ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.1, January 2008, pp.48−61
DOI: 10.3724/SP.J.1001.2008.00048

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

聚类算法研究*

孙吉贵[1,2], 刘 杰[1,2+], 赵连宇[1,2]

[1](吉林大学 计算机科学与技术学院,吉林 长春 130012)

[2](符号计算与知识工程教育部重点实验室,吉林 长春 130012)

Clustering Algorithms Research

SUN Ji-Gui[1,2], LIU Jie[1,2+], ZHAO Lian-Yu[1,2]

[1](College of Computer Science and Technology, Jilin University, Changchun 130012, China)

[2](Key Laboratory of Symbolic Computation and Knowledge Engineering of the Ministry of Education, Changchun 130012, China)

+ Corresponding author: Phn: +86-431-85166478, E-mail: liu_jie@jlu.edu.cn

Sun JG, Liu J, Zhao LY. Clustering algorithms research. *Journal of Software*, 2008,19(1):48−61. http://www.jos.org.cn/ 1000-9825/19/48.htm

Abstract: The research actuality and new progress in clustering algorithm in recent years are summarized in this paper. First, the analysis and induction of some representative clustering algorithms have been made from several aspects, such as the ideas of algorithm, key technology, advantage and disadvantage. On the other hand, several typical clustering algorithms and known data sets are selected, simulation experiments are implemented from both sides of accuracy and running efficiency, and clustering condition of one algorithm with different data sets is analyzed by comparing with the same clustering of the data set under different algorithms. Finally, the research hotspot, difficulty, shortage of the data clustering and some pending problems are addressed by the integration of the aforementioned two aspects information. The above work can give a valuable reference for data clustering and data mining.

Key words: clustering; algorithm; experiment

摘 要: 对近年来聚类算法的研究现状与新进展进行归纳总结.一方面对近年来提出的较有代表性的聚类算法,从算法思想、关键技术和优缺点等方面进行分析概括;另一方面选择一些典型的聚类算法和一些知名的数据集,主要从正确率和运行效率两个方面进行模拟实验,并分别就同一种聚类算法、不同的数据集以及同一个数据集、不同的聚类算法的聚类情况进行对比分析.最后通过综合上述两方面信息给出聚类分析的研究热点、难点、不足和有待解决的一些问题.上述工作将为聚类分析和数据挖掘等研究提供有益的参考.

关键词: 聚类;算法;实验

中图法分类号: TP18 **文献标识码**: A

聚类分析研究有很长的历史,几十年来,其重要性及与其他研究方向的交叉特性得到人们的肯定.聚类是数

* Supported by the National Natural Science Foundation of China under Grant Nos.60473003, 60573073 (国家自然科学基金); the Major Research Program of National Natural Science Foundation of China under Grant No.60496321 (国家自然科学基金重大项目)

Received 2007-04-24; Accepted 2007-08-03

据挖掘、模式识别等研究方向的重要研究内容之一,在识别数据的内在结构方面具有极其重要的作用.聚类主要应用于模式识别中的语音识别、字符识别等,机器学习中的聚类算法应用于图像分割和机器视觉,图像处理中聚类用于数据压缩和信息检索.聚类的另一个主要应用是数据挖掘(多关系数据挖掘)、时空数据库应用(GIS等)、序列和异类数据分析等.此外,聚类还应用于统计科学.值得一提的是,聚类分析对生物学、心理学、考古学、地质学、地理学以及市场营销等研究也都有重要作用[1−3].

本文一方面从算法思想、关键技术和优缺点等方面对近年提出的较有代表性的聚类算法进行了分析、介绍;另一方面又选用多个知名数据集对一些典型算法进行了测试.而后综合这两方面信息得出一些相应的结论.

本文第 1 节简单介绍聚类概念、聚类过程与聚类算法的类别.第 2 节重点阐述 17 个较有代表性的算法.第 3 节描述 8 种聚类算法的模拟实验结果,并结合文献[4]进行分析.第 4 节给出本文的一些结论.

1 聚类与聚类算法类别

1.1 聚类概念与聚类过程

迄今为止,聚类还没有一个学术界公认的定义.这里给出 Everitt[5]在 1974 年关于聚类所下的定义:一个类簇内的实体是相似的,不同类簇的实体是不相似的;一个类簇是测试空间中点的会聚,同一类簇的任意两个点间的距离小于不同类簇的任意两个点间的距离;类簇可以描述为一个包含密度相对较高的点集的多维空间中的连通区域,它们借助包含密度相对较低的点集的区域与其他区域(类簇)相分离.

事实上,聚类是一个无监督的分类,它没有任何先验知识可用.聚类的形式描述如下:

令 $U=\{p_1,p_2,\ldots,p_n\}$表示一个模式(实体)集合,p_i 表示第 i 个模式 $i=\{1,2,\ldots,n\}$;$C_t\subseteq U,t=1,2,\ldots,k$,$C_t=\{p_{t_1},p_{t_2},\ldots,p_{t_w}\}$;$proximity(p_{ms},p_{ir})$,其中,第 1 个下标表示模式所属的类,第 2 个下标表示某类中某一模式,函数 *proximity* 用来刻画模式的相似性距离.若诸类 C_t 为聚类之结果,则诸 C_t 需满足如下条件:

1) $\bigcup_{t=1}^{k}C_t=U$.

2) 对于$\forall C_m,C_r\subseteq U,C_m\neq C_r$,有 $C_m\cap C_r=\varnothing$(仅限于刚性聚类);

$\mathrm{MIN}_{\forall p_{mu}\in C_m,\forall p_{rv}\in C_r,\forall C_m,C_r\subseteq U\&C_m\neq C_r}(proximity(p_{mu},p_{rv}))>\mathrm{MAX}_{\forall p_{mx},p_{my}\in C_m,\forall C_m\subseteq U}(proximity(p_{mx},p_{my}))$.

典型的聚类过程主要包括数据(或称之为样本或模式)准备、特征选择和特征提取、接近度计算、聚类(或分组)、对聚类结果进行有效性评估等步骤[3,6,7].

聚类过程:

1) 数据准备:包括特征标准化和降维.
2) 特征选择:从最初的特征中选择最有效的特征,并将其存储于向量中.
3) 特征提取:通过对所选择的特征进行转换形成新的突出特征.
4) 聚类(或分组):首先选择合适特征类型的某种距离函数(或构造新的距离函数)进行接近程度的度量;而后执行聚类或分组.
5) 聚类结果评估:是指对聚类结果进行评估.评估主要有 3 种:外部有效性评估、内部有效性评估和相关性测试评估.

1.2 聚类算法的类别

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[7].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.聚类算法有多种分类方法,本文将聚类算法大致分成层次化聚类算法、划分式聚类算法、基于密度和网格的聚类算法和其他聚类算法,如图 1 所示的 4 个类别.

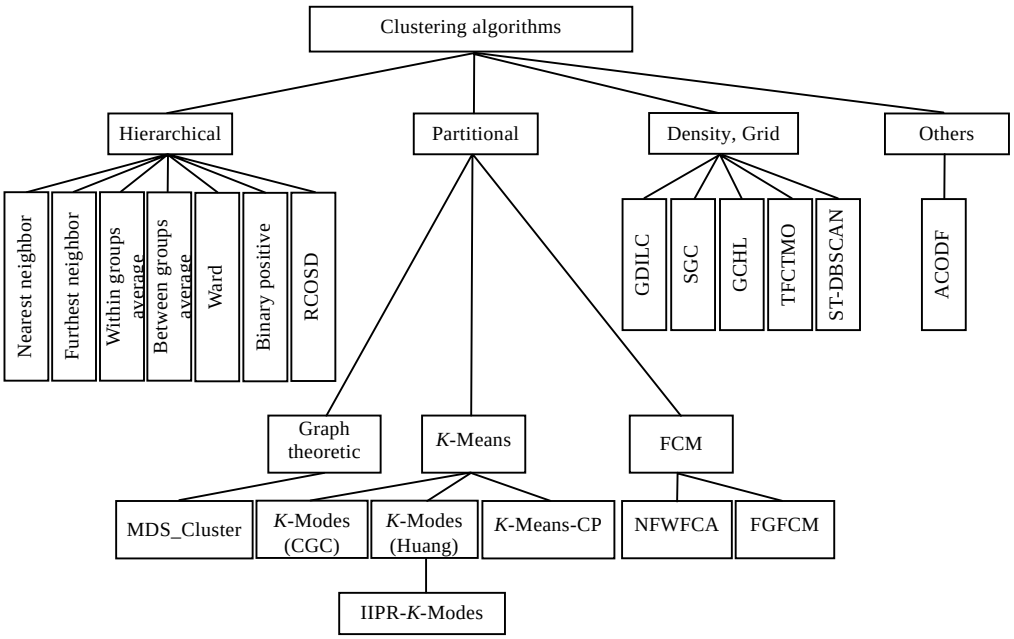

Fig.1 The classification chart of clustering algorithms

图 1 聚类算法分类图

2 聚类算法

2.1 层次聚类算法

层次聚类算法又称为树聚类算法[8,9],它使用数据的联接规则,透过一种层次架构方式,反复将数据进行分裂或聚合,以形成一个层次序列的聚类问题解.本文仅以层次聚类算法中的层次聚合算法为例进行介绍.层次聚合算法的计算复杂性为 $O(n^2)$,适合于小型数据集的分类.

2.1.1 层次聚合算法

该算法由树状结构的底部开始逐层向上进行聚合,假定样本集 $S=\{o_1,o_2,\ldots,o_n\}$共有 n 个样本.

HA1[初始化]. 置每个样本 o_i 为一个类; /*共形成 n 个类:$o_1,o_2,\ldots,o_n$*/

HA2[找最近的两个类]. $distance(o_r,o_k)=\min_{\forall o_u,o_v\in S,o_u\neq o_v} distance(o_u,o_v)$;

/*从现有的所有类中找出距离最近(相似度最大)的两个类 o_r 和 o_k*/

HA3[合并 o_r 和 o_k]. 将类 o_r 和 o_k 合并成一个新类 o_{rk}; /*现有的类数将减 1*/

HA4. 若所有的样本都属于同一个类,则终止本算法;否则,返回步骤 HA2.

2.1.2 传统聚合规则

两个类之间距离的度量方法是传统层次聚合算法的重要组成部分,它主要包括两个重要参数相似性度量方法和联接规则.这里采用欧式距离作为相似性度量方法,联接规则主要包括单联接规则、完全联接规则、类间平均联接规则、类内平均联接规则和沃德法.这几种联接规则可定义如下[8](其中,含$\|x-y\|$是欧几里德范数, n_i 和 n_k 分别指类 o_r 和 o_k 中的样本个数,$C(n_i+n_k,2)$表示从 n_i+n_k 个元素中抽出两个元素的不同组合的方法总数):

单联接聚合规则: $d(o_i,o_k)=\min_{x\in o_i,y\in o_k}\|x-y\|$;

全联接聚合规则: $d(o_i,o_k)=\max_{x\in o_i,y\in o_k}\|x-y\|$;

类间平均联接聚合规则: $d(o_i,o_k)=(1/n_in_k)\sum_{x\in o_i}\left(\sum_{y\in o_k}\|x-y\|\right)$;

类内平均联接聚合规则: $d(o_i,o_k)=(1/C(n_i+n_k,2))\sum_{x,y\in(o_i,o_k)}\|x-y\|$;

沃德法: $d(o_i,o_k)=(1/(n_i+n_k))\sum_{x\in(o_i,o_k)}\|x-n\|^2$,其中,$n$ 是融合聚类的中心.

2.1.3 新层次聚合算法

(1) Binary-Positive 方法

2007 年,Gelbard 等人[4]提出了一种新的层次聚合算法,被称为正二进制(binary-positive)方法.该方法把待分类数据以正的二进制形式存储于一个二维矩阵中,其中,行表示记录(对象),列表示其属性的可能取值.记录对应的取值为 1 或者 0,分别表示此记录有对应的属性值或者不存在对应属性值.因此,相似性距离计算只在被比较的二进制向量中的正比特位上进行,即只在取值为 1 的记录(对象)之间进行.有以 Dice 距离为代表的多种 Binary-Positve 相似性测量方法[10,11].

Gelbard 等人采用 Wine,Iris,Ecolic 和 Psychology balance 这 4 种数据集对 11 种聚类算法进行了实验,结果表明,对于此 4 种数据集中的任意一种数据的聚类结果,Binary-Positive 等 4 种方法在聚类结果的准确率方面,从总体上来看都是最好的.同时他们还认为,将原始数据转换成正二进制会改善聚类结果的正确率和聚类的鲁棒性,对于层次聚类算法尤其如此.

(2) 连续数据的粗聚类算法(rough clustering of sequential data,简称 RCOSD)

2007 年,Kumar 等人[12]面向连续数据提出了一种新的基于不可分辨粗聚合的层次聚类算法 RCOSD.在该算法中,不可分辨关系被扩展成具有不严格传递特性的容差关系.使用相似性的上近似形成初始类,使用约束相似性的上近似概念形成后续类,其中的一个相对的相似性条件被用作合并准则.RCOSD 的关键思想是寻找能捕捉数据序列的连续信息及内容信息的一个特征集,并把这些特征集映射到一个上近似空间,应用约束相似性上近似技术获得粗类簇的上近似,其中一个元素可以属于多个类簇.该算法引入 S^3M 作为 Web 数据的相似性度量方法,S^3M 既考虑了项的出现次序又考虑了集合内容.该算法每一次迭代可以合并两个或多个类,所以加快了层次聚类速度.该算法能够有效挖掘连续数据,并刻画类簇的主要特性,帮助 Web 挖掘者描述潜在的新的 Web 用户组的特性.

Pradeep Kumar 等人在本质连续的 MSNBC Web 导航数据集上的实验结果表明,与使用序列向量编码的传统层次化聚类算法相比,RCOSD 聚类算法是可行的.算法给出的描述方法能够帮助 Web 挖掘者鉴别潜在的有意义的用户组.

2.2 划分式聚类算法

划分式聚类算法需要预先指定聚类数目或聚类中心,通过反复迭代运算,逐步降低目标函数的误差值,当目标函数值收敛时,得到最终聚类结果.

2.2.1 K 均值聚类

1967 年,MacQueen 首次提出了 K 均值聚类算法(K-means 算法).迄今为止,很多聚类任务都选择该经典算法.该算法的核心思想是找出 K 个聚类中心 $c_1,c_2,\ldots,c_K$,使得每一个数据点 x_i 和与其最近的聚类中心 c_v 的平方距离和被最小化(该平方距离和被称为偏差 D).

K 均值(K-means)聚类算法[8](对 n 个样本进行聚类)

K1[初始化]. 随机指定 K 个聚类中心($c_1,c_2,\ldots,c_K$);

K2[分配 x_i]. 对每一个样本 x_i,找到离它最近的聚类中心 c_v,并将其分配到 c_v 所标明类;

K3[修正 c_w]. 将每一个 c_w 移动到其标明的类的中心;

K4[计算偏差]. $D=\sum_{i=1}^{n}[\min_{r=1,\ldots,K}d(x_i,c_r)^2]$;

K5[D 收敛?]. 如果 D 值收敛,则 $return(c_1,c_2,\ldots,c_K)$并终止本算法;否则,返回步骤 K2.

K-means 算法的优点与不足[13].优点:能对大型数据集进行高效分类,其计算复杂性为 $O(tKmn)$,其中,t 为迭代次数,K 为聚类数,m 为特征属性数,n 为待分类的对象数,通常,$K,m,t<<n$.在对大型数据集聚类时,K-means 算法比层次聚类算法快得多.不足:通常会在获得一个局部最优值时终止;仅适合对数值型数据聚类;只适用于聚类

结果为凸形(即类簇为凸形)的数据集.

以经典 *K*-means 算法为基础,研究者们提出了很多新的改进的 *K*-means 算法,下面对其中的一些算法加以介绍.

2.2.2 *K*-modes 算法

(1) *K*-modes-Huang 算法[14]

在阐述 *K*-modes 算法之前,先对 Means 与 Modes 做简单介绍.

在 *K*-means 算法中,mean 为类簇中心或称为质心,是指一个类簇中所有对象关于属性的均值,最初可随机指定.在 *K*-modes 算法中,modes 可定义如下:设 $X=\{X_1,X_2,\ldots,X_n\}$是一个数据集,$\forall X_i\in X$ 由 m 个分类属性$\{A_1,A_2,\ldots,A_m\}$来描述,X_i 可表示成向量$\langle x_{i1},x_{i2},\ldots,x_{im}\rangle$,又可表示成属性-值对的合取式$[A_1=x_{i1}]\wedge\ldots\wedge[A_m=x_{im}]$;$Q$ 是 X 的一个 mode,Q 可表示成向量$\langle q_1,q_2,\ldots,q_m\rangle$,也可表示成属性-值对的合取式$[A_1=q_1]\wedge\ldots\wedge[A_m=q_m]$,$Q$ 需使 $\sum_{i=1,\ldots,n}d_1(X_i,Q)$ 取最小值,$d_1(X_i,Q)$表示 X_i 与 Q 之间的距离,Q 不必是 X 的一个元素.

1998 年,Huang 为克服 *K*-means 算法仅适合于数值属性数据聚类的局限性,提出了一种适合于分类属性数据聚类的 *K*-modes 算法.该算法对 *K*-means 进行了 3 点扩展:引入了处理分类对象的新的相异性度量方法(简单的相异性度量匹配模式),使用 modes 代替 means,并在聚类过程中使用基于频度的方法修正 modes,以使聚类代价函数值最小化.

这些扩展允许人们能够直接使用 *K*-means 范例聚类有分类属性的数据,无须对数据进行变换.*K*-modes 算法的另一个优点是 modes 能给出类的特性描述,这对聚类结果的解释是非常重要的.事实上,*K*-modes 算法比 *K*-means 算法能够更快收敛.Huang 使用众所周知的大豆疾病数据集对其算法进行了测试,结果表明,*K*-modes 算法具有很好的聚类性能.进一步地,他用包含 50 万条记录和 34 个分类属性的健康保险数据集进行了测试,结果证明,该算法在(聚类的)类数和记录数两个方面是真正可伸缩的.

与 *K*-means 算法一样,*K*-modes 算法也会产生局部最优解,依赖于初始化 modes 的选择和数据集中数据对象的次序.初始化 modes 的选择策略尚需进一步研究.

1999 年,Huang 等人[15]证明了经过有限次迭代 *K*-modes 算法仅能收敛于局部最小值.

(2) K-modes-CGC 算法[16]

2001 年,Chaturvedi 等人提出一种面向分类属性数据(名义尺度数据)的非参数聚类方法,称为 *K*-modes-CGC 算法,类似于面向数值数据(间隔尺度数据)的传统 *K*-means 算法.与现存的大多数面向分类属性数据的聚类方法不同,*K*-modes-CGC 算法显式地优化一个基于 L_0 范数的损失函数.

在蒙特卡罗模拟中,Chaturvedi 等人用 *K*-modes-CGC 和潜类算法[17]来恢复一个已知的潜在类结构,结果表明,两者具有相等的执行效率.然而,*K*-modes-CGC 算法不但在速度方面比潜类算法快一个数量级,而且更少遇到局部最优的情况.对于包含大量分类变量的数据集,潜类算法计算极其缓慢,变得不可行.

尽管在一些情况下,潜类算法比 *K*-modes-CGC 算法执行得更好,但 Chaturvedi 猜测在另外一些情况下,潜类算法很可能是不可行的.因此,Chaturvedi 等人建议在执行聚类分析时应互补地使用这两种方法,同时给出了 *K*-modes-CGC 算法和潜类算法的经验比较,结果表明前者更占优势.

2003 年,Huang[18]证明了 *K*-modes-CGC 算法与 *K*-modes-Huang 算法是等价的.

2.2.3 迭代初始点集求精 *K*-modes 算法

2002 年,Sun 等人[19]将 Bradley 等人的迭代初始点集求精算法[20]应用于 *K*-modes 算法(Huang,1998).尽管 Huang 的 *K*-modes 算法能够聚类分类数据,但它需要预先决定或随机选择类(簇)的初始 modes,并且初始 modes 的差异常常会导致截然不同的聚类结果.文中,Sun 等人给出了一个关于应用 Bradley 等人的迭代初始点求精算法于 *K*-modes 聚类的实验研究.

Sun 等人用知名大豆疾病[21]数据集进行测试,大豆疾病数据包含 47 个记录,每个记录由 35 个特征描述.每个记录都被标记为以下 4 种疾病中的一种:Diaporthe StemCanker,Charcoal Rot,Rihizoctonia Root Rot 以及 Phytophthora Rot,除了 Phytophthora Rot 有 17 个记录外,其他 3 种疾病都有 10 个记录.针对 *K*-modes 算法,分两

种方案对大豆疾病数据集进行聚类实验:方案 1 随机选择初始点集;方案 2 采用迭代初始点集求精方法选择初始点集.实验结果表明,采用方案 2 的 *K*-modes 算法能够产生更高精度和更可靠的聚类结果.求精算法在给定数据集的一个小子样本集上进行,因此只需存储全部数据的内存空间的一小部分.然而,对于更大、更复杂分布的数据集,关于算法的可伸缩性和适应性方面还有许多问题需要研究.

2.2.4 一致性保留 *K*-means 算法(*K*-means-CP)

2004 年,Ding 等人[22]提出一致性保留 *K*-means 算法(*K*-means-CP).最近邻一致性是统计模式识别中的一个重要概念,他们将这个概念扩展到数据聚类,对一个类中的任意数据点,要求它的 *k* 最近邻和 *k* 互最近邻都必须在该类中.他们研究了类的 *k* 最近邻一致性的性质,提出了 *k*NN 和 *k*MN 一致性强制和改进算法,并提出了将类 *k* 最近邻或类 *k* 互最近邻一致性作为数据聚类的一种重要质量度量方法.他们选用互联网上 20 个新闻组数据集进行了实验,结果表明,*k* 最近邻一致性、*k* 互最近邻一致性以及算法聚类的正确率都得到显著改善.同时,这也表明局部一致性信息可帮助全局聚类目标函数优化.

算法 *K*-means-CP.

1[初始化]. 随机选择 *K* 个点作为初始类的中心($c_1,c_2,\ldots,c_K$);

2[分配近邻集]. 分配一个近邻集 *S*; //*将 *S* 分配到离其最近的类 *Cp* 中, $p=\arg\min_{v=1,\ldots,K}\sum_{x_i\in S}(x_i-m_v)^2$

3[更新类中心]. 置 $m_v=\sum_{x_i\in C_v}x_i/n_v$; //*更新聚类中心(即质心),$m_v$ 是类 C_v 的中心,$n_k=|C_k|$

4[收敛否?]. 质心不再移动,则终止算法;否则返回步骤 2. //* $J_{Km}=\sum_{v=1,\ldots,K}\sum_{x_i\in C_v}(x_i-m_v)^2$ 判断收敛

2.2.5 模糊聚类算法

1969 年,Ruspini 首次将模糊集理论应用到聚类分析中,提出了模糊聚类算法(fuzzy *c*-means,简称 FCM).FCM 算法是图像分割使用最多的方法之一,它的成功主要归功于为解决每个图像像素的隶属需要引入了模糊性.比之脆弱(crisp)或硬分割方法,FCM 能够保留初始图像的更多信息.然而,FCM 的一个缺点是不考虑图像上下文中的任何空间信息,这使得它对噪声和其他人造图像非常敏感.人们围绕 FCM 算法开展了大量研究,下面只对这方面的最新研究作简单介绍[23,24].

2006 年,李洁等人[25]提出基于特征加权的模糊聚类新算法 NFWFCA.传统模糊 *K*-均值算法、*K*-modes 算法和 *K*-原型算法都假定样本矢量的各维特征对聚类贡献相同.但在实际应用中,由于样本矢量的各维特征来自不同传感器,存在测量精度及可靠性等差异,样本矢量的各维特征对聚类影响不尽相同.以模糊 *K*-原型算法为基础,算法 NFWFCA 采用 ReliefF 算法[26]确定各维特征的权重,数值特征权值的计算方法为

$$\lambda^r=\lambda^r-\frac{diff\_hit^r}{R}+\frac{diff\_miss^r}{R}.$$

属性特征权值的计算方法为

$$\lambda^c=\lambda^c-\frac{diff\_hit^c}{R}+\frac{diff\_miss^c}{R}.$$

从而修正目标函数为

$$J(W,P)=\sum_{i=1,\ldots,k}\left[\sum_{j=1,\ldots,n}w_{ij}^2\sum_{m=1,\ldots,t}\lambda_m^r|x_{jm}^r-p_{jm}^r|^2+\sum_{j=1,\ldots,n}w_{ij}^2\sum_{q=t+1,\ldots,m}\lambda_q^c\delta(x_{jq}^c,p_{jq}^c)\right].$$

当 *J*(*W*,*P*)最小时,聚类结果最优.NFWFCA 还可以将模糊 *K*-均值、*K*-modes 和 *K*-原型等算法合而为一.当 $\lambda^c=0$ 时,对应加权模糊 *K*-均值算法;当 $\lambda^r=0$ 时,对应加权模糊 *K*-modes 算法;当 $\lambda^c\neq0$ 且 $\lambda^r\neq0$ 时,对应加权模糊 *K*-原型算法.

通过各种实际数据集的测试,实验结果表明,该算法的聚类结果较之传统模糊 *K*-均值算法、*K*-modes 算法和 *K*-原型算法要更准确、更高效.同时,该算法还可以分析各维特征对聚类的贡献度,有效进行特征提取和优选,这对聚类算法研究及其应用都有一定的意义.

2007 年,Cai 等人[27]结合局部空间和灰度信息,提出快速通用 FCM 聚类算法 FGFCM,其特点为:(1) 用一个新因子 S_{ij} 作局部(空间和灰度)相似性度量,不仅确保图像的抗扰性、保留图像细节,而且除去了经验调节参数

α;(2) 分割时间只与灰度级数 q 有关,与图像大小 $N(>>q)$无关,因此,其聚类时间复杂性由 $O(NcI_1)$减少到 $O(qcI_2)$,其中,c 为聚类数目,I_1 和 $I_2(<I_1)$分别为 FCM 和 FGFCM 的迭代次数;(3) FGFCM 作为一个通用框架,可用于图像分割的很多其他算法,快速 FCM,EnFCM,FGFCM_S1 和 FGFCM_S1 等均可作为其特例被导出.关于合成和真实世界图像所进行的实验表明,FGFCM 是通用的、简单的,并且适合于有噪声和无噪声的多种类型图像;另一方面,FGFCM 是快速的,适合大幅灰度图像.Cai 等人指出,进一步的研究工作包括算法的聚类有效性、自适应决定聚类数量以及图像增益场评估等其他应用研究.

2.2.6 图论算法

1999 年,Jain[3]指出著名的图论分裂聚类算法的主要思想是:构造一棵关于数据的最小生成树(minimal spanning tree,简称 MST),通过删除最小生成树的最长边来形成类.基于图论的聚类算法主要包括:Random Walk,CHAMELEON, AUTOCLUST[28−30]等.

2007 年,Li[31]提出一种基于最大θ距离子树的聚类算法 MDS_CLUSTER,使用阈值剪枝,剪掉最小生成树中所有长度大于阈值$\theta\geq 0$ 的边,从而生成最大θ距离子树集,其中每个最大θ距离子树的顶点集正好形成一个类.该算法的特点是:能发现任意形状非重叠的类,只要简单说明一个参数,该参数系指每个类中最少应包含的元素个数;还能提供一个分层体系结构中几个主要的类层次,这不同于由传统层次聚合方案所生成的包括所有层次的分层体系结构.此外,该算法能将小类中的元素作为数据集中的奇异值检测出来,如果奇异值数量相对大,则将这些奇异值合并成一个新类(称为背景类).模拟实验表明了该聚类方案的有效性.

2.3 基于网格和密度的聚类算法

基于网格和密度的聚类方法是一类重要的聚类方法,它们在以空间信息处理为代表的众多领域有着广泛应用.特别是伴随着新近处理大规模数据集、可伸缩的聚类方法的开发,其在空间数据挖掘研究子域日趋活跃.

与传统聚类算法不同:基于密度的聚类算法,通过数据密度(单位区域内的实例数)来发现任意形状的类簇;基于网格的聚类算法,使用一个网格结构,围绕模式组织由矩形块划分的值空间,基于块的分布信息实现模式聚类.基于网格的聚类算法常常与其他方法相结合,特别是与基于密度的聚类方法相结合.

2001 年,Zhao 和 Song[32]给出网格密度等值线聚类算法 GDILC.密度等值线图能够很好地描述数据样本的分布.算法 GDILC 的核心思想——用密度等值线图描述数据样本分布.使用基于网格方法计算每一个数据样本的密度,发现相对的密集区域——类(或称为类簇).GDILC 具有消除奇异值和发现各种形状的类的能力,它是一种非监督聚类算法.他们的实验表明,GDILC 算法具有聚类准确率高和聚类速度快等特点.

2004 年,Ma[33]提出一种新的基于移位网格概念的基于密度和网格的聚类算法 SGC.SGC 是一种非参数类型的算法,它不需要用户输入参数,它把数据空间的每一维分成某些间隔以形成一个数据空间的网格结构.基于滑动窗口概念,为获得一个被更多描述的密度剖面引入了整个网格结构的移位概念,因此能够提高聚类结果的精度(准确度).与许多传统算法相比,该算法是高效的,因为类数据是基于网格单元的.该算法的主要优点可概括为:计算时间与数据集样本数无关;在处理任意形状类簇时展现了极好的性能;不需要用户输入参数;当处理大型数据集时,很少遇到内存受限问题.

2005 年,Pileva 等人[34]提出一种用于大型、高维空间数据库的网格聚类算法 GCHL.GCHL 将一种新的基于密度——网格的聚类算法和并行轴划分策略相结合,以确定输入数据空间的高密度区域——类簇.该算法能够很好地工作在任意数据集的特征空间中.GCHL 的主要特点为:(1) 只对数据扫描一次;将大型数据集划分成子部分,使用有限内存缓冲区一部分接一部分地进行处理;(2) 将类簇看成是由数据空间中的低密度区域划分的对象密集区域,能发现任意形状的类簇;(3) 能发现奇异值,对噪声数据不敏感;(4) 将数据空间量化为用于形成网格数据结构的有限数量的单元,所有的聚类操作都在网格结构上进行;聚类快速,聚类时间独立于数据对象数目和数据次序;(5) 适合大型、高维数据集的聚类.

Pileva 等人的实验结果表明,该算法所获得的聚类结果是高质量的,具有发现凹/更深、凸/更高区域的能力,对奇异值和噪声的稳健性以及极好的伸缩性,这使其能够很好地应用于医疗和地理领域.

2006 年,Micro 等人[35]面向移动对象轨迹数据处理领域,基于简单的轨迹间距离概念,提出了一种基于密度

的自适应聚类方法 TFCTMO,进一步考虑时态内在语义,给出时间聚焦方法以提高轨迹聚类效果.Mirco 等人将对象间的空间距离概念扩展到轨迹间的时空距离概念,由此将基于密度的聚类方法应用到轨迹上.Mirco 等人的关键思想是,将时态信息和空间信息相结合,使时态信息在移动对象轨迹聚类中起到了重要作用:根据所选取的时间区间的大小,轨迹间的相关程度是不同的.时间聚焦(temporal focusing)方法能够发现最有意义的时间区间,提高了移动对象轨迹聚类的质量.

2007 年,Derya 等人[36]对 DBSCAN(density-based spatial clustering of applications with noise)进行了与辨识核对象、噪音对象和邻近类簇相关的 3 个边缘扩展,进而提出一种新的基于密度的聚类算法 ST-DBSCAN(spatial-temporal DBSCAN).与现有的基于密度聚类算法相比,该算法具有依据非空间值、空间值和时态值发现类簇的能力.

2.4 其他聚类算法

2.4.1 ACODF 聚类算法

2004 年,Tsai 等人[37]提出一个新颖的具有不同偏好的蚁群系统(novel AS)——ACODF(a novel data clustering approach for data mining in large databases),用来解决数据聚类问题(当时未见用于数据聚类的 ACO(ant colony optimization)算法的报道).设计一种不需要求解任何硬子问题(any hard sub-problem),但能给出近似最优解的聚类算法,是人们所期待的.ACODF 能够快速获得最优解,它所包含的 3 个重要策略介绍如下:

(1) 应用不同偏好的(favorable)ACO 策略.每个蚂蚁只需访问全部城市数的十分之一,并且访问城市数目逐次减少;几次循环之后,两点间相对短的路径的信息素浓度增加,两点间相对长的路径的信息素减少.因此,蚂蚁喜欢访问距离近的节点,并用自己的信息素加强此路径(由其喜欢访问的节点组成);最后形成具有较高浓度的路径,即聚类完成.

(2) 为减少获得局部最优解所需要访问的城市数量,对蚁群采用模拟退火策略.为此设计了两个公式:

$$ns(t+1)=ns(t)\times T,$$

其中,ns 是蚁群在 T_0 函数期间访问的节点数,$ns(t+1)$表示当前蚁群的访问的节点数,$ns(t)$表示上一次循环蚁群访问的节点数,T 是一个常数(T=0.95).

$$nf(t+1)=2\times ns(t)/3-i\times ns(t)/(run\times 3),$$

其中,nf 是蚁群在 T_1 函数期间访问的节点数,$nf(t+1)$表示蚁群当前访问的节点数,$nf(t)$表示上一次循环蚁群访问的节点数,run=2,$i\in\{1,2\}$.

(3) 使用锦标赛(tournament)选择策略.与传统 ACO 不同,ACODF 采用锦标赛选择技术进行路径选择.即从 N 条路径中随机选择 K 条路径,再从这 K 条路径中选择最短路径($N>K$).

Tsai 等人分别进行了模拟和实际数据实验.模拟数据实验:首先选含 579 个数据的数据集,分别用 ACODF,GKA 和 FSOM+K-means 等 3 种算法进行非球形聚类;然后选含 300 个数据的数据集,依次用上面 3 种算法进行球形聚类.实际数据实验:采用 732 个客户信用卡上的 8 维实际数据,根据客户收入和消费进行聚类.实验结果表明,大多数情况下,ACODF 的搜索速度比 GKA 和 FSOM+K-means 更快,且错误率比它们更小.

3 实 验

为了对有一定代表性的聚类算法给出进一步的分析,我们从重点介绍的 19 种算法中选出 8 种算法,从 UCI 机器学习数据集储存库中选择了人们常用的 5 个数据集,分别针对分类属性数据和数值型数据对这 8 种算法进行了对比实验.实验的计算机环境为:处理器为 Pentium M 1.4GHz,内存 512MB,硬盘 80G,操作系统为 Windows XP,编程语言为 VC 6.0.

3.1 数据集

本文采用 Iris,Wine,Soybean,Zoo 和 Image 数据集作为测试数据集,前 4 个数据集为常用的知名数据集,已知其聚类结果可靠、并取得一致意见,适合做聚类分析的基准数据集.本文选用 Image 数据集的主要目的是与 Iris

和 Wine 这两个基准数据集进行比较.

针对数值型数据,分别采用 Iris,Wine 和 Image 等 3 个数据集进行测试.

Iris 包含 3 个类,每类各有 50 个元素,每一类代表一种类型的鸢尾花,150 个样本在 3 个类簇中分布均匀;其中,一类与另外两类线性可分,另外两类有部分重叠.Wine 数据集具有好的聚类结构,它包含 178 个样本,13 个数值型属性,分成 3 个类,每类中样本数量不同.Image 取自 UCI 机器学习数据集,本文作者在众多文献中未见其被使用.该数据集是从包含 7 个户外图像集合的数据库中随机选取的,并采用手工进行分类.

针对分类属性数据,分别采用 Soybean 和 Zoo 数据集进行测试.

Soybean 数据集共有 47 个样本,具有 35 个属性,分为 4 类,是线性可分的,其所有属性都可作为分类属性.Zoo 数据集共有 101 个记录,分为 7 类,是线性不可分的.在 Zoo 中,由 16 个属性来描述样本,其中 15 个为布尔属性值{0.1}和 1 个分类属性值属性(腿的数量){0,2,4,5,6,8}.

3.2 针对分类属性数据的实验

针对分类属性数据聚类,我们对 *K*-modes 算法、迭代初始点集求精 *K*-modes 算法分别采用线性可分大豆疾病数据和线性不可分动物园数据进行 20 次随机实验.

3.2.1 大豆疾病数据实验

大豆疾病数据实验结果:我们采用 Sun 等人[19]提出的计算正确率的方法.正确率计算公式为

$$r=\sum_{i=1,\ldots,k}(a_i/n).$$

a_i 是出现在第 i 个类簇(执行算法得到的)及其对应的类(初始类)中的样本数,k 是类数(这里有 k=4,聚类数),n 是数据集中样本总数(即 47).实验结果见表 1 和表 2.

Table 1 Clustering results of 20 random tests for soybean disease data set on 2 algorithms

表 1 两种算法对大豆疾病数据集 20 次随机实验聚类结果

Cases / Algorithm / Accuracy (%)	*K*-modes	Iterative initial-points refinement *K*-modes
98	5	7
94	6	8
89	0	3
77	0	1
70	7	1
68	2	0

Table 2 Average run time of 20 random tests for soybean disease data set on 2 algorithms

表 2 两种算法对大豆疾病数据进行 20 次随机实验的平均运行时间

Algorithm	Average running time (s)
K-modes	0.008 173 31
Iterative initial-points refinement *K*-modes	0.011 782 65

从大豆疾病数据集的实验结果来看,迭代初始点集求精 *K*-modes 算法明显好于 *K*-modes 算法,两者的平均正确率分别为 92.6%和 84%.从算法运行时间来看,迭代初始点集求精 *K*-modes 算法所需时间略长.

3.2.2 动物园数据实验

下面对 *K*-modes 算法和迭代初始点集求精 *K*-modes 算法,用动物园数据[21]进行 20 次随机实验,实验结果见表 3.聚类正确率计算公式为 r=1–(错分样本个数/样本总数),且以下实验均采用该正确率计算公式.

Table 3 Clustering results of 20 random tests for zoo data set on 2 algorithms

表 3 两种算法对动物园数据进行 20 次随机实验的聚类结果

Algorithm	Average mistaken partition numbers (include mammalia)	Average accuracy (%)	Average running time(s)
K-modes	15	85.00	0.010 460 660
Iterative initial-points refinement *K*-modes	13	86.45	0.027 089 395

从以上实验结果可以得出,大豆集的分类效果整体好于动物园数据集,这与大豆集数据线性可分而动物园数据线性不可分是一致的.对于大豆集和动物园两个数据集,迭代初始点集求精 *K*-modes 算法的分类正确率都好于 *K*-modes 算法,这说明初始化时选择一个接近真实 modes 的初始值,通过不断迭代更容易得到正确的聚类结果.另外,从运行时间来看,迭代初始点集求精 *K*-modes 算法的运行时间比 *K*-modes 算法长一些.

3.3 针对数值型数据进行实验

3.3.1 层次聚合算法和 *K*-means 算法比较

针对数值型数据,我们分别采用层次聚合算法中的单一联接法、完全联接法、类间平均联接法、沃德法和划分式聚类算法中的 *K*-means 算法,用 UCI 中的数据集 Iris,Wine,Image[29]随机进行了 20 次聚类实验,对比结果见表 4.

Table 4 Clustering results of 20 random tests for Iris ,Wine, Image data sets on several algorithms

表 4 几种算法对 Iris,Wine,Image 数据集 20 次随机实验的聚类结果

Algorithm	Average accuracy of running 20 cycles (%)			Average running time (s)		
	Iris	Wine	Image	Iris	Wine	Image
Nearest neighbor	68.00	42.70	30.00	1.583 102 5	3.134 614 5	5.241 43
Furthest neighbor	84.00	67.40	39.00	1.504 258 5	3.143 374	5.670 8
Between groups average	74.70	61.20	37.00	1.502 659 5	3.152 568 5	5.785 28
Ward method	89.30	55.60	60.00	2.379 265	4.775 662 5	8.959 95
K-means	81.60	87.96	56.00	0.002 553 522 5	0.003 764 25	0.045 662 835

实验结果表明,传统层次聚合算法对聚类结构好的 Wine 数据集分类结果并不理想,这与传统层次聚合算法的再分配能力差相关(即若在初始阶段把一些数据分配给某个类簇,那么这些数据就不能再被分配给其他类簇);而对于 Image 数据集来讲,无论层次聚合算法还是 *K*-means 算法都基本上不能对其进行正确分类,这可能与 Image 数据集的聚类结构等有关系;*K*-means 的运行效率远高于传统层次聚合算法.我们还发现,聚类结果有其不可预见性,对于不同数据集合,同一算法的聚类正确率可能会大不相同;对于同一数据集合,采用不同的聚类算法,其聚类结果和效率也会有很大差异.因此在实际应用中,应根据待聚类数据集的数据类型、聚类结构(若可得到的话)选择相应的聚类算法,以取得最佳聚类效果.

3.3.2 *k* 最近邻一致性强制与保留算法 *K*-means-CP 关于不同 *K* 值的实验

选择 Iris 和 Wine 数值属性数据集,针对 *K*-means-CP 算法(采用欧式距离进行相似性计算),*K* 取 1~4,分别进行 20 次随机实验.实验结果(见表 5)表明,无论对数据集 Iris 还是数据集 Wine,都是在 *K*=3 时达到最高正确率.对于数据集 Iris,*K*=3 时正确率为 84.65%;对于数据集 Wine,*K*=3 时正确率为 64.00%.这说明 *K*-means-CP 算法对数据集的初始分类数具有一定的预测功能.此外,聚类结果在很大程度上依赖于所用相似性度量方式.

Table 5 Clustering results of 20 random tests for Iris, Wine data sets on *K*-means-CP

表 5 一致性保留 *K*-means 算法对 Iris,Wine 数据集进行 20 次随机实验的聚类结果

K	Iris		Wine	
	Average accuracy (%)	Average running time (s)	Average accuracy (%)	Average running time (s)
K=1	81.00	0.015 179 07	55.45	0.023 532 9
K=2	81.40	0.012 644 315	56.55	0.043 986 175
K=3	84.65	0.012 979 565	64.00	0.089 074 95
K=4	82.50	0.013 717 84	50.10	0.189 232 5

3.4 *K*-means算法与*k*最近邻一致强制和保留算法比较

为判断 *k* 最近邻一致强制和保留算法是否明显优于 *K* 均值(*K*-means)算法、*k*NN 一致性与聚类质量之间有何关系,本文针对 *K*-means 算法、1 最近邻一致强制和保留算法(*k*=1,简记为 cp1 算法)和 2 最近邻一致强制和保留算法(*k*=2,简记为 cp2 算法),关于 Imagine,Iris,Wine,Glass,Ionosphere 等数值型数据集进行了 20 次随机聚类实验.从聚类结果的正确率和总体质量(简称质量)两个方面来评价聚类结果之优劣.总体质量(质量)可用类间差异与类内差异之比来度量.一类簇的紧凑程度可用该类簇中每个数据到该类簇质心之间距离的平方和来刻画.整

个聚类的类簇内差、整个聚类之类簇间的差异以及总体质量则分别由下面的式(1)~式(3)来计算:

$$\sum_{v=1,\ldots,k}\sum_{x\in C_v} d(x,\bar{x}_v)^2 \tag{1}$$

$$\sum_{1\le j<i\le k} d(\bar{x}_j,\bar{x}_i)^2 \tag{2}$$

$$\sum_{1\le j<i\le k} d(\bar{x}_j,\bar{x}_i)^2 \Big/ \sum_{v=1,\ldots,k}\sum_{x\in C_v} d(x,\bar{x}_v)^2 \tag{3}$$

其中,k 为聚类结果包含的类簇数,C_v 表示类簇 v, $\bar{x}_v$ 表示 C_v 的质心, $\bar{x}_j$,$\bar{x}_i$ 分别表示类簇 j 和 i 的质心,d 为距离函数.这里的质量只有相对意义,对相同算法不同数据集"质量值"间的相互比较没有意义.实验结果见表 6.实验结果表明,从聚类正确率和总体质量来看,k 最近邻一致强制和保留算法不优于 *K*-means 算法,kNN 一致性与聚类质量无关.

Table 6 Clustering results of 20 random tests for 5 data sets on *K*-means, cp1 & cp2 algorithms

表 6 *K*-means,cp1 和 cp2 算法关于 5 个数据集的 20 次随机实验聚类结果

Imagin	Average accuracy 20 times)	Average quality 0 times)
cp1 (1N	0.623 571 428 571 428	0.778 036 750 839 380 0
cp2 (2N	0.609 523 809 523 809	0.764 753 617 717 611 0
K-mean	0.632 380 952 380 952	0.734 076 358 291 719 0
Iris	Average accuracy (20 times)	Average quality (20 times)
cp1 (1NN)	0.840 000 000 000 000	0.258 626 172 448 124 0
cp2 (2NN)	0.892 333 333 333 333	0.322 489 157 412 046 0
K-means	0.862 333 333 333 334	0.290 268 692 364 311 0
Wine	Average accuracy 20 times)	Average quality 0 times)
cp1 (1N	0.898 314 606 741 573	0.045 433 239 324 063 6
cp2 (2N	0.905 337 078 651 685	0.045 155 360 976 705 9
K-mean	0.946 910 112 359 550	0.049 098 735 880 057 5
Glass	Average accuracy (20 times)	Average quality (20 times)
cp1 (1NN)	0.511 915 887 850 467	0.400 881 509 658 679
cp2 (2NN)	0.531 542 056 074 766	0.404 061 886 906 006
K-means	0.542 523 364 485 981	0.453 522 047 430 905
Ionosphere	Average accuracy (20 times)	Average quality (20 times)
cp1 (1NN)	0.691 880 341 880 342	0.003 812 476 851 341 2
cp2 (2NN)	0.682 051 282 051 282	0.003 555 311 462 034 7
K-means	0.710 256 410 256 410	0.003 784 599 450 916 1

4 结 论

尽管聚类分析有着几十年的研究历史,众多聚类算法相继被提出、相关的应用被展开,但聚类问题仍然存在着巨大的挑战.

通过对一些比较有代表性的聚类算法的总结,可以得出如下一些结论:

大多数聚类算法都需要预先给出参数,事实上,如果没有相关知识和经验,这在多数情况下是不可行的.对于层次化聚类算法,如何找到聚合或分裂过程的有效终止条件仍然是一个开问题.由此,开展非参数聚类算法、将聚类算法与参数自动生成算法相结合、展示聚类过程等研究可能富有前景.Binary-Positive 方法(2007 年)的研究表明,将数据转换成正二进制会改善聚类结果的正确率和鲁棒性.粗聚类算法 RCOSD(2007)能够有效挖掘连续数据,并能描述类簇的主要特性,有助于理解聚类结果.

快速找到类的合理个数和较好的初始类中心点集,使算法终止于全局最优解等是划分式聚类算法的研究热点;对于 *K*-means 和 Fuzzy *C*-means 算法,还有使其适合分类属性数据集等研究课题.*K*-modes-Huang 算法适合分类属性数据,能给出类的特性描述,其对聚类数目和数据集规模都是可伸缩的,但已证明该算法经有限次迭代只能收敛于局部最优.2002 年的迭代初始点集求精 *K*-modes 算法较好地解决了 *K*-modes-Huang 算法常因初始 modes 选择差异导致聚类结果截然不同的情况.2004 年,一致性保留算法 *K*-means-CP 的作者提出将 *K* 最近邻一致性作为聚类质量的度量方法,并给出局部一致性信息能支持全局聚类目标函数优化和聚类正确率有明显改善等结果,但我们的实验结果未能支持该论文的观点和结论.2006 年,基于特征加权模糊聚类算法通过分析各维特征对聚类的贡献度,有效进行特征提取和优选,在聚类效率和准确率方面较传统模糊聚类算法都有明显提

高.2007 年,快速通用模糊聚类算法,一个通用框架,很多图像分割聚类算法都是其特例,它适合有噪声、无噪声多种类型图像和大幅灰度图像.

基于密度和网格聚类算法多用于时空信息处理、消除奇异值、发现各种形状的类簇,对噪声不敏感,适合大型、高维数据集等方面具有好的特性.网格密度等值线聚类算法 GDILC(2001)用密度等值线图描述样本分布,具有消除奇异值和发现各种形状类簇的能力.基于密度和网格的聚类算法 SGC(2004)是一种非参数类型的算法;计算时间与数据集规模无关;适于任意形状类簇.网格聚类算法 GCHL(2005)能够发现任意形状类簇和奇异值,对噪声数据不敏感;聚类快速,聚类时间独立于数据规模和数据次序,伸缩性极好;适合大型、高维数据集.基于密度自适应聚类方法 TFCTMO(2006)结合时态信息和空间信息,时间聚焦能够提高移动对象轨迹聚类质量.基于密度聚类算法 ST-DBSCAN(2007)能够综合使用非空间值、空间值和时态值实现聚类.

在很多文献中,研究者们给出了各自的聚类算法评价指标,并只给出其算法的优点.我们认为,开展聚类算法(全面、客观的)评价标准、数据集特性的描述方法等研究,不仅时机成熟,而且有着重要意义.

下面我们将给出关于文献[4]就 11 种算法和我们就 8 种算法所作的实验的分析,以作为对上述总结的补充.同时给出部分代表性算法的比较(见表 7).

Table 7 Comparative results of part typical clustering algorithms

表 7 部分代表性聚类算法比较

Algorithm	Years	Sort	Similarity measure	Paranumber	Noise	Cluster shape	Scaled, imension	Others
K-means	1967	Partition	Distance function	1	Sensitive	Hypersphere	Large, numeric	–
K-modes-Huang	1998	Partition	Category similarity measure	1	Sensitive	Sphere	Large, category	Describe cluster well
K-means-CP	2004	Partition	Distance function	1	Sensitive	Sphere	Large-Scale	*K*NN consistency is irrelevant with clustering accuracy
MDS_CLUSTER	2007	Partition	Eulidean distance	1	In-Sensitive	Arbitrary non-overlap	–	One simple parameter
Feature weighted fuzzy clustering	2006	Partition	Eulidean distance, category similarity measure	1	In-Sensitive	Sphere	Small, mix	Feature weighted
Nearest neighbor	1967	Hierarchy	Distance function	1	In-Sensitive	Filamentary	Small and midlow-dimension	–
Furthest neighbor	1967	Hierarchy	Distance function	1	–	Sphere	Small and midlow-dimension	–
Between groups average	1967	Hierarchy	Distance function	1	–	Manifold	Small and midlow-dimension	–
Sequence data rough clustering	2007	Hierarchy	S^3M	2	–	Sequence data	Large-Scale	Depict cluster feature
SGC	2004	Density	Distance function	None	In-Sensitive	Arbitrary shape	Large and midhigh-dimension	Mostly used for spatial
GCHL	2005	Grid	Eulidean distance	2	In-Sensitive	Arbitrary shape	Oversizehigh-dimension	Information processing
ACODF	2004	Others	Eulidean distance	1	–	Sphere, non-sphere	Small, high-dimension	Get optimal value fast

文献[4]对 11 种算法采用 4 个知名数据集进行实验.其中,4 个数据集由 2 个类属性数据集和 2 个数值型数据集组成,由于对 *K*-means 和传统层次化算法采用了欧式距离作为相似性度量函数,所以针对 2 个类属性数据集所得到的测试结果不宜作为分析的依据.实验结果:对所选的 2 个数值型数据集,非层次化算法的分类结果优于层次化算法;对相同数据集,不同聚类算法产生了不同的聚类结果;对同一种算法、不同的数据集,其聚类的正确率不同.

本文对 8 种算法从 UCI 中选择 4 个知名聚类分析基准数据集和 1 个不常用数据集分别进行 20 次随机实验,并采用聚类正确率和运行时间作为衡量指标分别对数值型和类属性数据集进行实验;对 *K*-means-CP 算法,

选数值型数据集,*K* 取不同值进行实验;对 *K*-means-CP 算法,选择相同数据集,用不同相似性度量方式进行测试.实验结果:对不同数据集、同一算法,其聚类正确率不相同;对同一数据集、不同聚类算法,其聚类正确率和效率会有很大差异;将 *K*-means 算法与 *K*-means-CP 算法使用不同数值型数据集进行了比较实验,结果表明,*K*-means-CP 算法丝毫也不优于 *K*-means 算法,*k* 最近邻一致性与聚类正确率无关,用 *k* 最近邻一致性刻画聚类质量是不合适的;对同一算法和同一数据集,不同的相似性度量方式,其聚类结果也不相同.

综合文献[4]和本文的实验得出的主要结论是:聚类算法的聚类结果有一定的不可预见性,在实际应用中应根据数据类型选择合适的聚类算法(和可恰当的相似性度量方式),以取得最佳的聚类效果.针对不同数据集,进一步开展聚类算法预测分类数的能力研究.

致谢 感谢刘大有教授对本文提纲和一些重点内容所给予的有益建议,感谢金弟同学对 *K*-means-CP 算法所做的编程和实验.

References:

[1] Jain AK, Flynn PJ. Image segmentation using clustering. In: Ahuja N, Bowyer K, eds. Advances in Image Understanding: A Festchrift for Azriel Rosenfeld. Piscataway: IEEE Press, 1996. 65−83.

[2] Cades I, Smyth P, Mannila H. Probabilistic modeling of transactional data with applications to profiling, visualization and prediction, sigmod. In: Proc. of the 7th ACM SIGKDD. San Francisco: ACM Press, 2001. 37−46. http://www.sigkdd.org/kdd2001/

[3] Jain AK, Murty MN, Flynn PJ. Data clustering: A review. ACM Computing Surveys, 1999,31(3):264−323.

[4] Gelbard R, Goldman O, Spiegler I. Investigating diversity of clustering methods: An empirical comparison. Data & Knowledge Engineering, 2007,63(1):155−166.

[5] Jain AK, Dubes RC. Algorithms for Clustering Data. Prentice-Hall Advanced Reference Series, 1988. 1−334.

[6] Jain AK, Duin RPW, Mao JC. Statistical pattern recognition: A review. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2000,22(1):4−37.

[7] Sambasivam S, Theodosopoulos N. Advanced data clustering methods of mining Web documents. Issues in Informing Science and Information Technology, 2006,(3):563−579.

[8] Marques JP, Written; Wu YF, Trans. Pattern Recognition Concepts, Methods and Applications. 2nd ed., Beijing: Tsinghua University Press, 2002. 51−74 (in Chinese).

[9] Fred ALN, Leitão JMN. Partitional vs hierarchical clustering using a minimum grammar complexity approach. In: Proc. of the SSPR&SPR 2000. LNCS 1876, 2000. 193−202. http://www.sigmod.org/dblp/db/conf/sspr/sspr2000.html

[10] Gelbard R, Spiegler I. Hempel's raven paradox: A positive approach to cluster analysis. Computers and Operations Research, 2000, 27(4):305−320.

[11] Zhang B, Srihari SN. Properties of binary vector dissimilarity measures. In: Proc. of the JCIS CVPRIP 2003. 2003. 26−30. http://www.ee.duke.edu/JCIS/

[12] Kumar P, Krishna PR, Bapi RS, De SK. Rough clustering of sequential data. Data & Knowledge Engineering, 2007,3(2):183−199.

[13] Huang Z. A fast clustering algorithm to cluster very large categorical data sets in data mining. In: Proc. of the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery. Tucson, 1997. 146−151. http://www.informatik.uni-trier.de/~ley/db/conf/sigmod/sigmod97.html

[14] Huang Z. Extensions to the *k*-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge, Discovery II, 1998,(2):283−304.

[15] Huang Z, Ng MA. Fuzzy *k*-modes algorithm for clustering categorical data. IEEE Trans. on Fuzzy Systems, 1999,7(4):446−452.

[16] Chaturvedi AD, Green PE, Carroll JD. *K*-modes clustering. Journal of Classification, 2001,18(1):35−56.

[17] Goodman LA. Exploratory latent structure analysis using both identifiable and unidentifiable models. Biometrika, 1974,61(2): 215−231.

[18] Huang ZX, Michael K. A note on *K*-modes clustering. Journal of Classification, 2003,20(2):257−26.

[19] Sun Y, Zhu QM, Chen ZX. An iterative initial-points refinement algorithm for categorical data clustering. Pattern Recognition Letters, 2002,23(7):875−884.

[20] Bradley PS, Fayyad UM. Refining initial points for *k*-means clustering. In: Proc. of the 15th Internet Conf. on Machine Learning. San Francisco: Morgan Kaufmann Publishers, 1998. 91−99. http://www.cs.wisc.edu/icml98/

[21] http://www.ics.uci.edu/~mlearn/databases/

[22] Ding C, He X. *K*-Nearest-Neighbor in data clustering: Incorporating local information into global optimization. In: Proc. of the ACM Symp. on Applied Computing. Nicosia: ACM Press, 2004. 584−589. http://www.acm.org/conferences/sac/sac2004/

[23] Lyer NS, Kandel A, Schneider M. Feature-Based fuzzy classification for interpretation of mammograms. Fuzzy Sets System, 2000, 114(2):271−280.

[24] Yang MS, Hu YJ, Lin KCR, Lin CCL. Segmenttation techniques for tissue differentiation in MRI of ophthalmology using fuzzy clustering algorithm. Journal of Magnetic Resonance Imaging, 2002,(20):173−179.

[25] Li J, Gao XB, Jiao LC. A new feature weighted fuzzy clustering algorithm. ACTA Electronica Sinica, 2006,34(1):412−420 (in Chinese with English abstract).

[26] Kononenko I. Estimating attributes: Analysis and extensions of relief. In: Proc, of the 17th European Conf. On Machine Learning. LNCS 784, 1994. 171−182.

[27] Cai WL, Chen SC, Zhang DQ. Fast and robust fuzzy *c*-means clustering algorithms incorporating local information for image segmentation. Pattern Recognition, 2007,40(3):825−833.

[28] Harel D, Koren Y. Clustering spatial data using random walks. In: Proc. of the 7th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining. New York: ACM Press, 2001. 281−286. http://www.sigkdd.org/kdd2001/

[29] Karypis G, Han EH, Kumar V. CHANELEON: A hierarchical clustering algorithm using dynamic modeling. IEEE Computer, 1999, 2(8):68−75.

[30] Estivill-Castro V, Lee I. AUTOCLUST: Automatic clustering via boundary extraction for mining massive point-data sets. In: Abrahart J, Carlisle BH, eds. Proc. of the 5th Int'l Conf. on Geocomputation. 2000. 23−25. http://www.geocomputation.org/2000/index.html

[31] Li YJ. A clustering algorithm based on maximal θ-distant subtrees. Pattern Recognition, 2007,40(5):1425−1431.

[32] Zhao YC, Song J. GDILC: A grid-based density isoline clustering algorithm. In: Zhong YX, Cui S, Yang Y, eds. Proc. of the Internet Conf. on Info-Net. Beijing: IEEE Press, 2001. 140−145. http://ieeexplore.ieee.org/iel5/7719/21161/00982709.pdf

[33] Ma WM, Chow E, Tommy WS. A new shifting grid clustering algorithm. Pattern Recognition, 2004,37(3):503−514.

[34] Pilevar AH, Sukumar M. GCHL: A grid-clustering algorithm for high-dimensional very large spatial data bases. Pattern Recognition Letters, 2005,26(7):999−1010.

[35] Nanni M, Pedreschi D. Time-Focused clustering of trajectories of moving objects. Journal of Intelligent Information Systems, 2006, 27(3):267−289.

[36] Birant D, Kut A. ST-DBSCAN: An algorithm for clustering spatial-temporal data. Data & Knowledge Engineering, 2007,60(1): 208−221.

[37] Tsai CF, Tsai CW, Wu HC, Yang T. ACODF: A novel data clustering approach for data mining in large databases. Journal of Systems and Software, 2004,73(1):133−145.

附中文参考文献:

[8] Marques JP,著;吴逸飞,译.模式识别——原理、方法及应用.北京:清华大学出版社,2002.51−74.

[25] 李洁,高新波,焦李成.基于特征加权的模糊聚类新算法.电子学报,2006,34(1):412−420.

孙吉贵(1962－),男,辽宁庄河人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为人工智能,约束规划,决策支持系统.

赵连宇(1984－),男,硕士生,主要研究领域为数据挖掘.

刘杰(1973－),女,博士生,讲师,主要研究领域为数据挖掘,模式识别.