k 的值，DBSCAN 算法取 k=4，则 MinPts=4。

⑥如果对经验值聚类的结果不满意，可以适当调整 Eps 和 MinPts 的值，经过多次迭代计算对比，选择最合适的参数值。可以看出，如果 MinPts 不变，Eps 取得值过大，会导致大多数点都聚到同一个簇中，Eps 过小，会导致一个簇的分裂；如果 Eps 不变，MinPts 的值取得过大，会导致同一个簇中点被标记为噪声点，MinPts 过小，会导致发现大量的核心点。

我们需要知道的是，DBSCAN 算法，需要输入 2 个参数，这两个参数的计算都来自经验知识。半径 Eps 的计算依赖于计算 k-距离，DBSCAN 取 k=4，也就是设置 MinPts=4，然后需要根据 k-距离曲线，根据经验观察找到合适的半径 Eps 的值。

4.3 算法优缺点

优点：对噪声不敏感；能发现任意形状的聚类。

缺点：聚类的结果与参数有很大的关系；DBSCAN 用固定参数识别聚类，但当聚类的稀疏程度不同时，相同的判定标准可能会破坏聚类的自然结构，即较稀的聚类会被划分为多个类或密度较大且离得较近的类会被合并成一个聚类。

4.4 常见的算法及改进

DBSCAN 对这两个参数的设置非常敏感。DBSCAN 的扩展叫 OPTICS（Ordering Points To Identify 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。

5、基于网络的方法（Grid-based methods）

5.1 基本思想

基于网络的方法：这类方法的原理就是将数据空间划分为网格单元，将数据对象集映射到网格单元中，并计算每个单元的密度。根据预设的阈值判断每个网格单元是否为高密度单元，由邻近的稠密单元组形成”类“。

5.2 算法流程

这些算法用不同的网格划分方法，将数据空间划分成为有限个单元（cell）的网格结构,并对网格数据结构进行了不同的处理，但核心步骤是相同的：

1、 划分网格

2、 使用网格单元内数据的统计信息对数据进行压缩表达

3、 基于这些统计信息判断高密度网格单元

4、 最后将相连的高密度网格单元识别为簇

5.3 算法优缺点

优点：速度很快，因为其速度与数据对象的个数无关，而只依赖于数据空间中每个维上单元的个数。

缺点：参数敏感、无法处理不规则分布的数据、维数灾难等；这种算法效率的提高是以聚类结果的精确性为代价的。经常与基于密度的算法结合使用。

5.4 常见的算法及改进

STING（STatistical INformation Grid）算法、WAVE-CLUSTER 算法和 CLIQUE（CLustering In QUEst）是该类方法中的代表性算法。

6、基于模糊的聚类（FCM 模糊聚类）

6.1 基本思想

1965 年美国加州大学柏克莱分校的扎德教授第一次提出了‘集合’的概念。经过十多年的发展，模糊集合理论渐渐被应用到各个实际应用方面。为克服非此即彼的分类缺点，出现了以模糊集合论为数学基础的聚类分析。用模糊数学的方法进行聚类分析，就是模糊聚类分析。

基于模糊集理论的聚类方法，样本以一定的概率属于某个类。比较典型的有基于目标函数的模糊聚类方法、基于相似性关系和模糊关系的方法、基于模糊等价关系的传递闭包方法、基于模 糊图论的最小支撑树方法，以及基于数据集的凸分解、动态规划和难以辨别关系等方法。FCM 算法是一种以隶属度来确定每个数据点属于某个聚类程度的算法。该聚类算法是传统硬聚类算法的一种改进。

6.2 算法流程

FCM 模糊聚类算法流程：

1、 标准化数据矩阵；

2、 建立模糊相似矩阵，初始化隶属矩阵；

3、 算法开始迭代，直到目标函数收敛到极小值；

4、 根据迭代结果，由最后的隶属矩阵确定数据所属的类，显示最后的聚类结果。

FCM 算法需要两个参数一个是聚类数目 C，另一个是参数 m。一般来讲 C 要远远小于聚类样本的总个数，同时要保证 C>1。对于 m，它是一个控制算法的柔性的参数，如果 m 过大，则聚类效果会很次，而如果 m 过小则算法会接近 HCM 聚类算法。

算法的输出是 C 个聚类中心点向量和 C*N 的一个模糊划分矩阵，这个矩阵表示的是每个样本点属于每个类的隶属度。根据这个划分矩阵按照模糊集合中的最大隶属原则就能够确定每个样本点归为哪个类。聚类中心表示的是每个类的平均特征，可以认为是这个类的代表点。

6.3 算法优缺点

优点：从算法的推导过程中我们不难看出，算法对于满足正态分布的数据聚类效果会很好，另外，算法对孤立点是敏感的。

缺点：由于不能确保 FCM 收敛于一个最优解。算法的性能依赖于初始聚类中心。因此，我们要么用另外的快速算法确定初始聚类中心，要么每次用不同的初始聚类中心启动该算法，多次运行 FCM。

6.4 常见的算法及改进

模糊 C 均值（简称 FCM）聚类算法是 HCM 聚类算法的改进。

7、其他聚类

除此以外还有一些其他新的发展方法，在此只简要介绍。

7.1 基于约束的方法

真实世界中的聚类问题往往是具备多种约束条件的 ， 然而由于在处理过程中不能准确表达相应的约束条件、不能很好地利用约束知识进行推理以及不能有效利用动态的约束条件 ， 使得这一方法无法得到广泛的推广和应用。这里的约束可以是对个体对象的约束 ， 也可以是对聚类参数的约束 ， 它们均来自相关领域的经验知识。该方法的一个重要应用在于对存在障碍数据的二维空间数据进行聚类。 COD (Clustering with Ob2structed Distance) 就是处理这类问题的典型算法 ，其主要思想是用两点之间的障碍距离取代了一般的欧氏距离来计算其间的最小距离。

7.2 量子聚类：

受物理学中量子机理和特性启发，可以用量子理论解决聚类记过依赖于初值和需要指定类别数的问题。一个很好的例子就是基于相关点的 Pott 自旋和统计机理提出的量子聚类模型。它把聚类问题看做一个物理系统。并且许多算例表明，对于传统聚类算法无能为力的几种聚类问题，该算法都得到了比较满意的结果。

7.3 核聚类

核聚类方法增加了对样本特征的优化过程，利用 Mercer 核 把输入空间的样本映射到高维特征空间，并在特征空间中进行聚类。核聚类方法是普适的，并在性能上优于经典的聚类算法，它通过非线性映射能够较好地分辨、提 取并放大有用的特征，从而实现更为准确的聚类；同时，算法的收敛速度也较快。在经典聚类算法失效的情况下，核聚类算法仍能够得到正确的聚类。代表算法有 SVDD 算法，SVC 算法。

7.4 谱聚类

首先根据给定的样本数据集定义一个描述成对数据点相似度的亲合矩阵,并计算矩阵的特征值和特征向量,然后选择合适的特征向量聚类不同的数据点。谱聚类算法最初用于计算机视觉、VLSI 设计等领域,最近才开始用于机器学习中,并迅速成为国际上机器学习领域的研究热点。

谱聚类算法建立在图论中的谱图理论基础上,其本质是将聚类问题转化为图的最优划分问题,是一种点对聚类算法。