**聚类算法综述**

聚类是数据挖掘的重要工具， 根据数据间的相似性将数据库分成多个类， 每类中数据应尽可能相似。 从机器学习的观点来看， 类相当于隐藏模式， 寻找类是无监督学习过程。目前己有应用于统计、模式识别、机器学习等不同领域的几十种聚类算法。该文对数据挖掘中的聚类算法进行了归纳和分类，总结了几类算法并分析了其性能特点。

关键词：聚类算法;相似性度量;K-means;EM;

Abstract

Clustering plays an outstanding role in data mining applications.Clustering is a division of databases into groups of similar objects based on the similarity.From amachine learning perspective clusters correspond to hidden patterns,the search for clusters is unsupervised learning.There are tens of clustering algorithms used in various fields such as statistics,pattern recognition and machine learning now.This paper concludes the clustering algorithms used in data mining and assorts them into many classes.Each types of algorithms are summarized and their performances are analyzed here.

KeyWords: clustering algorithm; similarity measurement;K-means;EM;

**1、引言**

聚类就是将物理或抽象对象的集合分组成为由类似的对象组成的多个类的过程。由聚类所生成的簇是一组数据对象的集合，这些对象与同一簇中的对象彼此相似，与其它簇中的对象相异。聚类分析是一种无监督的过程，它同分类的根本区别在于：分类是需要开始知道所根据的特征，而聚类是要准确的找到这个数据特征，因此在许多的应用中，聚类分析更是定义为一种数据预处理的过程，是进一步解析和处理数据的根本。它已经被广泛地应用于统计学、机器学、空间数据库、生物学以及市场营销等领域，聚类分析还可以作为独立的数据挖掘工具来了解数据分布，或者作为其他数据挖掘算法（如关联规则、分类等）的预处理步骤。聚类算法可以分为基于的层次方法、基于划分的方法、基于网格的方法、基于密度的方法和基于模型的方法。

**2、数据的相似性度量**

聚类分析按照样本点之间的亲疏远近程度进行分类。为了使类分得合理，必须描述样本之间的亲疏远近程度。刻画聚类样本点之间的亲疏远近程度主要的方法是利用距离度量的方法，常用的距离度量方法有欧几里德距离、余弦距离和马氏距离等，下面分别对这几种距离进行了阐述。给定数据集Z={},其中是维特征空间中的一个特征向量，而是特征空间Z中特征向量的个数。[1]

2.1、欧几里德距离

欧几里德距离又叫欧氏距离，它的定义为：

d(,) = =|| 公式(1)

欧氏距离即两项间的差是每个变量值差的平方和再平方根，目的是计算其间的整体距离即不相似性。欧氏距离虽然简单而且最常用，但是它有一个缺点是它将样本的不同属性（即各指标或各变量）之间的差别等同看待。这一点在很多的应用中都不能满足要求。

2．2 余弦距离

余弦距离定义为如下：

＜＞＝ 公式(2)

其中<>[-1,1]。余弦距离是通过测量两个向量内积空间的夹角的余弦值来度量它们之间的相似性。余弦距离可以用在任何维度的向量比较中，它尤其在高维正空间中的利用尤为频繁。它通常用于文本挖掘中的文件比较。此外，在数据挖掘领域中，用它来衡量集群内部的凝聚力。

2.2 马氏距离

马氏距离是由印度统计学家马哈拉诺比斯提出的，它的定义为如公式（3）所示。

()=() 公式(3)

它是一种有效的计算两个未知样本集的相似度的方法。与欧氏距离不同的是它考虑到各种特性之间的联系,并且是尺度无关的（scale-invariant），即独立于测量尺度。

**3、聚类方法**

目前聚类算法基本上是基于层次的聚类方法和基于划分的聚类方法。下面对分别这两种方法进行介绍。

3.1 基于层次的聚类方法

层次聚类算法，它是通过将数据组织为若干组并形成一个相应的树来进行聚类的。根据层次是自底向上还是自顶而下形成，层次聚类算法可以进一步分为凝聚型的聚类算法和分裂型的聚类算法。一个完全层次聚类的质量由于无法对已经做的合并或分解进行调整而受到影响。但是层次聚类算法没有使用准则函数，它所含的对数据结构的假设更少，所以它的通用性更强[2,3]。

3.1.1两种基本的层次聚类方法

凝聚的层次聚类是将这种自底向上的策略首先将每个对象作为一个簇，然后合并这些原子簇为越来越大的簇，直到所有的对象都在一个簇中，或者某个终结条件被达到要求。大部分的层次聚类方法都属于一类，它们在簇间的相似度的定义有点不一样。

分裂的层次聚类是将像这样的自顶向下的策略与凝聚的层次聚类有些不一样，它首先将所有对象放在一个簇中，然后慢慢地细分为越来越小的簇，直到每个对象自行形成一簇，或者直达满足其他的一个终结条件，例如满足了某个期望的簇数目，又或者两个最近的簇之间的距离达到了某一个阈值。

3.1.2基于距离度量的方法

在凝聚和分裂的层次聚类之间，我们又依据计算簇间的距离的不同，分为下面的几类方法：

1. 单连锁（single linkage）,又称最近邻（nearest neighbor）方法。指两个不一样的簇之间任意两点之间的最近距离。这里的距离是表示两点之间的相异度，所以距离越近，两个簇相似度越大。这种方法最善于处理非椭圆结构。却对于噪声和孤立点特别的敏感，取出距离很远的两个类之中出现一个孤立点时，这个点就很有可能把两类合并在一起。距离公式如公式4所示。

 （公式4）

1. 全连锁（comlpete linkage），又称最远邻（furthest neighbor）方法。指两个不一样的簇中任意的两点之间的最远的距离。它面对噪声和孤立点很不敏感，趋向于寻求某一些紧凑的分类，但是，有可能使比较大的簇破裂。距离公式如公式5所示。

 (公式5)

1. 组平均方法(grou Paverage linkage)，定义距离为数据两两距离的平均值。这个方法倾向于合并差异小的两个类，产生的聚类具有相对的鲁棒性。距离公式如公式6所示。

 （公式6）

1. 平均值方法(centroid linkage),现计算各个类的平均值，然后定义平均值之差为两类的距离。距离公式如公式3.4所示。

 （公式7）

其中,是两个类，|为对象p和之间的距离，，分别为,的对象个数，，分别为类,的平均值。

3.1.4 层次聚类方法的优缺点

一般地,基于层次的聚类方法具有以下几个优点：

(1)对初绐数据集不敏感

(2)能很好的处理孤立点和“噪声”数据

(3)不需要指定簇的个数

基于层次的聚类方法的缺点有：

(1)它的复杂度高，时间复杂度为：O()，空间复杂度为：O()，所以，基于层次的聚类方法不适应于大数据集。

(2)一旦一个合并或分裂被执行，就不能修正。

(3)重叠的点往往很很难决定要合并或者分裂，这样会使聚类结果有所偏差。

3.2 基于划分的聚类方法

给定一个数据库包含个数据对象以及数目K的即将生成的簇,一个划分类的算法将对象分为K个划分，其中，这里的每个划分分别代表一个簇，并且K<=。其中的K需要人为指定。它一般从一个初始划分开始，然后通过重复的控制策略，使某个准则函数最优化。因此，它可以被看作是一个优化问题，而优化问题往往是NP-难问题。基于划分的聚类方法的优缺点跟层次的聚类方法的优缺点刚刚好反，层次聚类算法的优点恰恰是划分聚类方法的缺点，反之亦然。根据它们之间的优缺点，人们往往会更趋向于使用划分的聚类方法，所以，本文着重于讲解基于划分的聚类方法。

基于划分的聚类方法常常是迭代而且是局部收敛的，根据Hamerly和Elkan[4]的文章，基于划分的聚类方法的迭代步骤为：

(1)随机选择K个聚类中心。

(2)重复3、4步，直至满足一定条件。

(3)对于每一个数据点，计算它到每个中心的隶属函数和权重，根据一定的准则函数把每个数据点分配到相应的簇。

(4)重新计算每个簇的中心点：

= (公式8)

其中隶属函数是表示一个数据点隶属于簇k的程度的函数。硬聚类算法用硬的隶属函数，而模糊的聚类算法使用软隶属函数[4]。本算法中是用来确定数据点属于簇k中的程度的函数。它必须满足的条件是：

1. >=0,p=1，…，,k = 1,…,K (公式9)
2. = 1,p=1，…， (公式10)

其中权重是用来确定数据点在下一次迭代中重新计算中心点时所占的比例，而>0。不同的终止条件会用在不同的聚类算法中，例如：

(1)迭代次数超出一定次数。

(2)每个簇的均值的变化小于一定阈值。

(3)量化误差足够小。

基于划分的聚类算法有许多，下面介绍几中常见的基于划分的聚类算法。

3.2.1 K-means

k-means算法是最为经典的基于划分的聚类方法，是十大经典数据挖掘算法之一。k-means算法的基本思想是：以空间中k个点为中心进行聚类，对最靠近他们的对象归类。通过迭代的方法，逐次更新各聚类中心的值，直至得到最好的聚类结果。该算法的迭代的终止条件是直至中心点收敛。因此，K-means算法需要优化的目标函数是：

(公式11)

其中隶属函数和权重分别定义为：

= (公式12)

=1 (公式13)

所以，该算法的隶属函数是一个硬隶属函数，而它的权重是一个恒定的常数，即每个数据点同等重要。该算法的最大优势在于简洁和快速，对于处理大数据集，该算法是相对可伸缩和高效率的。算法的缺点是簇的数目必须人为的指定，并且对初值敏感，对于不同的初值，可能会导致不同结果，而且不适合于发现非凸面形状的簇或者大小 差别很大的簇，它对于“躁声”和孤立点数据是敏感的。该算法的关键在于初始中心的选择和距离公式。

而K-medoids算法跟k-means算法的不同之处在于K-medoids用接近聚类中心的一个对象来表示每个簇而K-means用簇中对象的平均值来表示每个簇。K-medoids算法的时间复杂度为O()，所以，它不能应用于大数据集。

3.2.2 模糊C均值算法

模糊聚类分析作为无监督机器学习的主要技术之一，是用模糊理论对重要数据分析和建模的方法，建立了样本类属的不确定性描述，能比较客观地反映现实世界，它已经有效地应用在大规模数据分析、数据挖掘、矢量量化、图像分割、模式识别等领域，具有重要的理论与实际应用价值，随着应用的深入发展，模糊聚类算法的研究不断丰富。在众多模糊聚类算法中，模糊C-均值（ FCM） 算法应用最广泛且较成功，它通过优化目标函数得到每个样本点对所有类中心的隶属度，从而决定样本点的类属以达到自动对样本数据进行分类的目的。它的思想就是使得被划分到同一簇的对象之间相似度最大，而不同簇之间的相似度最小。模糊C均值算法是普通C均值算法的改进，普通C均值算法对于数据的划分是硬性的，而FCM则是一种柔性的模糊划分。FCM的目标函数为：

(公式14)

其中q1是一个加权指数,是指第p个数据点在第k个簇中的隶属度，它必须满足：,p=1,…,,k=1,…,K，，FCM的隶属函数和权重定义为：

= (公式15)

=1 (公式16)

所以，该算法的隶属函数是一个软隶属函数，即一个数据点可能属于多个簇，而它的权重是一个恒定的常数。模糊C均值算法可以很好的处理临界数据点，它和K-means算法一样，需要人为的指定簇的数目。该算法的缺点是可能会产生重合聚类，需要人为指定簇的个数，收敛于局部最优以及初始条件对聚类结果影响很大。

3.2.3 EM算法

EM（Expectation-maximization algorithm） 算法是在概率模型中寻找参数最大似然估计或者最大后验估计算法，其中概率模型依赖于无法观测的隐藏变量。最大期望算法经过两个步骤交替进行计算，第一步是计算期望（E），利用对隐藏变量的现有估计值，计算其最大似然估计值；第二步是最大化（M），最大化在 E 步上求得的最大似然值来计算参数的值。M 步上找到的参数估计值被用于下一个 E 步计算中，这个过程不断交替进行。EM算法的目标函数为

=- (公式17)

其中是高斯分布基于质心产生的的概率，是先验概率。EM算法的隶属函数和权重分别为：

= (公式18)

=1 (公式19)

所以EM是一个软隶属函数而且权重是一个常数。EM算法的特点有它会收敛到局部极值，但不保证收敛到全局最优。而且对初值很敏感，通常需要一个好的、快速的初始化过程。EM算法适应于缺失数据不太多以及数据高时，因为如果数据集的维数太高的话，E步的计算很费时。最大期望经常用在机器学习和计算机视觉的数据聚类领域。

3.2.4 KHM算法

KHM(The K-harmonic means algorithm)算法[5]是基于中心的迭代过程 ,它采用所有数据点到每个聚类中心的和平均值的和作为目标函数.目标函数为

(公式20)

其中是模糊指数，是一个用户指定的参数，通常.KHM算法的隶属函数和权重分别为：

= (公式21)

(公式22)

所以，KHM算法的权重是个变量，它把距离中心点远的数据点赋以高的权重，这样可以让质心能够很好的覆盖整个数据集。KHM算法对初始值不敏感，适合处理大数据集，然而KHM算法容易陷入局部最优及簇个数需要预先指定的问题。综合上来说它胜过K-means、FCM和EM算法。

3.3 非迭代的划分的聚类方法

另外的一种聚类算法就是非迭代的划分的聚类方法，最常用的非迭代的算法是MacQueen 的K-means算法[6]。该算法的思想是，给定一个数据集，找到指定数量的聚类中心，然后把数据集聚类到相应的簇。该算法对初始值敏感，为了解决这个问题，可以打乱数据集中数据点的顺序。一般情况下来说，迭代的算法要比非迭代的算法要高效的多。

3.4 其它的聚类方法

3.4.1最近邻聚类算法

最近邻聚类算法（Nearest Neighbor clustering algorithm）是一个用于处理多密度数据集的聚类算法，其主要思想可概括为：对于数据集中每个点，找出距离其最近的K个邻近点，形成一个集合。然后考虑数据集中的任意两个点，若对应于这两个对的K个邻近点集合交集部分的点数超过一个阈值，则将这两个点归于一类。SNN算法的优点是可以对不同密度和形状的数据集进行聚类，能处理高维数据集和自动识别簇的数目。缺点是在多密度聚类和处理孤立点或噪声方面SNN算法精度都不高，并且该算法对参数是敏感的。

3.4.2 谱聚类

谱聚类算法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,并且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 、VLS I 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，与传统的聚类算法相比，它具有能在任意形状的样本空间上聚类且收敛于全局最优解的优点。

3.4.3 MeanShift算法

MeanShift (均值漂移)是一种非参数概率密度估计的方法，一种最优的寻找概率密度极大值的梯度上升法，在解决计算机视觉底层过程中表现出了良好的鲁棒性和较高的处理速度。MeanShift 算法一般指的是一个迭代的步骤，即先算出当前点的漂移均值，移动该点到其漂移均值，然后以此为新的起始点，继续移动，直到满足一定的条件结束。

3.5各类算法比较

基于上述的分析，下面对常用聚类算法的性能从可伸缩性、发现聚类的形状、对“噪声”的敏感性、对数据输入顺序的敏感性、高维性和是否是硬聚类六个方面进行比较，如表1所示。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 算法 | 可伸缩性 | 发现聚类的形状 | 对“噪声”的敏感性 | 对数据输入顺序的敏感性 | 高维性 | 硬聚类 |
| K-means | 不好 | 凸形 | 敏感 | 敏感 | 不好 | 是 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| FCM | 好 | 任意形状 | 敏感 | 不敏感 | 好 | 否 |
| EM | 不好 | 任意形状 | 敏感 | 不敏感 | 不好 | 否 |
| KHM | 好 | 任意形状 | 敏感 | 不敏感 | 不好 | 否 |
| SNN | 好 | 任意形状 | 敏感 | 不敏感 | 好 | 是 |

**表1：各类算法比较**

**4、总结**

聚类分析是数据挖掘中一种非常有用的技术，它可作为特征和分类算法的预处理步骤，这些算法再在生成的簇上进行处理，也可将聚类结果用于进一步关联分析。还可以作为一个独立的工具来获得数据分布的情况，观察每个簇的特点，集中对特定的某些簇做进一步分析。其应用范围涉及商务，生物，地理，Ｗｅｂ文档分类，仿真等诸多领域。

聚类能更好地应用到现实生活中是很必要的。这些新算法正努力把静态的聚类推向动态的、适应性强的、带约束条件的及与生活联系紧密的聚类。同时，对目前可有效处理二维和小的数据集的聚类方法进行强化和修改，以使其能处理大的和高维的数据，这也是努力的一个方向。

参考文献

[1]G.H.Omran,P.Engelbrecht and salman.Intelligent Data Analysis:An overview of clustering methods.2007,583-605.

[2] J.Han,M.Kamber著.范明,孟小峰等译.数据挖掘概念与技术[M].北京：机械工业出版社,2001:196~220.

[3] 段明秀.层次聚类算法的研究及应用[D]:[硕士学位论文].长沙：中南大学,2009.

[4]G.Hamerly and C.Elkan,CIKM-2002:Alternatives to the K-means Algorithm that Find Better Clusterings.2002,600-607.

[5]B.zhang,Generalized K-Harmonic Means-Boosting in Unsupervised Learning.Technical Report HPL-2000-137.Hewlett-Packard Labs,2000.

[6] J.MacQueen, SomeMethods for Classiﬁcation and Analysis ofMultivariate Observations. In Proceedings Fifth Berkeley Symposium on Mathematics, Statistics and Probability, vol. 1, 1967, 281–297.