第5章

聚类分析模型及应用

【本章内容】 本章首先给出聚类分析的概念和分类，然后详细描述聚类算法的性能度量，最后给出聚类中常用的 K-Means 算法、DBSCAN 算法、AGNES 算法、高斯混合聚类算法、LVQ、CLIQUE 网格聚类的算法原理、计算流程、实例分析和 Python 实现。

【学习要求】 理解聚类分析的概念和分类；理解聚类算法的性能度量；掌握聚类中常用的 K-Means 算法、DBSCAN 算法、AGNES 算法、高斯混合聚类算法、LVQ、CLIQUE 网格聚类的算法原理、计算流程、实例分析和 Python 实现。


聚类与分类的区别

5.1 聚类概述

5.1.1 聚类的定义和分类

1. 聚类的概念、目标与定义的解释

俗话说："物以类聚，人以群分"，这就是聚类的概念。研究和学习聚类的目标在于：把相似的东西聚在一起；换句话说，聚类的目标就是**在相似的基础上收集数据来分类，以形成不同的类或者簇**。聚类算法与分析模型及应用是大数据的重要研究内容。

那么什么是聚类的定义呢？在自然科学和社会科学中，存在着大量的聚类定义问题。在"无监督学习"(unsupervised learning)中，训练样本的标记信息是未知的，目标是通过对无标记训练样本的学习来揭示数据的内在性质及规律，为进一步的数据分析提供基础。此类学习任务中研究最多、应用最广的就是"聚类"(clustering)。聚类试图将数据集中的样本划分为若干个通常是不相交的子集，每个子集称为一个"簇"(cluster)。通过这样的划分，每个簇可能对应于一些潜在的概念(类别)，如"浅色瓜""深色瓜""有籽瓜""无籽瓜"，甚至"本地瓜""外地瓜"等。

聚类分析起源于分类学，在古老的分类学中，人们主要依靠经验和专业知识来实现分类，很少利用数学工具进行定量的分类。随着人类科学技术的发展，人们对分类的要求越来越高，以致有时仅凭经验和专业知识难以确切地进行分类，于是人们逐渐地把数学工具引用到了分类学中，形成了数值分类学，之后又将多元分析的技术引入数值分类学形成了聚类分析。

自 Everitt 给出聚类定义以来，不少学者投身于聚类分析的研究当中，提出了

不少聚类分析的原理和算法。聚类分析作为热门的研究领域，涉及数据挖掘、模式识别、机器学习、数据分析等众多学科，Everitt 于 1974 年对聚类分析定义如下：旨在将样本按其自身的属性聚成若干类，以保证类内的样本相似度尽可能高，而类间的样本相似度尽可能低。也有学者将聚类分析称为无监督的分类，因为其在进行聚类之前没有任何的先验信息可以使用，并且聚类将样本聚成几类也是未知的。从定义上讲，聚类就是针对大量数据或者样品，根据数据本身的特性研究分类方法，并遵循这个分类方法对数据进行合理的分类，最终将相似数据分为一组，也就是“同类相同、异类相异”。图 5.1 给出了聚类示意图。

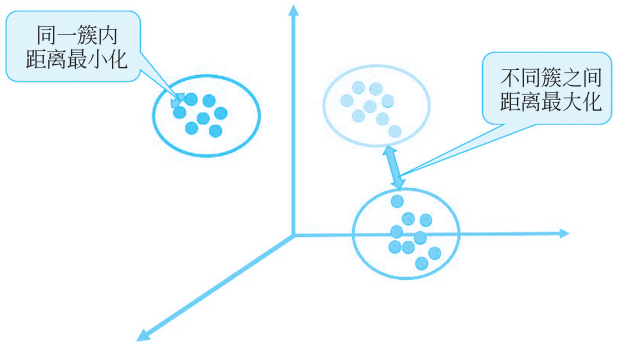

图 5.1　聚类示意图

按照聚类的目标得知：在聚类过程中，通过无标记训练样本的学习来揭示数据的内在性质及规律，是无监督学习过程。在无监督学习中，训练样本标记信息是未知的，聚类试图将数据集中的样本划分为若干个通常不相交的子集，每个子集称为一个“簇”，每个簇可能对应于一些潜在的类别，这些类别概念对聚类算法而言事先是未知的，聚类过程仅能自动形成簇结构，簇所对应的概念语义需要由使用者来把握和命名。

2. 典型的聚类分析应用内容的解释

典型的聚类分析应用内容：① 如果把人和其他动物放在一起比较，可以很轻松地找到一些判断特征，如肢体、嘴巴、耳朵、皮毛等，根据判断指标之间的差距大小划分出某一类为人，某一类为狗，某一类为鱼等，这就是聚类。② 通过对特定运营目的和商业目的所挑选出的指标变量进行聚类分析，把目标群体划分成几个具有明显特征区别的细分群体，把付费用户按照几个特定维度，如利润贡献、用户年龄、续费次数等聚类分析后得到不同特征的群体，将客户分为高价值成熟客户、活跃型老客户、非活跃型高价值老客户等细分客户群体，从而可以在运营活动中为这些细分群体采取精细化、个性化的运营和服务，最终提升运营的效率和商业效果。③ 学生网课学习产生的线上学习数据，通过对学生的视频观看时长、讨论数、访问数等学习特征进行聚类，能够将学生聚成具有不同特征的学生类群，将学生分为积极主动型、被动学习型、按部就班型等学生群体，这样就可以根据学生的学习情况，进行教学预警或者针对性的层次性教学，进而以学生为中心提升教学质量。

目前，聚类分析内容非常丰富，聚类分析算法的分类可以分为以下几大类：分裂法（Partitioning Methods）、基于密度（Density-Based Methods）的方法、层次法（Hierarchical Methods）、基于网格（Grid-Based Methods）的方法和基于模型（Model-Based Methods）的方法等。

3. 聚类算法分类体系结构

没有任何一种聚类技术(聚类算法)可以普遍适用于揭示各种多维数据集所呈现出来的多种多样的结构。根据数据在聚类中的积聚规则以及应用这些规则的方法,有多种聚类算法。聚类算法分类体系结构如图 5.2 所示。

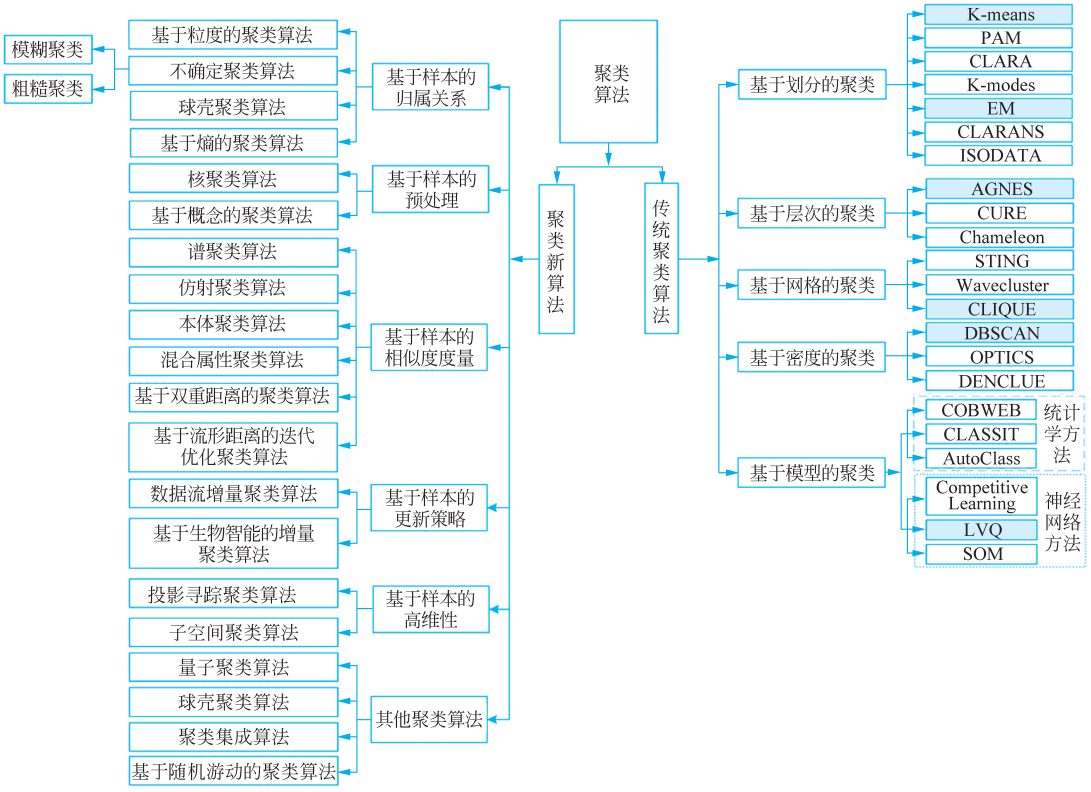

图 5.2 聚类算法分类体系结构

依据图 5.2,其基于划分聚类的思想是:每一个分组至少包含一个数据记录;每一个数据记录属于且仅属于一个分组(注意:这个要求在某些模糊聚类算法中可以放宽);对于给定的 K,算法首先给出一个初始的分组方法,以后通过反复迭代的方法改变分组,使得每一次改进之后的分组方案都较前一次好。使用这个基本思想的算法有:K-Means 算法、K-Medoids 算法、CLARANS 算法。基于密度的方法与其他方法的一个根本区别是:它不是基于各种各样的距离的,而是基于密度的,这样就能克服基于距离的算法只能发现“类圆形”的聚类的缺点。这个方法的指导思想就是,只要一个区域中的点的密度大过某个阈值,就把它加到与之相近的聚类中去。代表算法有:DBSCAN 算法、OPITICS 算法、DENCLUE 算法等。层次聚类对给定的数据集进行层次的分解,直到某种条件满足为止。具体又可分为“自底向上”和“自顶向下”两种方案。例如,在“自底向上”方案中,初始时每一个数据记录都组成一个单独的组,在接下来的迭代中,把那些相互邻近的组合并成一个组,直到所有的记录组成一个分组或者某个条件满足为止。代表算法有:AGNES 算法、BIRCH 算法、CURE 算法、CHAMELEON 算法等。基于网格的方法(Grid-Based Methods)首先将数据空间划分成有限个单元(Cell)的网格结构,所有的处理都是以单个单元为对象的。这样处理的一

个突出的优点就是处理速度很快，通常与目标数据库中的记录的个数无关，它只与把数据空间分为多少个单元有关。代表算法有：STING 算法、CLIQUE 算法、WAVE-CLUSTER 算法，既是基于密度的又是基于网格的。基于模型的方法给每一个聚类假定一个模型，然后去寻找能够很好地满足这个模型的数据集。这样一个模型可能是数据点在空间中的密度分布函数或者其他。它的一个潜在的假定就是：目标数据集是由一系列的概率分布所决定的。通常有两种尝试方向：统计的方案和神经网络的方案。

聚类算法的用户不但需要深刻地了解所用的特殊技术，而且还要知道数据收集过程的细节及拥有应用领域的专家知识。用户对手头数据了解得越多，用户越能成功的评估它的真实结构。聚类分析就是研究如何在没有训练的条件下把对象划分为若干类。

需要说明的是，这些算法本身无所谓优劣，而最终运用于数据的效果却存在好坏差异，这在很大程度上取决于数据使用者对于算法的选择是否得当。

本章重点介绍的主要内容是：由 Lloyd、J.B.MacQueen 等人基于划分思想提出的 K-Means 聚类算法、基于密度聚类的典型算法 DBSCAN 算法、基于层次聚类的 AGNES 算法、基于概率统计模型的高斯混合聚类算法、基于神经网络模型的 LVQ 聚类算法，以及基于网格的 CLIQUE 聚类算法。后文中将详细介绍这六类具有代表性的聚类分析算法的原理和 Python 代码实现。

5.1.2　聚类分析中的相关概念

1. 簇

聚类是把各不相同的个体分割为有更多相似性子集合的工作。它试图将数据集中的样本划分为若干个子集——通常是不相交的子集，而每个子集称为一个“簇”(Cluster)，也可以称之为类或类别。聚类生成的子集，合称为簇、簇识别(Cluster Identification)，一般用 C_i 来表示第 i 个簇。假定有一些数据，现在将相似数据归到一起，簇识别会告诉我们这些数据到底都是什么数据。

下面以表 5.1 中的西瓜样本数据集 4.0 为例，来说明聚类分析中的概念以及聚类分析算法的过程和性能度量。通过对这一批西瓜进行聚类分析，分析出西瓜的聚类情况，进而对西瓜的类别判断提供参考。

本数据集选取了 30 条西瓜特征的数据，包含两个特征，分别是西瓜的密度和西瓜的含糖率。为了方便说明，将编号为 i 的样本称为 $\boldsymbol{x}_i$。

表 5.1　西瓜样本数据集 4.0

编号	密度	含糖率	编号	密度	含糖率
1	0.697	0.460	8	0.437	0.211
2	0.774	0.376	9	0.666	0.091
3	0.634	0.264	10	0.243	0.267
4	0.608	0.318	11	0.245	0.057
5	0.556	0.215	12	0.343	0.099
6	0.403	0.237	13	0.639	0.161
7	0.481	0.149	14	0.657	0.198

续表

编号	密度	含糖率	编号	密度	含糖率
15	0.360	0.370	23	0.483	0.312
16	0.593	0.042	24	0.478	0.437
17	0.719	0.103	25	0.525	0.369
18	0.359	0.188	26	0.751	0.489
19	0.339	0.241	27	0.532	0.472
20	0.282	0.257	28	0.473	0.376
21	0.748	0.232	29	0.725	0.445
22	0.714	0.346	30	0.446	0.459

(*注：表 5.1 数据集来源于周志华《机器学习》教材，在本章的 K-Means、DBSCAN、AGNES、高斯混合聚类、LVQ 聚类以及 CLIQUE 等聚类算法中都将使用此数据集作为示例数据集进行算法演示。)

例如，将上述西瓜数据集中 30 个样例数据聚为 3 类，假设聚类分析之后将序号为 1，6，15，21，22，24，25，28，29 的聚为第一类，序号为 9，10，11，13，16，20，26 的聚为第二类，而剩余的序号为 2，3，4，5，7，8，12，14，17，18，19，23，27，30 的聚为第三类，那么这个例子中一共有三个簇，分别命名为 C_1、C_2 和 C_3，具体内容如下。

$$C_1=\{\boldsymbol{x}_1,\boldsymbol{x}_6,\boldsymbol{x}_{15},\boldsymbol{x}_{21},\boldsymbol{x}_{22},\boldsymbol{x}_{24},\boldsymbol{x}_{25},\boldsymbol{x}_{28},\boldsymbol{x}_{29}\}$$

$$C_2=\{\boldsymbol{x}_9,\boldsymbol{x}_{10},\boldsymbol{x}_{11},\boldsymbol{x}_{13},\boldsymbol{x}_{16},\boldsymbol{x}_{20},\boldsymbol{x}_{26}\}$$

$$C_3=\{\boldsymbol{x}_2,\boldsymbol{x}_3,\boldsymbol{x}_4,\boldsymbol{x}_5,\boldsymbol{x}_7,\boldsymbol{x}_8,\boldsymbol{x}_{12},\boldsymbol{x}_{14},\boldsymbol{x}_{17},\boldsymbol{x}_{18},\boldsymbol{x}_{19},\boldsymbol{x}_{23},\boldsymbol{x}_{27},\boldsymbol{x}_{30}\}$$

通过这样的划分，每个簇 C_i 可能对应于一些潜在的概念(类别)，如“好瓜”“中等瓜”“坏瓜”等。需要说明的是，这些概念对聚类算法而言事先是未知的，聚类过程仅能自动形成簇结构，簇所对应的概念语义需由使用者来把握和命名。

2. 相似度、相异度

聚类分析的实质是：以相似性为基础的无监督学习，它将相似的对象归类到同一个簇中，将不相似的对象归到不同簇中。它在一个聚类中的模式之间比不在同一聚类中的模式之间具有更多的相似性。而相似这一概念取决于所选择的相似度计算方法。

例如，学生在线学习数据被聚成 5 类之后，生成的簇内部的任意两个对象之间具有较高的相似度，属于不同簇的两个对象间具有较高的相异度。

对于不同数据类型，具有不同的相异度计算方法。聚类算法的基本出发点在于根据对象间相似度将对象划分为不同的类。对于 n 个数据对象，其可能具有 m 个属性变量，其中，属性变量可能是区间标度变量、二元变量、标称变量、序数型变量、比例标度变量等，对于不同类型的属性变量以及由各种类型变量组成的混合类型变量的相似度计算，需要采用特定的方法。这部分内容比较多，关于相似度和距离的概念及相关内容(知识)，运用图 5.3 来详细描述相似度/相异度计算所涉及的术语。

具体解释：假定样本集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$ 包含 m 个无标记样本，每个样本 $\boldsymbol{x}_i=(\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{in})$ 是一个 n 维特征向量，则聚类算法将样本集 D 划分为 k 个不相交的簇 $\{C_l\mid l=1,2,\cdots,k\}$，其中，$C_{l'}\bigcap\limits_{l'\neq l}C_l=\varnothing$ 且 $D=\bigcup\limits_{l=1}^{k}C_l$。相应地，用 $\lambda_j\in\{1,2,\cdots,k\}$ 表示样本 $\boldsymbol{x}_j$ 的

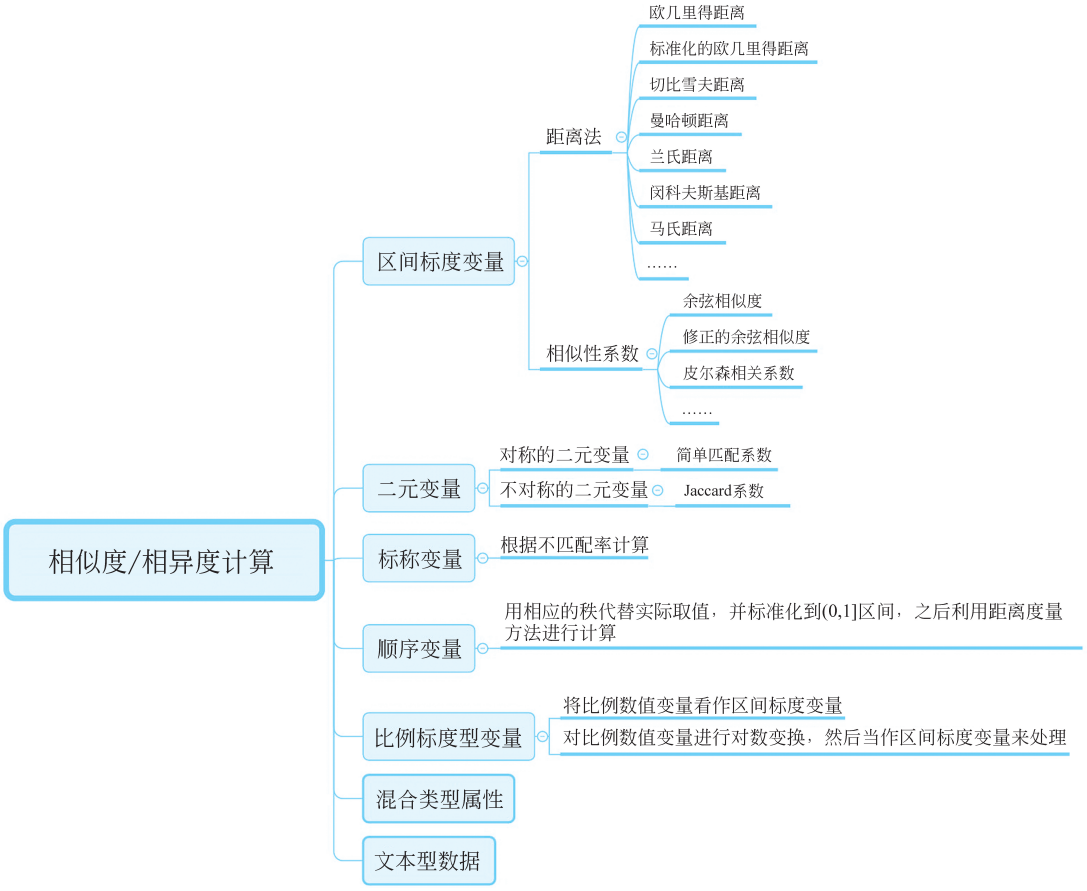

图 5.3　相似度/相异度计算所涉及的术语

“簇标记”(cluster label)，即 $\boldsymbol{x}_j \in C_{\lambda_j}$，于是，聚类的结果可用包含 m 个元素的簇标记向量 $\boldsymbol{\lambda}=(\lambda_1,\lambda_2,\cdots,\lambda_m)$ 表示。

聚类既能作为一个单独过程，用于找寻数据内在的分布结构，也可作为分类等其他学习任务的前驱过程。例如，在一些商业应用中需要对新用户的类型进行判别，但定义“用户类型”对商家来说却可能不太容易，此时往往可先对用户数据进行聚类，根据聚类结果将每个簇定义为一个类，然后再基于这些类训练分类模型，用于判别新用户的类型。

基于不同的学习策略，人们设计出多种类型的聚类算法，本章后半部分将对其中最常用的 K-Means 均值聚类、DBSCAN 密度聚类、AGNES 层次聚类、高斯混合聚类、LVQ 神经网络模型聚类、CLIQUE 网格聚类等典型的聚类算法进行重点介绍，同时在此之前，还将在 5.1.4 节讨论聚类算法涉及的两个基本问题——性能度量和距离计算。

5.1.3　聚类分析的应用

聚类分析源于许多研究领域，包括数据挖掘、统计学、生物学，以及机器学习。聚类的用途很广，在商业上，聚类可以帮助市场分析人员从消费者数据库中区分出不同的消费群体来，并且概括出每一类消费者的消费模式或者说习惯。它作为数据挖掘中的一个模块，可以作为一个单独的工具以发现数据库中分布的一些深层的信息，并且概括出每一类的特点，或

者把注意力放在某一个特定的类上以做进一步的分析;并且,聚类分析也可以作为数据挖掘算法中其他分析算法的一个预处理步骤。

例如,为了满足某些数据挖掘算法(如关联规则)的需要,需要对连续的数据进行离散化处理,使条件属性和决策属性值简约化、规范化。这时就需要对数据进行聚类处理,下面就详细介绍一下聚类处理。

上述提到的聚类(Clustering)应用的重要任务是:将数据对象分组成为多个类或簇(Cluster),在同一个簇中的对象之间具有较高的相似度,而不同簇中的对象差别较大。相似度是根据描述对象的属性值来计算的。聚类是经常采用的度量方式。

聚类分析是一个具有很强挑战性的领域,还存在一些潜在的应用,如客户价值分析、文本分类、基因识别、空间数据处理、卫星图片分析等。数据分析、统计学、机器学习、空间数据库技术、生物学和市场学这些潜在的应用也推动了聚类分析研究领域的进展和拓宽了应用领域。

一些应用还对分析算法提出了特别的要求,具有以下典型的特征。

(1) 伸缩性。

这里的可伸缩性是指算法要能够处理大数据量的数据库对象,如处理上百万条记录的数据库。这就要求算法的时间复杂度不能太高,最好是多项式时间的算法。值得注意的是,当算法不能处理大数据量时,用抽样的方法来弥补也不是一个好主意,因为抽样的方法通常会导致歪曲的结果。

(2) 处理不同字段类型的能力。

算法不仅要能处理数值性的字段,还要有处理其他类型字段的能力,例如,布尔型、枚举型、序数型及混合型等。

(3) 发现具有任意形状的聚类的能力。

很多聚类分析算法采用基于欧几里得距离的相似性度量方法,这一类算法发现的聚类通常是一些球状的、大小和密度相近的类;但可以想象,显示数据库中的聚类可能是任意形状的,甚至是具有分层树的形状,故要求算法有发现任意形状的聚类的能力。

(4) 输入参数对领域知识的依赖性。

很多聚类算法都要求用户输入一些参数,例如,需要发现的聚类数、结果的支持度及置信度等。聚类分析的结果通常都对这些参数很敏感,但另一方面,对于高维数据,这些参数又是相当难以确定的。这样就加重了用户使用这个工具的负担,使得分析的结果很难控制。一个好的聚类算法应当针对这个问题,给出一个好的解决方法。

(5) 能够处理异常数据。

现实数据库中常常包含异常数据,例如,数据不完整、缺乏某些字段的值,甚至是包含错误数据现象。有一些数据算法可能会对这些数据很敏感,从而导致错误的分析结果。

(6) 结果对输入记录顺序的无关性。

有些分析算法对记录的输入顺序是敏感的,也即对同一个数据集,将它以不同的顺序输入到分析算法,得到的结果会不同,这是我们不希望的。

(7) 处理高维数据的能力。

一个数据库或者数据仓库都有很多的字段或者说明,一些分析算法对处理维数较少的数据集时表现不错,例如,二维、三维的数据。人的理解能力也可以对二维、三维数据的聚类

分析结果的质量做出较好的判别，但对于高维数据就没有那么直观了。所以对于高维数据的聚类分析是很具有挑战性的，特别是考虑到在高维空间中，数据的分布是极其稀疏的，而且形状也可能是极其不规则的。

(8) 增加限制条件后的聚类分析能力。

现实的应用中总会出现各种其他限制，我们希望聚类算法可以在考虑这些限制的情况下，仍旧有很好的表现。

(9) 结果的可解释性和可用性。

聚类的结果最终都是要面向用户的，所以结果应该是容易解释和理解的，并且是可应用的。这就要求聚类算法必须与一定的语义环境及语义解释相关联。领域知识如何影响聚类分析算法的设计是很重要的一个研究方面。

5.1.4　聚类性能度量

聚类性能度量也称为聚类“有效性指数”(validity index)，与监督学习中的性能作用相似，对于聚类结果，一方面需通过某种性能度量来评估其好坏；另一方面，若明确了最终将使用的性能度量，则可直接将其作为聚类过程的优化目标，从而更好地得到符合要求的聚类结果。

聚类是将样本集 D 划分为若干互不相交的子集，即样本簇。那么什么样的聚类结果比较好呢？直观上看，我们希望“物以类聚”，即同一簇的样本尽可能彼此相似，不同簇的样本尽可能不同。换言之，聚类结果的“簇内相似度”(intra-cluster similarity)高且“簇间相似度”(inter-cluster similarity)低。

1. 距离计算

在计算聚类的性能指标时需要用到距离计算，这一部分将首先介绍和聚类性能指标相关的距离概念及相应计算。

对于函数 dist(·,·)，若它是一个“距离度量”(distance measure)，则需要满足如下一些基本性质。

非负性，如式(5-1)所示。

$$\mathrm{dist}(\boldsymbol{x}_i,\boldsymbol{x}_j)\geqslant 0; \tag{5-1}$$

同一性，如式(5-2)所示。

$$\mathrm{dist}(\boldsymbol{x}_i,\boldsymbol{x}_j)=0 \quad \text{当且仅当 } \boldsymbol{x}_i=\boldsymbol{x}_j \tag{5-2}$$

对称性，如式(5-3)所示。

$$\mathrm{dist}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\mathrm{dist}(\boldsymbol{x}_j,\boldsymbol{x}_i) \tag{5-3}$$

直通性，如式(5-4)所示。

$$\mathrm{dist}(\boldsymbol{x}_i,\boldsymbol{x}_j)\leqslant \mathrm{dist}(\boldsymbol{x}_i,\boldsymbol{x}_k)+\mathrm{dist}(\boldsymbol{x}_k,\boldsymbol{x}_j) \tag{5-4}$$

给定样本 $\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{in})$ 与 $\boldsymbol{x}_j=(x_{j1},x_{j2},\cdots,x_{jn})$，最常用的距离是“闵可夫斯基距离”(Minkowski distance)，如式(5-5)所示。

$$\mathrm{dist}_{\mathrm{mk}}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\left(\sum_{u=1}^{n}|x_{iu}-x_{ju}|^p\right)^{\frac{1}{p}} \tag{5-5}$$

当 $p\geqslant 1$ 时，式(5-5)显然满足式(5-1)～式(5-4)距离度量的基本性质。

当 $p=2$ 时，闵可夫斯基距离即欧氏距离(Euclidean distance)，如式(5-6)所示。

$$\mathrm{dist}_{\mathrm{ed}}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\|\boldsymbol{x}_i,\boldsymbol{x}_j\|_2=\sqrt{\sum_{u=1}^{n}|x_{iu}-x_{ju}|^2} \tag{5-6}$$

当 $p=1$ 时,闵可夫斯基距离即曼哈顿距离(Manhattan distance),如式(5-7)所示。

$$\mathrm{dist}_{\mathrm{man}}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\|\boldsymbol{x}_i,\boldsymbol{x}_j\|_1=\sum_{u=1}^{n}|x_{iu}-x_{ju}| \tag{5-7}$$

我们常将属性划分为"连续属性"(continuous attribute)和"离散属性"(categorical attribute),前者在定义域上有无穷多个可能的取值,后者在定义域上是有限个取值。然而,在讨论距离计算时,属性上是否定义了"序"关系更为重要。例如,定义域为{1,2,3}的离散属性与连续属性的性质更接近一些,能直接在属性值上计算距离:"1"与"2"比较接近、与"3"比较远,这样的属性称为"有序属性"(ordinal attribute);而定义域为{飞机,火车,轮船}这样的离散属性则不能直接在属性值上计算距离,称为"无序属性"(non-ordinal attribute)。显然,闵可夫斯基距离可用于有序属性。

对无序属性可采用 VDM (Value Difference Metric)[Stanfill, Waltz,1986]。令 $m_{u,a}$ 表示在属性 u 上取值为 a 的样本数,$m_{u,a,i}$ 表示在第 i 个样本簇中在属性 u 上取值为 a 的样本数,k 为样本簇数,则属性 u 上两个离散值 a 与 b 之间的 VDM 距离为式(5-8):

$$\mathrm{VDM}_p(a,b)=\sum_{i=1}^{k}\left|\frac{m_{u,a,i}}{m_{u,a}}-\frac{m_{u,b,i}}{m_{u,b}}\right|^p \tag{5-8}$$

于是,将闵可夫斯基距离和 VDM 结合即可处理混合属性。假定有 n_c 个有序属性、$n-n_c$ 个无序属性,不失一般性,令有序属性排列在无序属性之前,则有式(5-9):

$$\mathrm{MinkovDM}_p(\boldsymbol{x}_i,\boldsymbol{x}_j)=\left(\sum_{u=1}^{n_c}|x_{iu}-x_{ju}|^p+\sum_{u=u_c+1}^{n}\mathrm{VDM}_p(x_{iu},x_{ju})\right)^{\frac{1}{p}} \tag{5-9}$$

当样本空间中不同属性的重要性不同时,可使用"加权距离"(weighted distance)。以加权闵可夫斯基距离为例,见式(5-10):

$$\mathrm{dist}_{\mathrm{wmk}}(\boldsymbol{x}_i,\boldsymbol{x}_j)=(w_1\cdot|x_{i1}-x_{j1}|^p+\cdots+w_n\cdot|x_{in}-x_{jn}|^p)^{\frac{1}{p}} \tag{5-10}$$

其中,权重 $w_i\geqslant 0(i=1,2,\cdots,n)$ 表征不同属性的重要性,通常 $\sum_{i=1}^{n}w_i=1$。

需要注意的是,通常我们是基于某种形式的距离来定义"相似度度量"(similarity measure),距离越大,相似度越小。然而,用于相似度度量的距离未必一定要满足距离度量的所有基本性质尤其是直通性。例如,在某些任务中可能希望有这样的相似度度量:"人"和"马"分别与"人马"相似,但"人"与"马"很不相似;要达到这个目的,可以令"人""马"与"人马"之间的距离都比较小,但"人"与"马"之间的距离很大。此时该距离不再满足直通性;这样的距离称为"非度量距离"(non-metric distance)。

此外,本节介绍的距离计算式都是事先定义好的,但在不少现实任务中,有必要基于数据样本来确定合适的距离计算式,这可通过"距离度量学习"(distance metric learning)来实现。

综上对于变量,或者属性,大致可以分为两类:第一类,定量变量即通常所说的连续型变量;第二类,定性变量即这些量并非真有数量上的变化,而只有性质上的差异。这些量又可以分为两种,一种是有序变量,另一种是名义变量。

(1) 对于连续型变量,常用的相似性度量用的是距离,典型的距离定义有如表 5.2 所示

的表示形式。

表 5.2　典型的距离定义的表示形式

距　离	定　义　式	说　明
绝对值距离	$d_{ij}(1)=\sum_{k=1}^{p}\lvert x_{ik}-x_{jk}\rvert$	绝对值距离是在一维空间下进行的距离计算
欧氏距离	$d_{ij}(2)=\sqrt{\sum_{k=1}^{p}(x_{ik}-x_{jk})^2}$	欧氏距离是在二维空间下进行的距离计算
闵可夫斯基距离	$d_{ij}(q)=\left[\sum_{k=1}^{p}(x_{ik}-x_{jk})^q\right]^{1/q},q>0$	闵可夫斯基距离是在 q 维空间下进行的距离计算
切比雪夫距离	$d_{ij}(\infty)=\max_{1\leqslant k\leqslant p}\lvert x_{ik}-x_{jk}\rvert$	切比雪夫距离是 q 取正无穷大时的闵可夫斯基距离，即切比雪夫距离是在 $+\infty$ 维空间下进行的距离计算
Lance 距离	$d_{ij}(L)=\sum_{k=1}^{p}\frac{\lvert x_{ik}-x_{jk}\rvert}{x_{ik}+x_{jk}}$	减弱极端值的影响能力
归一化距离	$d_{ij}=\sum_{k=1}^{p}\frac{\lvert x_{ik}-x_{jk}\rvert}{\max(x_k)-\min(x_k)}$	自动消除不同变量间的量纲影响，其中每个变量 k 的距离取值均是[0,1]

(2) 对于离散型变量，常见的相似性度量有**相似系数**。

两个仅包含二元属性的对象之间的相似性度量也称相似系数。

假设两个对象的比较有四种情况：$f_{00}=x$ 取 0 并且 y 取 0 的属性个数；$f_{01}=x$ 取 0 并且 y 取 1 的属性个数；$f_{10}=x$ 取 1 并且 y 取 0 的属性个数；$f_{11}=x$ 取 1 并且 y 取 1 的属性个数。

则有如下简单匹配系数(SMC)和杰卡德系数(Jaccard，JC)定义。

① 简单匹配系数。

$$\begin{aligned}\text{SMC}&=\text{值匹配的属性个数}\ /\ \text{属性总个数}\\&=(f_{11}+f_{00})\ /\ (f_{01}+f_{10}+f_{11}+f_{00})\end{aligned}$$

② Jaccard(杰卡德) 系数。

$$\begin{aligned}\text{JC}&=\text{值为 1 匹配的个数}\ /\ \text{不涉及 0-0 匹配的属性个数}\\&=(f_{11})/(f_{01}+f_{10}+f_{11})\end{aligned}$$

例 5.1　简单匹配系数和 Jaccard 系数的计算。

假设有两个二元向量，分别为 $\boldsymbol{x}$ 和 $\boldsymbol{y}$：

$$\boldsymbol{x}=(1,0,0,0,0,0,0,0,0,0)$$
$$\boldsymbol{y}=(0,0,0,0,0,0,1,0,0,1)$$

则有：

$$f_{00}=7\quad(x\text{ 取 0 并且 }y\text{ 取 0 的属性个数})$$
$$f_{01}=2\quad(x\text{ 取 0 并且 }y\text{ 取 1 的属性个数})$$
$$f_{10}=1\quad(x\text{ 取 1 并且 }y\text{ 取 0 的属性个数})$$
$$f_{11}=0\quad(x\text{ 取 1 并且 }y\text{ 取 1 的属性个数})$$

根据匹配系数和 Jaccard 系数的计算公式容易得到：