一种基于网格距离的融合式聚类算法*)

凌 萍[1,2] 周春光[1] 王 喆[1]

(吉林大学计算机科学与技术学院教育部符号计算与知识工程重点实验室 长春 130012)[1]

(徐州师范大学计算机科学与技术学院 徐州 221116)[2]

摘 要 提出了一种基于网格距离的融合式聚类算法(Agglomerative Clustering algorithm based on Grid Distance, ACGD)。为规模不同的数据集分别设计了初始球状网格和初始矩形网格,并以此作为合并过程的起点。基于随机映射思想设计了网格之间的距离定义并以此完成聚类任务。ACGD的参数以自适应学习策略确定。真实数据集上的实验表明,ACGD具有良好聚类效果,具有比同类算法更高的效率和算法鲁棒性。

关键词 网格距离,融合聚类,球状初始网格,初始矩形网格,数据粒单元

Agglomerative Clustering Algorithm Based on Grid Distance

LING Ping[1,2] ZHOU Chun-guang[1] WANG Zhe[1]

(College of Computer Science, Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, Jilin University, Changchun 130012, China)[1]

(School of Computer Science, Xuzhou Normal University, Xuzhou 221116, China)[2]

Abstract Proposeed an agglomerative clustering algorithm based on grid distance, named as ACGD. ACGD starts initial aggregation from data grids instead of data points. For large-sized dataset and small-sized dataset, sphere-shaped grid and rectangular-shaped initial grid were designed respectively. The aggregating process was conducted based on a novel grid distance definition, which is developed according to random projection idea. ACGD is equipped with the self-tuning parameterization strategies. Experimental evidence of real datasets demonstrates the fine clustering performance of ACGD, and its advantage in efficiency and robustness over its peers.

Keywords Grid distance, Agglomerative clustering algorithm, Sphere-shaped initial grid, Rectangular-shaped initial grid, Data cell

1 引言

聚类是数据挖掘领域中重要的学习任务,是在无监督信息的情况下进行知识提取的过程。聚类算法完全依赖数据特征建立某种测度,将相近的数据归为一类。概括地讲,聚类算法可分为三种类型:划分式聚类算法、层次式聚类算法和基于混合模型的聚类算法。划分式方法力图找到一个对数据集的最优划分,它须事先指定簇的数目,并建立正确的划分评价体系,其典型代表是K-means[1]。层次式方法分步找到聚类结构,其算法实施过程体现为逐渐地融合数据或切分超簇。根据不同的处理方式,层次式聚类算法分为合并式算法(自底向上式方法)和分裂式算法(自上向下式方法)。聚类之后将形成表示合并或分裂过程的层次图,通过对图的分析得到聚类结果。其中合并式方法更主要,它的两种代表是基于最近邻距离的融合聚类算法[2]和基于最远邻距离的融合聚类算法[3]。分裂式聚类应用较少,支持向量聚类[4]可看成其中一种。第三种方法通过建立簇的入集判别模型完成数据聚类,数据的分布函数往往是建模的主要依据。

三种方法之中,合并式聚类不必事先指定簇的数目,只要获知数据间的相互距离关系便可进行聚类分析,这适用于一些不易获知数据的具体表达的场合。但已有的合并式聚类算法大多以单个数据点作为合并操作的起点,当面对大规模数据集,庞大的距离矩阵的存取和计算耗费对算法的性能产生很大影响。另外,现有的距离定义往往是欧式距离定义的变体,它们在低维数据空间可有效工作,然而在稀疏的高维数据空间,其所提供的距离值可能失真,从而影响聚类结果。文献[5]提出了基于信息量的距离定义,这在一定程度上克服了高维数据存在的"维数灾难"问题[6],但算法本身并未针对高维空间做特别设计,因而在处理高维数据时无法达到较优状态。

针对以上两点,本文提出一种新的基于网格距离的合并式聚类算法(ACGD)。ACGD以数据网格为合并操作的起点,从而降低了合并过程的路径长度。合并所依据的网格距离是基于随机映射思想进行定义,保证算法在高维空间也可准确提供数据间的关系信息。针对不同规模的数据集设计了两种初始网格:球状初始网格和矩形初始网格。前者由寻求最小超球体的支持向量过程生成,后者由一个数据划分过程生成。一并给出了网格定义中涉及的参数的启发式学习方案,这降低了算法的运行代价,同时增加了算法的数据适应性。

*)本文获国家自然科学基金重点项目(60433020,60673099,60773095),国家高技术研究发展计划(863计划)(课题编号:2007AA04Z114),985工程:"计算与软件科学科技创新平台"项目,以及教育部"符号计算与知识工程"重点实验室资助。凌 萍 博士研究生,主要研究方向为计算智能、支持向量技术、多关系数据挖掘;周春光 教授,博士生导师,CCF会员,主要研究方向为计算智能、机器味觉、图像处理;王 喆 博士,讲师,主要研究方向为数据挖掘、商务智能。

2 初始网格的生成

对小规模数据集，使用球状初始网格，该类网格形状灵活，并在初始就引入数据的局部特征信息，使网格中数据的类别纯度较高。但其生成过程计算耗费稍高，因此只在小数据集中使用。对大规模数据集，使用矩形初始网格，该类网格形状统一，操作简单且易于实现，生成代价低，但其所生成的初始网格中不能保证较高的类别纯度。

2.1 球状初始网格的生成

球状网格本质上是一些数据的邻域，这些数据是表述了数据集的分布特征的数据代表。本文中将从数据集中抽取那些描述了数据簇的轮廓和簇内部的重要位置——分布密度发生显著变化的位置——的数据点作为数据代表，生成邻域。设 R^n 空间中的数据集 $X(|X|=N)$，根据寻找数据轮廓的方法[4]，本文通过优化如下的目标函数找到这些数据代表：

$$\max_\gamma \sum_i \gamma_i K(x_i,x_i)-\sum_{i,j}\gamma_i\gamma_j K(x_i,x_j) \quad (1)$$

$$\text{s.t. } \sum_i\gamma_i=1,\ 0\leqslant\gamma_i\leqslant C$$

那些满足 $0<\gamma_i<C$ 的数据点在文献[4]的支持向量聚类过程中被称为支持向量(SV)，它们描述了数据簇的轮廓。本文中，它们将作为生成初始球状网格的中心点。K 为 Kernel 核函数，一般使用 Gaussian Kernel。我们对 K 做调整，使生成的 SV 不仅可描述数据簇的轮廓，还可表示簇内部的重要位置。改进的关键在于令 K 中的尺度参数从单一固定的取值改变为自适应的多样取值。即对任一数据点 x，设置其尺度因子为 $\sigma_x=\|x-x_r\|$。当衡量 x 与另一数据点 y 间的相似度时，二者的尺度因子结合起来构成 K 的尺度因子：

$$K(x,y)=\exp\left(-\frac{\|x-y\|^2}{\sigma_x\cdot\sigma_y}\right)=\exp\left(-\frac{\|x-y\|^2}{\|x-x_r\|\cdot\|y-y_r\|}\right) \quad (2)$$

r 可视为 x 所在的自然簇的大小，设置步骤为：

1) 从小到大排列 x 与其它点的距离，构成序列 $\{\|x-x_j\|\}$；2) 取 $r=\max_j\left\{\frac{\|x-x_{j+1}\|-\|x-x_j\|}{\|x-x_{j+1}\|}\right\}$。

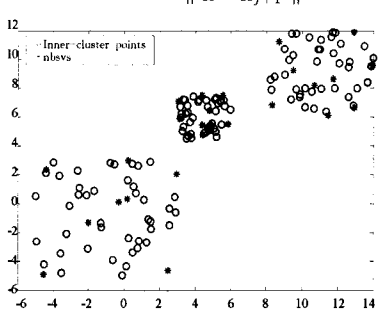

图 1 使用调整 Kernel 生成的 SVs

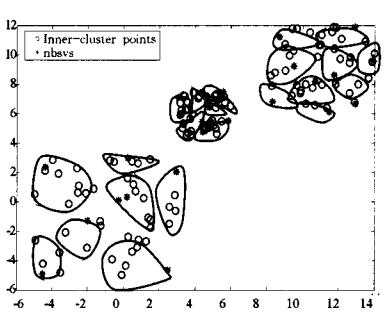

图 2 SVs 的邻域情况

将(2)应用于(1)，得到支持向量 $\{SV_i\}$。对任一 SV_i，其邻域的生成以 Kernel 相似度为依据。即对任意数据点，它将位于与之最相似的 SV_i 中。图 1 给出了使用经过改进的 Gaussian Kernel 之后生成的 SVs 的结果，这些 SVs 确实位于数据簇边缘处和簇内部分布密度发生显著改变的地方。这些 SVs 的邻域将作为球状的初始网格参与后继合并操作。图 2 给出了 SVs 的邻域情况。

2.2 矩形初始网格的生成

矩形网格通过对数据空间的划分生成。在每一维上进行区间划分，在整个数据空间上得到矩形数据网格。具体地，在 i^{th} 维划分出 2^{e_i} 个大小相等的区间，定义 $\sum_{i=1}^n e_i=e$，则整个数据空间被分成 $2e$ 个矩形网格。e_i 取决于 i^{th} 维上值的方差，方差越大，e_i 越大。e 是控制网格多少的重要参数，为保证网格中类别的纯度，其值应稍大。但网格数目多，将带来额外计算耗费。本文设置 $e=8$，各个 e_i 根据其方差比例确定。

3 网格距离定义

已有的同类定义往往利用数据的分布模型，从统计信息中提取其数据特征，这在数据分布模型未知的情况下难以工作。本文从网格中数据的整体特征提取信息作为定义网格距离的依据。具体地，将建立数据的随机映射，通过分析网格中数据在映射下的射影值来获知数据的特征。这不仅克服了分布模型未知带来的困难，还具有处理高维数据的能力。因为在维数增加时，随机映射可较好地保持数据间的相对距离关系信息，其生成的投影与用主成分分析产生的投影效果几乎一致。

本文建立 P 次随机映射，以克服随机性。每一次映射由 L 个随机数对 (d,v_d) 构成具体的映射机制。其中 $d\in[1,n]$，v_d 取自 d^{th} 维上的值域。数对 (d,v_d) 表达了一个局部约束：$x_d<v_d$，其中 x_d 是 x 在 d^{th} 维上的坐标。x 对此约束的满足情况构成了在该局部映射下的射影值。经过一次随机映射中的 L 个局部映射之后，x 对应了一个长度为 L 的布尔型射影向量。具有相同射影向量的数据定义为数据集的粒单元。一个网格可能覆盖一个或多个粒单元，这些粒单元的射影向量之并，被定义为该网格的射影值。设网格 G 覆盖了 t 个粒单元，则 G 的射影值记作 $G=\{V_1\cdots V_t\}$，其中 V_i 是 t^{th} 网格的射影向量，$V_i\in\{0,1\}^L$。

两个网格在此随机映射之下的距离被定义为它们射影值之间的差异。设两个网格 $A=\{VA_1\cdots VA_t\}$，$B=\{VB_1\cdots VB_h\}$，其中 VA_i 和 VB_i 是各自粒单元的射影向量。定义二者之间的距离为

$$D(A,B)=\frac{1}{2}\left[\exp\left(-\frac{|A\cap B|\cdot 2}{t+h}\right)+\exp\left(-\frac{1}{t+h}\sum_{i,j}(VA_i\oplus VB_j)\right)\right] \quad (3)$$

式(3)右端第一项表示了 A,B 之间的最近距离，第二项用异或操作求两网格所含粒单元之间的平均距离。当执行了 P 次随机映射，A,B 之间的距离是各次距离的平均值。具体地，设 A^I,B^I 为 A,B 在 I^{th} 次随机映射之下的射影值 $A^I=\{VA_i^I\}$，$B^I=\{VB_j^I\}$，D^I 表示二者在此随机映射之下的距离，那么 A,B 间的网格距离 GD 定义为

$$GD(A,B)=\frac{1}{P}\sum_{I=1}^{P}D^I(A,B) \quad (4)$$

其中

$$D^I(A,B)=\frac{1}{2}\left[\exp\left(-\frac{|A^I\cap B^I|\cdot 2}{|A^I|+|B^I|}\right)+\right.$$

$$\exp\left(-\frac{1}{|A^l|+|B^l|}\sum_{i,j}(VA_i^l \oplus VB_j^l)\right)] \quad (5)$$

4 ACGD 算法及参数化方案

4.1 ACGD 算法步骤

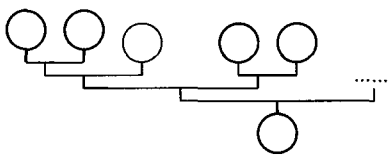

图 3 合并式聚类路径图

ACGD 算法的步骤为：1) 生成初始网格；2) 求网格间 GD 值；3) 确定 $\min\{GD(A, B)\}$，合并 A, B 得到新的网格；4) 返回 2)，直到所有网格合并为一个网格为止。A, B 合并后的新网格的射影值为双方射影值之并集。网格合并过程形成一个聚类路径图，如图 3 所示。该图是倒置的树状结构，每一树叶表示一个初始网格，树枝表示了合并过程，树的某一深度上的水平切面构成了一种聚类结果。

4.2 ACGD 参数化方案

ACGD 中涉及的主要参数除了核函数的尺度参数外，还有 L，P，d，v_d。设计 L 为取自$[\sqrt{n}, n]$的随机数，并扩展其为可变状态，即在 l^{th} 次随机映射中建立 L_l 个局部约束，L_l 为取自$[\sqrt{n}, n]$的随机数。d 是具体考察的维，它可随机生成，也可根据相应维上坐标值的方差确定。v_d 是在 d 维上实施划分的分割点，可随机生成，但本文给出如下设计来找到较好的 v_d 值：

1) 在数据集中任取 L_l 个数据点；2) 取用某一数据点 d^{th} 维上的坐标值作为 v_d 值，保证每个数据点使用且仅使用一次。用如此的方式来给出 L_l 个维上的 v_d 值，是利用数据的边缘分布进行取样，依据取样的样本值确定分割点，进而希望每次均在相应维的稠密区进行分割。

P 决定了随机映射的强度，P 越大，随机映射数目多，对数据的特征提取强度越大。但 P 不可过大，否则一方面会降低算法效率，另一方面当 P 超过一定值之后，P 的增加并不能提高距离信息的准确性。我们做了如下实验，分析 P 的较优取值范围。由高斯分布生成一个 5 维数据集，含 1000 个数据点。在其上生成球状初始网格。若视网格为集合，可用笛卡尔积计算网格距离，并将此距离与用 GD 计算得到的距离值进行比较。基于笛卡尔积的距离定义为

$$DD(A,B)=\exp\left(-\frac{A\times B}{|A|\cdot|B|}\right) \quad (6)$$

设由 GD 和 DD 产生的距离矩阵分别为 M_{GD} 和 M_{DD}，我们定义 $z = \exp(-\|M_{GD} - M_{DD}\|_F)$来考察二者之间的差异，其中 $\|\cdot\|_F$ 定义为：

$$\|M\|_F=\frac{1}{|M|}\sum_{i,j}(M_{ij})^2 \quad (7)$$

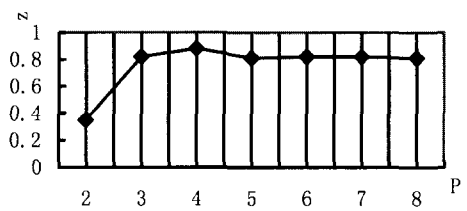

图 4 P 变化示意图

我们令 P 从 2 渐增，画出 z 随 P 变化的曲线图，如图 4 所示。由于 M_{DD} 分析了两网格中任意一对数据点间的相互关系，较全面地提取了信息，因而所形成的距离可视为较准确的距离值。在图 4 中，当 $P = 4$ 时，z 最大，即 M_{GD} 与 M_{DD} 最接近，故认为 $P = 4$ 时对应了 GD 的最佳状态。此后 P 的增大只是维持甚至降低了距离信息的准确性。

根据以上分析，我们设计如下的步骤确定 P 值：1) 在数据集中任取一些点并生成球状初始网格；2) 用 GD 和 DD 计算网格距离；3) 令 P 从 2 渐增寻找满足 $\max\{\exp(-\|M_{GD} - M_{DD}\|)\}$的 P 值。

5 实验

根据初始网格的两种形状，我们把基于球状初始网格和初始矩形网格的聚类方法分别命名为 SACGD 和 RACGD。首先比较二者的性能。实验数据为 News Group[7]，该数据集中含有 20000 个文档(电子邮件)，被平均地分为 10 个类，我们记这些类别为

NG1：alt. atheism；NG2：comp. graphics；NG3：comp. os. ms. windows. misc；NG4：comp. sys. ibm. pc. hardware；NG5：comp. sys. mac. hardware；NG6：comp. windows. x；NG7：misc. forsale；NG8：rec. autos；NG9：rec. motorcycles；NG10：rec. sport. baseball；NG11：rec. sport. hockey；NG12：sci. crypt；NG13：sci. electronics；NG14：sci. med；NG15：sci. space；NG16：soc. religion. christian；NG17：talk. politics. guns；NG18：talk. politics. mideast；NG19：talk. politics. misc；NG20：talk. religion. misc.

取用其中的若干类构成多个实验数据集：

1) {NG1，NG2，NG7} (300)
2) {NG1，NG2，NG7} (400)
3) {NG1，NG2，NG7} (500)
4) {NG3，NG7，NG8} (200)
5) {NG3，NG7，NG8，NG12} (200)
6) {NG3，NG7，NG8，NG12，NG16，NG18} (150)
7) {NG2，NG3，NG4，NG5} (200)
8) {NG8，NG9，NG10，NG11} (150)
9) {NG17，NG18，NG19，NG20} (200)

括号中的数字是从各类中随机取出的样本数。1)－3)是同种类别组合，数据规模渐增；4)－6)是相同数据规模，类别渐增；7)－9)的实验数据集中类别之间有重叠，类别边界不清晰。本文使用 $tf.idf$ 方法把每个文档表示为向量，并对文档向量做标准化。那些出现此数较少的词被删除。表 1 列出了 SACGD 和 RACGD 的聚类准确率。

表 1 两种初始网格效果对比(%)

	1)	2)	3)	4)	5)
SACGD	84.9	84.7	85.0	84.1	80.6
RACGD	83.7	84.1	84.8	81.4	80
	6)	7)	8)	9)	
SACGD	78.3	67.2	69.8	66.7	
RACGD	76.9	65.7	66.3	64.9	

整体分析表 1 中的实验结果可看出，SACGD 优于 RACGD，这归功于 SACGD 初始网格的灵活形状，以及网格中数据较高的类别一致性。对于同种数据组合，当数据规模增大，二者的聚类准确率均提高。前者性能的提高是由于数据信息变得丰富而全面，后者性能的提高源于数据信息的丰

富而使得网格质量更优，而且后者提高的幅度更大，正如1)到3)上的实验结果所示。

对于相同规模的数据集，当类别增多，两种方法的聚类性能有所下降，而且RACGD的下降程度稍大。在7)－9)中，SACGD比RACGD的表现更好，这体现了球状初始网格对数据的适应性。即使是在类别边界模糊的场合，SACGD可通过提取局部分布特征生成较好的初始网格。而RACGD的初始网格在一定程度上是通过盲目划分产生，这使得在网格合并之初便携带了潜在的错分数据，从而影响后继聚类结果。但SACGD的性能是以较大的计算耗费为代价的。图5记录了两种方法在1)－6)的数据集上的时间耗费。SACGD计算耗费明显比RACGD要高，随着数据规模的增大，SACGD时间耗费的增长速度快于RACGD。

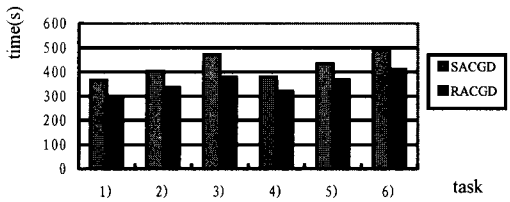

图5　SACGD和RACGD时间耗费对比

表2　数据集基本信息

	维数	数据集规模	实验集规模	类别数
Iris	4	150	150	3
Liver	7	345	345	2
Letter	17	20000	1500	26
Waveform	21	5300	1800	3
Musk1	16	476	476	2

第二部分实验比较ACGD与基于欧式距离的合并式聚类方法(命名为EDC)以及基于广义信息距离的合并式聚类方法[5](IDC)。一些流行的聚类方法也参与比较：k-means[1]，NJW[8]，SVC[4]。其中NJW是频谱聚类方法，其思想是从数据的相似度信息中提取数据的频谱坐标，在频谱空间中完成聚类。数据集选用UCI[9]：Iris，Liver，Letter，Waveform，Musk，其基本信息列于表2中。其中Letter数据集有20000个数据，本文从'A'－'J'前10个类中任取150个数据构成1500个点的实验数据集。Waveform数据含300个训练数据、5000个测试数据，本文从测试数据中随机抽取1800个点构成实验数据集。Musk数据集有两个版本，Musk1和Musk2，我们取用Musk1。各种方法的聚类结果列于表3。

表3　聚类结果对比

	Iris	Liver	Letter	Waveform	Musk1
k-means	95.8	71.7	86.9	76.3	89.0
NJW	97	73.7	90.1	79.6	92.6
SVC	96.7	73.1	87.1	77.6	91.1
EDC	96.8	71.6	88	78.7	91.8
IDC	97	73.8	89.5	80.8	93.6
SACGD	97	73.5	90.1	81.3	94.6
RACGD	97	72.8	88.2	80.3	93.8

分析EDC，IDC，SACGD，RACGD4个合并式聚类方法的结果发现，在前两个低维数据集中，IDC稍胜一筹。原因在于它提取了全面的数据信息进行距离定义，参与合并的数据粒单元是单个数据点，合并过程中拥有完整的数据特征信息。IDC与SACGD的表现接近，可见虽然SACGD是以较大的数据粒单元——网格——做数据合并，但由于其网格的合理性及良好的距离定义达到了用较小数据粒单元进行合并所产生的效果，这证明了SACGD方法的有效性。EDC和RACGD互有胜负，前者性能稍低，原因在于其僵硬的欧式测度定义与非均匀分布的数据集之间的不匹配。RACGD失误的原因在于其初始网格的质量不高。

在后3个高维数据集上，SACGD的聚类性能优于IDC和EDC，这种优势归功于以随机映射为思想的网格距离定义，能有效避免高维数据空间稀疏性引起的维数灾难问题，准确刻画高维数据的相对距离关系，从而产生较好的聚类效果。EDC中的欧式距离难以提供高维数据间的正确距离信息，所以性能差距明显。IDC虽将距离定义建立于信息熵之上，但其定义仍依赖于数据坐标的统计量。即它对数据坐标做无权重的使用，这类似于欧式距离定义对坐标的处理方式，因而也产生不佳的聚类结果。RACGD虽然逊于SACGD，但其中随机映射机制的运用使其在解决高维数据时表现出比IDC更优的性能。

再分析前3种聚类方法。其中，NJW呈现出最佳的聚类性能，它在低维数据集上与IDC相近，这说明NJW实施的频谱空间变换可有效地抽出数据的真实分布结构，较好地完成聚类。在高维数据集上NJW优于EDC，这是因为NJW是在频谱空间中进行分析，类别区分信息在此空间中能够有更清晰的表达。k-means完全基于欧式距离，聚类能力较低，该方法需要事先指定簇的数目，也为算法执行带来困难。SVC含有对数据空间的变换，在这一点上它与NJW等价，但在后继的簇辨认过程中含有大量随机性，这影响了最终聚类的结果，所以性能一般。

表4　三种网格定义的比较

	1)	2)	3)	7)	8)	9)
D_n	83.8	83.9	84.3	64.3	67.3	63.4
D_f	84.3	84.1	84.2	65.6	7.3	64.7
GD	84.9	84.6	85.2	67	69.5	66.8

为深入观察GD质量，将其与另外两种集合间距离定义(最近邻定义D_n[2]和最远邻定义D_f[3])做比较：

$$D_n(A,B)=\min\{\|a-b\|\ |\ a\in A,b\in B\}$$

$$D_f(A,B)=\max\{\|a-b\|\ |\ a\in A,b\in B\}$$

数据集选用前述News Group的1)－3)和7)－9)共6个数据集，合并起点为球状初始网格。3种方法的实验结果列于表4中。在实验中，GD的表现优于D_n和D_f。对于1)－3)，数据簇之间有相对清晰的边界轮廓，D_n和D_f的工作能力相近，但D_n的定义决定了其受孤立点和数据集上微小扰动的影响比D_f更大，因而其性能不如D_f稳定。在7)－9)上，数据簇的轮廓相对模糊，GD的优势更加明显，此时D_n和D_f均因为簇之间存在重叠区域，而在给出网格距离时包含了不准确的因素。在本文的实验中，D_f性能比D_n稍好且稳定，这是因为D_f在寻求网格间的最远距离时，间接地将网格间每一对数据点的距离信息考虑在内，所形成的距离比D_n形成的距离含有更全面的信息。但D_n也有其适用的场合，在一些具有特别形状，如线状分布的数据集上，D_n则会显示出特别的优势。上述实验中，GD的表现证明了其定义的有效性。

结束语　本文提出了一种新的层次式聚类算法ACGD。ACGD以数据网格为合并起点，用网格距离作为合并依据，用

(下转第231页)

```
Rear1←Rear;{排序尾端(结点)控制指针}
当 Visit1<>Rear1 @:\\{当排序遍数未完时}
Place1←Visit1;{标记当前范围最小者初始处}
Min←Visit1.Data;{标记当前范围最小者初始值}
Max←Min;{最小者初始值兼作最大者初始值}
Place2←Visit1;{最小者初始处兼作最大者初始处}
Visit2←Visit1.Succeed;{从下一结点起,择选当前最小、最大者}
当 Visit2<>Rear1.Succeed @:\\{当尚有未选结点时}
如果 Visit2.Data<Min{当前结点数据更小吗?}
T:\\ Min←Visit2.Data;Place1←Visit2 //{标记当前最小者及其位置}
F:如果 Visit2.Data>Max{当前结点数据更大吗?}
T:\\ Max←Visit2.Data;Place2←Visit2 //{标记当前最大者及其位置}
Visit2←Visit2.Succeed{指向下一待访结点}
//
如果 Place1<>Visit1{当前最小者不在其位吗?}
T:\\ Place1.Data←Visit1.Data;Visit1.Data←Min //{使当前最小者直接到其位}
如果 Place1.Data=Max{当前最大者恰在交换后处吗?(当前最大者已不在 Place2 处)}
T:Place2←Place1{标记第 i 大者新位(警告:必须使当前最大者又在 Place2 处)}
//;
如果 Place2<>Rear1{当前最大者不在其位吗?}
T:\\ Place2.Data←Rear1.Data;Rear1.Data←Max //{使当前最大者直接到其位}
Visit1←Visit1.Succeed{从头向中(心)步进:指向下一遍排序范围的始端结点}
如果 Visit1<>Rear1{当前排序范围内并非只剩最后两个结点吗?}
Rear1←Rear1.Precede{从尾向中(心)步进:指向下一遍排序范围的尾端结点}
//;
{"输出双向链表中递增有序化各数据"的处理}
Visit1←Head;{指针 Visit1 标记双向链表头结点}
当 Visit1<>Null @:\\{当访问指针 Visit1 非空指针时}
输出 Visit1.Data;{输出当前结点数据字段}
Visit1←Visit1.Succeed{从头向尾步进:指向下一结点位置}
//;
行输出;
!!!
```

结束语 依据同构化基本定理。本文研究和发现了基于传统内部首尾排序算法的同构化特点与本质;利用其同构化特点与本质,逐个更优地构造了基于链表的结点插删、结点标记、(结点数据字段)变量标记的 3 种首尾换排序新算法,有利于进一步深化和推广首尾排序算法的应用方式与实用范围。并且,这些基于链表的首尾排序算法及其编程实现,完全可作"程序设计"与"数据结构"的课程融合、教学改革、教育创新的一个有效案例,而这已被由作者成功主持与主研的 2006 年四川省级精品课程"数据结构"的教学研究与教改实践所证实。

参 考 文 献

[1] Knuth D E. The Art of Computer Programming, Volume 1, Fundamental Algorithms. Addison-Wesley Publishing Company, Inc., 1973

[2] Knuth D E. The Art of Computer Programming, Volume 3, Sorting and Searching. Addison-Wesley Publishing Company, Inc., 1973

[3] 周启海. C++同构化对象程序设计原理. 北京:清华大学出版社,北方交通大学出版社,2004

[4] 严蔚敏. 数据结构(C 语言版). 北京:清华大学出版社,2002

[5] 黄涛. 基于链表的择换排序新算法——"数据结构"与"程序设计"课程的融合创新案例研究[J]. 计算机科学,2008,35(11)

[6] 周启海. C 语言程序设计教程. 北京:机械工业出版社,2004

[7] 周启海,李朔枫,杨祥茂,等. 论程序设计课程教学中的同构化创新思想教育——"对→好→巧→妙→绝"的算法案例[J]. 计算机科学,2007,35(5)

(上接第 163 页)

SV 技术和数据划分技术生成两种形状的初始网格,用随机映射技术定义网格间的距离。实验表明所设计的网格距离定义具有较好的数据特征捕捉能力,ACGD 算法具有较好的聚类性能。

但是合并过程的控制仍是待解决的问题。在一个庞大的合并路径图上通过分析得出合理的簇划分的结果仍是较困难的任务,这也降低了聚类效率。如何评价合并状态的优劣来适时地终止合并操作是下一步的工作方向。

参 考 文 献

[1] Bradley P S, Fayyad U M. Refining Initial Points for k-Means Clustering // Proceedings of 15th International Conference on Machine Learning. San Francisco, USA, 1998:91-99

[2] Lance G N, Williams W T. A general theory of classificatory sorting strategies . 1. Hierarchical systems. The Computer Journal, 1967, 9(4): 373-380

[3] Dan K, Sepandar D K, Christopher D. Manning from Instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering // Morgan K, ed. Proceedings of the Nineteenth International Conference on Machine Learning. San Francisco, USA, 2002:307-314

[4] Ben-Hur A, Horn D, Siegelmann H T. Support Vector Clustering. Journal of Machine Learning Research, 2001, 2:125-137

[5] 丁世飞,史忠植,靳奉祥,等. 基于广义信息距离的直接聚类算法. 计算机研究与发展, 2007(4):674-679

[6] Novak E, Ritter K. The Curse of Dimension and a Universal Method for Numerical Integration // Nürnberger G, et al., eds. Multivariate Approximation and Splines. Germany, 1997:177-188

[7] http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html

[8] Ng A, Jordan M, Weiss Y. On Spectral Clustering: Analysis and Algorithm // Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2002

[9] http://www.uncc.edu/knowledgediscovery