基于改进自适应聚类算法的RBF神经网络分类器设计与实现

郝晓丽 张 靖
(太原理工大学计算机科学与技术学院 太原 030024)

摘 要 针对传统径向基函数神经网络构造的网络分类器通常存在分类精度不高、训练时间长等缺陷,首先提出了一种改进的自适应聚类算法,用于确定分类器的隐含层节点。该算法通过筛选基于轮廓系数的优秀样本群,来寻找最佳初始聚类中心,避免了传统K-means算法易受初始聚类中心点影响,导致最终的分类效果严重偏离全局等情况的发生。其次,将该改进算法用于构造径向基函数神经网络分类器和快速有效地确定隐含层节点径向基函数中心及函数的宽度。最后,通过大量UCI数据集的实验和仿真,验证了改进算法在聚类时间、聚类轮廓系数及聚类正确率等方面具有优越性。同时,大量的仿真实验也证明了基于改进算法构造的RBF分类器具有更高的分类精度。

关键词 聚类,K-means,径向基函数神经网络
中图法分类号 TP301.6 **文献标识码** A

Design and Realization of RBF Neural Network Classifier Based on Advanced Self-adaptive Clustering Algorithm

HAO Xiao-li ZHANG Jing
(College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan 030024, China)

Abstract Owing to defects of lower classification precision and longer training time of Radial Basis Function Neural Network (RBF) classifier, a new self-adaptive clustering algorithm was produced firstly, which can be applied into construction of nodes in implicit layer. The new algorithm optimizes initial cluster centers by choosing good samples based on silhouette coefficients. It not only avoids the effects of initial centers in traditional k-means, but also avoids classification deviation. Secondly, the new algorithm was introduced into designing of RBF classifier. It can ascertain centers of radial basis function and its width efficiently. Finally, by a large number of tests and simulation, the new clustering algorithm was testified to be superior in clustering time, silhouette coefficients and accuracy rate. Besides, RBF classifier based on the advanced algorithm was proved to have higher precision.

Keywords Clustering, K-means, Radial basis function neural network

1 引言

径向基函数神经网络(Radial Basis Function Neural Network,RBFNN)是由输入层、隐含层和输出层构成的一种特殊三层前馈神经网络,具有结构简单、非线性逼近能力强、收敛速度快以及全局收敛等优点,被广泛应用于智能控制、系统优化、信号及信息处理、模式识别等领域[1-3]。

RBF分类器的结构设计对其性能有重大的影响,而影响其结构的参数主要是隐含层节点的设定。RBF神经网络的隐含层也可称为特征抽取层,是输入模式的"内部表示":它将某类输入模式中所含的区别于其它类别的特征抽取出来,同时将这些特征传递给输出层[4,5]。隐含层节点选取包括节点数目、函数的中心、函数的宽度及隐层与输出层之间的联接权值。其中,隐含层中心位置的选取也是十分关键的,不恰当的隐含层中心位置会使网络无法正确反映输入样本空间的实际分布情况,不能很好地对输入空间进行拟合[6]。隐含层中心的宽度也是影响RBF神经网络性能的重要因素。当宽度过小时,类与类之间的界线就会变得比较模糊,从而降低分类的精度;当宽度过大时,基函数覆盖的区域相对就会比较小,从而降低网络的泛化能力。同时,隐含层节点的数量也影响着神经网络的复杂度以及泛化能力。若隐含层节点数量过少,则会导致模型的描述能力不足,出现欠拟合现象;而隐含层节点数量过多,就会出现过拟合现象,同样使其泛化能力降低。因此构造RBF神经网络分类器时,在隐节点选取过程中,常见的算法有固定法、随机选取法、遗传算法、K-means算法等[7-10]。文献[7]所采用的固定法中,将隐含层中心的数目设定得与训练样本的数目相同,使得该网络的拟合精度高,但当训练数据含有噪声时,泛化能力就会很差。文献[8,9]所采用的进化算法用于RBF神经网络的学习效果较好,但其计算量较大,训练时间较长。基于聚类分析的RBF神经网络可用于大数据、多样本、无明确函数的复杂问题处理,能够克服高维数据学习所造成的训练时间长、泛化能力差等缺点,其中K-

到稿日期:2013-07-30 返修日期:2013-12-03 本文受山西省自然科学基金项目(2013011016-3)资助。
郝晓丽(1973—),女,博士,副教授,硕士生导师,主要研究方向为人工智能、进化计算、图像处理、模式识别,E-mail:haoxiaoli@tyut.edu.cn;
张 靖(1985—),男,硕士生,主要研究方向为人工智能、进化计算等。

means 算法是一种简单易行的方法，然而由于数据集中孤立点和噪声点的存在，使得聚类中心容易偏离真正的数据密集区[10]。

由此可以看出，快速有效地确定隐含层节点的径向基函数中心 C_p 及函数的宽度是构造 RBF 神经网络分类器的关键任务之一。为克服传统 K-means 算法聚类的结果易受初始聚类中心影响的缺陷，本文提出了一种改进的自适应聚类算法，该算法运用轮廓系数作为聚类评定参数，在样本集中选取候选优秀样本群作为初始聚类中心，使得最初的初始聚类中心在空间分布上与数据实际的分布尽量一致。由改进聚类算法确定径向基函数的数目、基函数中心 C_p 及宽度，可以加速网络的训练速度。通过实验仿真可以看出，基于该方法设计的 RBF 分类器较基于其它算法的分类器，具有分类精度高、训练速度快等特点。

2 改进的自适应聚类算法

2.1 传统 K-means 算法的缺陷

(1)K-means 算法生成的聚类数是预先给定的，不能进行动态添加。在大多数情况下，无法预先确定给定的数据集应该划分的类别。

(2)K-means 算法对初始聚类中心依赖性比较大，若采用随机方法选取初始聚类中心，可能导致陷入局部最优值，使得最终的分类效果严重偏离全局最终分类。而当聚类数较大时，该缺点尤为明显。

(3)在 K-means 算法中，通常采用误差平方和准则函数作为聚类准则函数。若数据分布集中且各类之间区别明显，则采用误差平方和准则函数比较有效；若各类的形状及大小相差较大，则采用此方法的聚类效果较差。此外，最佳聚类结果通常是沿着目标函数减小的方向循进，由于目标函数存在着许多局部极小点，若初始化落在一个局部极小点附近，则会造成算法在局部极小处收敛。因此，初始聚类中心的随机选取尤其重要。

2.2 基于轮廓系数的自适应 K-means 算法

针对传统 K-means 算法聚类的结果易受初始聚类中心影响的缺陷，本文对随机选择的初始聚类中心的方式进行改进，尽量使最初的初始聚类中心在空间分布上与数据实际的分布相一致。

根据聚类的目的，将具有相同或相似特性的数据进行划分，若聚类效果越好，则某一类中数据之间的差异度越小，异类的差异度越大。若采用量化的标准对其进行评判，则可以使用基于距离和相似度计算的分裂函数判断法、Dunn 函数判断法、XB 函数判断法等。

轮廓系数是由 Kaufman 等人提出的基于距离的聚类效果判断方法，其中涉及到样本个体轮廓系统 O_i 和聚类轮廓系数 O，定义见式(1)－式(4)。

$$O=\frac{1}{n}\sum_{i=1}^{n}O_i \tag{1}$$

$$O_i=\frac{x(i)-y(i)}{\max[x(i),y(i)]} \tag{2}$$

$$x(i)=\frac{1}{n_c-1}\sum_{i,j\in C_c}d(i,j)\quad i\neq j \tag{3}$$

$$y(i)=\min_{p,p\neq c}[\frac{1}{n_p}\sum_{i,j\in C_c}d(i,j)]\quad i\neq j \tag{4}$$

式中，n 为样本总数，当样本 $i\in c$ 类时，$x(i)$ 表示样本 i 和同属 c 类的其它样本之间的平均距离。当样本 $i\notin c$ 类时，$y(i)$ 表示样本 i 和非 c 类的各个类中所有样本平均距离的最小值。个体轮廓系数 O_i 表示类内距离和类间距离，用以评价某个样本被划分到某一类别的合理性。若其取值范围为[－1，1]，越接近 1，则表示该样本的类内平均距离远小于最小的类间平均距离，该样本集的聚类效果越好。

在确定优秀样本比例时，受帕累托法则的启发，将优秀样本的比例设置为 20%，将与类中心距离位于前 20%的样本筛选为优秀样本，这些优秀样本所占比例小，但其对聚类结果产生的影响则很大。

基于轮廓系数的自适应 K-means 算法描述如下：

(1)设定参数 H，初始化 $j=1$，当 $j\leqslant H$ 时，完成如下循环：

①调用传统 K-means 算法，完成初始聚类；

②保留 K 个聚类中心，计算同属第 c 类样本点与该聚类中心点的距离矩阵 $Dist(j)_c(c=1,2,\cdots,k)$，对所有行进行排序；

③运用式(2)－式(4)分别计算 K 个聚类中心的个体轮廓系统，并相加，得到 $Sil(j)$。

(2)对 $Sil(j)$ 进行降序排列，标记为 $Sil(j)^h$，并将其对应的 $Dist(j)_c$ 标记为 $Dist(j)_c^h$，其中 $h=1,2,\cdots,H,c=1,2,\cdots,k$。

(3)取 $Dist(j)_c^1$ 的前 20%位所对应的样本作为候选优秀样本。

(4)以此类推，依次设定 $h=2,\cdots,H$，对 $Dist(j)_c^2$、$Dist(j)_c^3\cdots,Dist(j)_c^H$ 矩阵的每行，判断所对应的样本是否在(3)所确定的候选优秀样本群中。若在，则将该样本标志位设定为 1；若某样本被首次发现，则将其添加到候选优秀样本群中；若连续两个样本均不在候选优秀样本群中，则停止对该距离矩阵的样本选取，直到完成所有距离矩阵的选取。

(5)由上述方法得到优秀样本群，并对各类的优秀样本求均值，将其作为初始聚类中心，再完成聚类操作。

2.3 改进聚类算法的仿真实验

为验证改进算法的稳定性和有效性，本文选取了 7 个 UCI 数据集：Iris，Wine，New-thyroid，Diabetes，segment，waveform 以及 shuttle 进行实验，分别在聚类时间、聚类轮廓系数及聚类正确率等方面与文献[11-13]中的改进算法进行比较。在下述图中，将文献[11]的改进算法简称为 sk-means 算法，将文献[12]的改进算法简称为 DESk-means 算法，将文献[13]的改进算法简称为 MAXk-means 算法。

图 1 示出了当数据集规模大于 1000 时各个改进算法与传统 K-means 算法在完成不同数据集(Dlabetes、Segment. waveform、Shuttle)聚类时所需的时间，横坐标为 4 种待比较算法，纵坐标为时间差，每条图线上的点表示不同算法与传统 K-means 算法的时间差异。

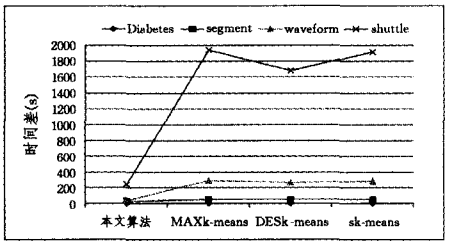

图 1 各种改进算法与 K-means 算法完成数据集聚类时所需时间差

图 1 是完成数据集聚类时，运用各种改进算法与传统 k-means 算法的时间差的直观体现。可以看出，当数据集规模大于 1000 时，如 segment、waveform，尤其是 shuttle，时间差异就十分明显，本文所提出的改进算法聚类时间明显少于 sk-means、DESk-means、MAXk-means 算法的聚类时间。

图 2 示出当聚类样本集的规模不断增大时各聚类算法完成聚类操作的时间。由图 2 可以看出，本文算法和传统 K-means 算法的聚类时间随样本数量的增加，其增长趋势比 sk-means、DESk-means、MAXk-means 算法的缓和，即本文算法比上述 3 种改进算法具有更快的收敛速度。

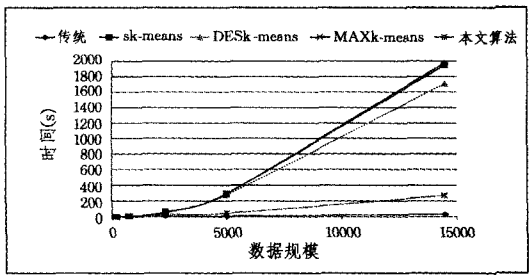

图 2　各算法聚类时间比较

图 3 是对各聚类算法的轮廓系数的比较，轮廓系数越大，则表明运用该算法的聚类效果越好。由图中可以看出，本文提出的算法得到的轮廓系数在每个数据集上的实验结果都是最高的，当然也比传统 K-means 算法聚类效果好。

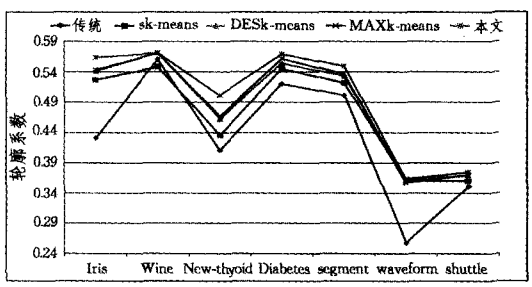

图 3　各算法聚类轮廓系数比较

图 4 是对各聚类算法的正确率的比较，由图 4 可以看出，改进算法(文献[11-13])的准确率优于传统 K-means 算法，并且本文改进的算法效果最好。

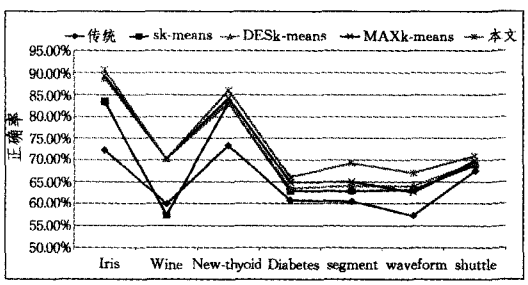

图 4　各算法聚类正确率比较

为了进一步说明改进算法的优越性，将新算法与基于密度的改进聚类算法 OPTICS[14]、基于多蚁型的蚁群聚类算法 MHAAC[15] 和改进的层次聚类算法 IG-CURE[16] 在 UCI 数据集上的聚类正确率进行了比较。

如表 1 所列，本文算法在 Wine 数据集上的准确率和 MHAAC、IG-CURE 算法基本相同，在其它数据集上的聚类结果均优于其他算法，准确率有较大提高。

表 1　各聚类算法准确率比较

数据集	OPTICS	MHAAC	IG-CURE	本文算法
Iris	81.33%	87.57%	88.93%	90.16%
Wine	56.30%	71.12%	71.62%	71.77%
New-thyroid	84.41%	82.90%	85.09%	86.13%
Diabetes	60.69%	62.14%	63.71%	65.92%
segment	63.16%	63.47%	66.86%	68.86%
waveform	62.13%	62.91%	63.57%	66.82%
shuttle	67.09%	68.31%	68.91%	71.61%

3　基于改进自适应 K-means 的 RBF 神经网络分类器

RBF 神经网络分类器的学习分为 3 个部分，分别为径向基函数中心 C_p 的学习、径向基函数宽度 σ_p 的学习、隐层与输出层之间联结权值 w_p 的学习。

(1)确定基函数中心 C_p

基函数中心的选取是至关重要的，是根据所有的输入样本决定隐藏层各节点的径向基函数的中心值 C_p。本文采用上述改进的自适应聚类算法求出输入样本的各类中心，并将其作为径向基函数的中心，提高了 RBF 网络的学习收敛速度和自适应性。

(2)径向基函数的宽度 σ_p

基函数宽度的选取往往根据聚类的结果来确定，令其等于聚类中心与训练样本之间的平均距离。$\sigma_i(i=1,2,\cdots,I)$ 表示 I 个基函数的方差，径向基函数设定为高斯函数，如式(5)所示：

$$\varphi(\| p_k-t_i \|)=\exp\{-\frac{1}{2\sigma_i^2}\| p_k-t_i \|^2\} \quad i=1,2,\cdots,I,\ k=1,2,\cdots,N \tag{5}$$

则其宽度可由式(6)得：

$$\sigma_1=\sigma_2=\cdots=\sigma_I=\frac{d_{\max}}{\sqrt{2I}} \tag{6}$$

式中，$d_{\max}$ 表示选择的中心两两之间的距离最大值。

(3)联结权值 w_p

在隐藏层节点数、基函数、基函数中心确定后，输出层权值可由线性方程组确定。

以 (x_i,d_i) 为例，计算网络的输出为：

$$y(p_i)=w_1\varphi_1(\| p_i-t_1 \|)+\cdots+w_{m1}\varphi_{m1}(\| p_i-t_{m1} \|) \tag{7}$$

对于每一个样本，都存在一个期望输出 d：

$$d_i=w_1\varphi_1(\| p_i-t_1 \|)+\cdots+w_{m1}\varphi_{m1}(\| p_i-t_{m1} \|) \tag{8}$$

亦或

$$d_i=[\varphi_1(\| p_i-t_1 \|)\cdots\varphi_{m1}(\| p_i-t_{m1} \|)][w_1\cdots w_{m1}]^{\mathrm{T}}$$

则对于所有的样本，有

$$[d_1\cdots d_N]^{\mathrm{T}}=\begin{bmatrix}\varphi_1(\| p_i-t_1 \|)\cdots\varphi_{m1}(\| p_i-t_{m1} \|)\\ \cdots\\ \varphi_1(\| p_N-t_1 \|)\cdots\varphi_{m1}(\| p_N-t_{m1} \|)\end{bmatrix}\cdot[w_1\cdots w_{m1}]^{\mathrm{T}} \tag{9}$$

令 $\Phi=\begin{bmatrix}\varphi_1(\| p_i-t_1 \|)\cdots\varphi_{m1}(\| p_i-t_{m1} \|)\\ \cdots\\ \varphi_1(\| p_N-t_1 \|)\cdots\varphi_{m1}(\| p_N-t_{m1} \|)\end{bmatrix}$

可得 $\begin{bmatrix}d_1\\ \cdots\\ d_N\end{bmatrix}=\Phi\begin{bmatrix}w_1\\ \cdots\\ w_{m1}\end{bmatrix}$

若用 Φ^+ 表示 Φ 的伪逆矩阵，可由式(10)计算得到权值 w：

$$[w_1 \cdots w_{m1}]^T = \Phi^+ [d_1 \cdots d_N]^T \quad (10)$$

基于改进自适应 K-means 算法的 RBF 神经网络分类器的设计流程如图 5 所示。

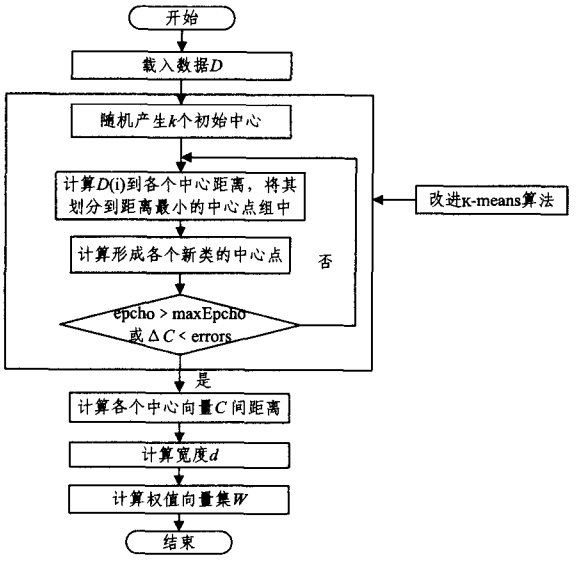

图 5 基于改进自适应 k-means 算法的 RBF 神经网络分类器设计流程图

4 仿真实验

为验证改进的自适应聚类算法的准确性和高效性，本文以 Matlab7 作为运行环境，对多维 UCI 数据进行实验仿真。针对 UCI 的多维数据集，随机选择训练样本完成对基于改进自适应聚类算法的 RBF 网络分类器的训练；再对测试数据进行分类，最终通过比较分类的错误率来衡量传统 RBF 网络分类器和改进的 RBF 网络分类器的性能优劣。

实验中随机选取数据集规模的 80%作为训练集，其余作为测试集，输出参数包括：对训练样本的回判错误率和对测试样本判断的错误率；为提高实验可信度，以 10 次算法调用为一个处理单位，求其平均值，共处理 15 次，求取两种算法的训练样本回判错误率和测试样本判断错误率之间的差异。

实验中选取了 Iris 数据集和 segment 数据集。

Iris 数据集共 150 个样本，训练集含 120 个样本，测试集含 30 个样本，实验结果如图 6 所示。

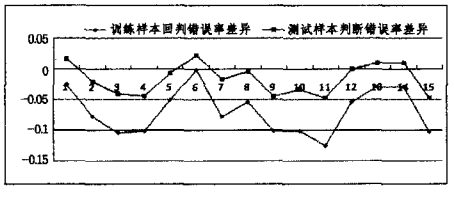

图 6 Iris 数据集错误率差异折线图

直观地比较两种算法的处理结果，当被减数为新算法的相关错误率，即差异为负值时，说明新算法的错误率低于传统算法。通过图 6 中折线可以看出，15 次实验中训练样本回判错误率差异都在横坐标轴下方，测试样本判断错误率差异除了第 1、6、13、14 次在横坐标上方，其余均在下方。

segment 数据集共 2310 个样本，训练集含 1848 个样本，测试集含 462 个样本，实验结果如图 7 所示。

通过对数据和折线图的观察，可看出训练样本回判错误率差异都为负值，在横坐标轴下方，而测试样本判断错误率差异除了第 2、3、8 次为正值，其余都为负值。

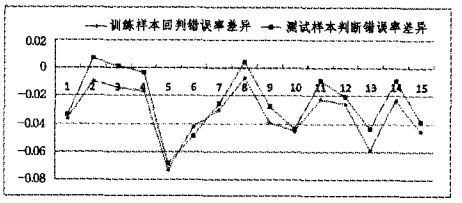

图 7 segment 数据集错误率差异折线图

可以看出，基于改进的自适应聚类算法的 RBF 神经网络分类器较传统 RBF 分类器，具有更高的分类精度。

结束语 为克服传统 K-means 算法受初始聚类中心影响的缺陷，本文首先提出了一种改进的自适应聚类算法。该算法引入轮廓系数，在样本集中筛选出候选优秀样本群，使得初始聚类中心在空间分布上尽量符合实际数据的分布，其次，运用该算法快速有效地确定隐含层节点的径向基函数中心 C_p 及函数的宽度，完成了 RBF 神经网络分类器的构造。最后，通过对改进的聚类算法及基于改进聚类算法的 RBF 分类器的大量仿真实验，充分说明前者具有收敛速度快、准确率高的特点，后者具有分类精度高、训练速度快的特点。

参 考 文 献

[1] 阮晓钢. 神经计算科学[M]. 北京：国防工业出版社，2006

[2] 杨戈，吕剑虹，刘志远，等. 一种新型 RBF 网络序贯学习算法[J]. 中国科学 E 辑，2004，34(7)：763-775

[3] 倪友平，姜卫东，陈曾平. 一种优化 RBF 神经网络训练算法及其在目标识别中的应用[J]. 测控技术，2005，194(3)：18-20

[4] 穆云峰. RBF 神经网络学习算法在模式分类中的应用研究[D]. 大连：大连理工大学，2006

[5] 郭伟. 基于互信息的 RBF 神经网络结构优化设计[J]. 计算机科学，2013，40(6)：252-255

[6] 张友民，李庆国，戴冠中，等. 一种 RBF 网络结构优化方法[J]. 控制与决策，1996，11(6)：667-671

[7] 高国平. 基于径向基函数神经网络的企业信用评分研究[D]. 沈阳：东北大学，2007

[8] 阎平凡，张长水. 人工神经网络与模拟进化计算[M]. 北京：清华大学出版社，2000

[9] 武方方，赵银亮. 一种基于蚁群聚类的径向基神经网[J]. 西安交通大学学报，2006，40(4)：385-388

[10] 岳彩表，常青美，庞学民，等. 基于聚类分析的 RBF 网络建模方法及应用的研究[J]. 计算机仿真，2006，23(1)：120-123

[11] 傅德胜，周辰. 基于密度的改进 K 均值算法及实现[J]. 计算机应用，2011，31(2)：432-434

[12] 谢娟英，郭文娟，谢维信，等. 基于样本空间分布密度的初始聚类中心优化 K-均值算法[J]. 计算机应用研究，2012，29(3)：888-892

[13] 陈光平，王文鹏，黄俊. 一种改进初始聚类中心选择的 K-means 算法[J]. 小型微型计算机系统，2012，33(6)：1320-1323

[14] 段明秀，唐超琳. 一种基于密度的聚类算法实现[J]. 吉首大学学报，2013，34(1)：26-27

[15] 李聪，封化民. 基于多蚁型的蚁群聚类算法[J]. 北京电子科技学院学报，2012，20(4)：6-12

[16] 刘一鸣，张化祥. 引入信息增益的层次聚类算法[J]. 计算机工程与应用，2012，48(1)：142-144