第35卷 第20期 计 算 机 工 程 2009年10月
Vol.35 No.20 Computer Engineering October 2009
·人工智能及识别技术· 文章编号：1000—3428(2009)20—0189—03 文献标识码：A 中图分类号：TP301.6

一种改进的 Chameleon 算法

龙真真[1,2]，张 策[2]，刘飞裔[1]，张正文[3]
(1. 国防科技大学信息系统与管理学院，长沙 410073；2. 空军装备研究院，北京 100085；
3. 中国科学院数学与系统科学研究院系统科学研究所，北京 100080)

摘 要：利用 Chameleon 算法进行 K 值选择、相似度函数阈值选择时需要人为给出一些参数，在没有先验知识的情况下，人为确定此类参数难度较大。针对该问题介绍模块度概念，根据结构等价相似度和模块度概念提出一种聚类算法——M-Chameleon。实验结果证明，M-Chameleon 可以客观地反映实际聚类情况。

关键词：聚类算法；Chameleon 算法；结构等价相似度；模块度

Improved Chameleon Algorithm

LONG Zhen-zhen[1,2], ZHANG Ce[2], LIU Fei-yi[1], ZHANG Zheng-wen[3]
(1. School of Information System and Management, National University of Defense Technology, Changsha 410073;
2. Equipment Academy of Air Force, Beijing 100085;
3. Institute of System Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100080)

【Abstract】 It is found that some parameters should be determined by hand when using Chameleon algorithm to choose K value and the threshold value of similarity degree function. It is difficult to determine these parameters without any prior domain knowledge. Aiming at this problem, this paper introduces modularity theory and proposes a clustering algorithm——M-Chameleon according to the structural equivalence similarity degree and modularity theory. Experimental results confirm that M-Chameleon can reflect the actual clustering situation objectively.

【Key words】 clustering algorithm; Chameleon algorithm; structural equivalence similarity degree; modularity

1 概述

随着科学技术的高速发展以及各种资源数量的不断增加，信息处理已成为一个研究热点，它涉及信息抽取、自然语言理解、自动聚类和分类、自动摘要、自动标注和主题识别、信息结构分析以及文本生成。其中，关于自动聚类的研究较深入，聚类技术已成为信息处理的核心技术。从 20 世纪 40 年代至今，研究者提出了很多聚类算法，主要分为基于层次的算法、基于平面分割的算法、基于密度的算法、基于规则和模型的算法以及基于网格和子空间的算法。

Chameleon[1]作为一种目前较好的层次聚类算法，具有发现任意形状簇的能力，但其主要缺点如下：(1)K-最近邻图中 K 值的确定需要人工进行；(2)最小二等分的选取困难；(3)相似度函数的阈值需要人工给定。上述缺点影响了聚类的无监督性，鉴于此，本文提出 M-Chameleon 算法。

2 Chameleon 算法

Chameleon 是一个在层次聚类中采用动态模型的通用聚类算法。Chameleon 算法进行聚类分析的过程如图 1 所示，该算法基于常用 K-最邻近图方法描述它的对象。K-最邻近图中的每个节点代表一个对象，如果一个对象是另一个对象的 K 个最类似对象之一，那么在这 2 个对象之间存在一条边，边的权重用 2 个对象间的相似度表示。做上述处理的优点在于距离很远的对象完全不相连，边的权重代表了潜在空间密度信息。在 Chameleon 算法中，2 个簇的相对互连性被定义为 2 个簇之间的绝对互连性关于 2 个簇间的内部互连性的规范化。相对近似性定义为 2 个簇之间绝对近似度关于 2 个簇间的内部近似度的规范化。相似度函数定义为 2 个簇的相对互联性和相对近似性的乘积。合并规则为优先合并相似度相近的簇。

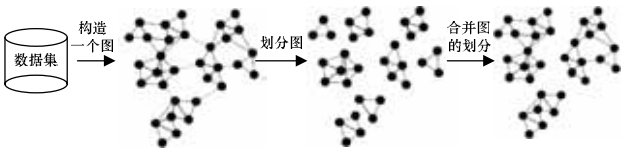


图 1 Chameleon 算法的主要过程

3 M-Chameleon 算法

3.1 加权图 G 的构建

以数据对象为顶点，对象间的距离为基础将数据对象集转化为一个加权图，与 Chameleon 算法相异的是，此时不使用 K-最邻近的概念，而是用权代表 2 个对象间的相似程度，避免了对 K 值的人工估计。

定义 1(度量空间) 一个度量空间是指一个数据对象集 X 和此数据对象集合上的距离函数 H 构成的二元组(X, H)，其中，距离函数 H 满足以下条件：

(1)非负性，有 $0 < H(x,y) < \infty$，$x \neq y$，$H(x,x) = 0$，对于任意 x,y 属于 X 都成立；

(2)对称性，$H(x,y) = H(y,x)$，对于任意 x,y 属于 X 都成立；

(3)三角不等式，$H(x,y) \le H(x,z) + H(z,y)$，对于任意 x,y,z 属于 X 都成立。

作者简介：龙真真(1983－)，男，硕士研究生，主研方向：系统论证与仿真评估；张 策，高级工程师、硕士；刘飞裔、张正文，硕士研究生

收稿日期：2009-05-07 **E-mail：**longzhenzhen@gmail.com

本文中距离函数 H 定义为欧氏距离，其计算公式为

$$H(i,j)=\sqrt{(|z_{i1}-z_{j1}|^2+|z_{i2}-z_{j2}|^2+\cdots+|z_{ip}-z_{jp}|^2)}$$

其中，$i=(x_{i1},x_{i2},\cdots,x_{ip})$；$j=(x_{j1},x_{j2},\cdots,x_{jp})$；$z_{if}=\dfrac{x_{if}-m_f}{S_f}$；

$S_f=\sqrt{\dfrac{n\sum_{i=1}^{n}x_{if}^2-(\sum_{i=1}^{n}x_{if})^2}{n(n-1)}}$；$m_f=(x_{1f}+x_{2f}+\cdots+x_{nf})/n$。

定义 2(加权图) 加权图 $G=(V,E,W)$ 是由有穷非空节点集 V、边集 E 和边权值函数 W 构成的三元组。无序偶对 (v_i,v_j) 表示节点 $v_i\in V$ 与 $v_j\in V$ 之间的联系，w_{ij} 为边 (v_i,v_j) 的权值。设(X,H)是一个度量空间，设常数 $C\sim O(\min H(i,j)),v_i,v_j\in V$，连接 2 个节点的边的权值函数为 $W_{ij}=\dfrac{1}{C+H(i,j)}$。

3.2 模块度

Chameleon 算法的实质是寻找内部连接紧密而外部连接稀疏的簇，并对其进行合理分割。但由于在该算法中包含最小二分概念，因此其实际操作困难。为解决此问题，本文引入模块度[2]概念。

定义 3(带权度) 节点 i 的带权度 k_i 定义为与该节点连接的其他节点的边的权重值之和，设 A ($n\times n$)为图 G 的邻接矩阵，则 $k_i=\sum_{j=1}^{n}A_{ij}$。

定义 4(模块度) 设某聚类结果将图 $G=(V,E)$ 划分为 k 个簇，即 $G=(G'_1,G'_2,\cdots,G'_k)$，$G'_i=(V'_i,E'_i)$。设 c_i 为节点 i 所在的簇标号，$\delta(c_i,c_j)=\begin{cases}1,\text{如果}c_i=c_j\\0,\text{其他}\end{cases}$。定义一个 $k\times k$ 维的对称矩阵 $\boldsymbol{E}=(\xi_{ij})$，$\xi_{ij}=\dfrac{\sum_{v_l\in V'_i,v_k\in V'_j,(v_l,v_k)\in\boldsymbol{E}}\delta(c_l,c_k)}{\sum_{v_l,v_k\in V}A_{lk}}$。矩阵 $\boldsymbol{E}$ 的迹 $Tre=\sum_{i=1}^{k}\xi_{ii}$，每行(列)中各个元素之和为 $a_i=\sum_{j=1}^{k}\xi_{ij}$。用下式定义模块度的衡量标准：

$$Q=\sum_{i=1}^{k}(\xi_{ii}-a_i^2)=Tre-\|\xi^2\|$$

其中，$\|.\|$ 表示矩阵所有元素之和。上式的物理意义如下：图中连接 2 个同类型的节点的边(即簇内部的边)的比例，减去在同样的图下任意连接这 2 个节点的边的比例的期望值。如果簇内部边的比例不大于任意连接时的期望值，则 $Q=0$。Q 的上限为 $Q=1$，Q 越接近该值，聚类结果越符合簇内连接紧密簇外连接稀疏的标准，即聚类结果越好。

定义 4 中的模块度是针对非加权网的，但 Newman 证明[3]，一个加权网可以视为数个非加权网的叠加。因此，假定聚类结果为 $G=(G'_1,G'_2,\cdots,G'_k)$，加权图 $G=(V,E,W)$ 的模块度可以定义为

$$Q=\frac{1}{2m}\sum_{v_i,v_j\in V}(W_{ij}-\frac{k_ik_j}{2m})\delta(c_i,c_j)$$

其中，$k_i=\sum_{j=1}^{n}A_{ij}$，即节点 i 的带权度；$m=\dfrac{1}{2}\sum_{v_i,v_j\in V}W_{ij}$ 是图中边的权值的和；c_i 为节点 i 所在簇标号；$\delta(c_i,c_j)=\begin{cases}1,\text{如果}c_i=c_j\\0,\text{其他}\end{cases}$。

3.3 算法流程

将本算法命名为 M-Chameleon，M 代表模块度。

在 Chameleon 算法中，相似度函数的作用是从拓扑结构的角度来度量 2 个簇之间的类似程度，但由于其需要人工指定阈值，造成使用上的困难，而不指定阈值，又将导致无法找到算法终结点。而本文算法由于应用了模块度概念，因此可以避免上述问题。本文采用结构等价相似度定义 2 个簇之间的类似程度。

定义 5(结构等价相似度[4]) 如果 2 个节点间具有完全相同的相邻节点，则称这 2 个节点为结构等价。因此，结构等价相似度计算式为

$$S_{i,j}=\sqrt{\sum_{k\neq i,j}(W_{ik}-W_{jk})^2}$$

其中，W_{ij} 是边 (v_i,v_j) 的权值。对于结构等价的节点对，结构等价相似度为 0。对于没有任何共同相邻节点的节点对，结构等价相似度取最大值。

定义 6 簇 r 和簇 h 间的结构等价相似度定义为

$$S_{r,h}=\frac{1}{n_rn_h}\sum_{i=1}^{n_r}\sum_{j=1}^{n_h}S_{x_{ri},hj}$$

其中，n_r 为簇 r 中节点个数；n_h 为簇 h 中的节点个数。

具体计算步骤如下：假设初始加权图分为 n 个独立的簇，每次选取结构等价相似度小的 2 个簇进行合并，并进行模块度的评估，直到模块度开始递减，从而得到上述聚类过程的一个树状图。最下面的聚集结构就是初始的 n 个节点，树状图的每个分支处的划分对应一种聚类结果，最终划分结果由模块度最大值决定。具体过程如下：

(1)采用定义 2 的结构，构建加权图 G，初始化簇的个数为 n，即每个节点就是一个簇；

(2)计算 $S_{i,j}=\sqrt{\sum_{k\neq i,j}(W_{ik}-W_{jk})^2}$ 或 $S_{r,h}=\dfrac{1}{n_rn_h}\sum_{i=1}^{n_r}\sum_{j=1}^{n_h}S_{x_{ri},hj}$，按结构等价相似度由小到大的顺序，合并簇；

(3)计算模块度 Q；

(4)重复进行(2)和(3)，不断合并簇，直到模块度 Q 下降，终止，输出结果。

该算法的基本思想和 Chameleon 算法一致，即将数据对象转化为加权图，并寻找一个最优划分，使簇内部连接紧密，外部连接稀疏。M-Chameleon 与 Chameleon 具有如下差异：(1)没有采用 K-最邻近图的概念，因此，不存在 K 值的人工选取问题；(2)没有采用最小二分概念，避免了操作的困难；(3)沿结构相似度从小到大的顺序进行合并，并由模块度决定算法的终止点，避免了相似度函数阈值的人工选取问题。

4 实验结果

本文采用文献[5]提出的聚类纯度检验聚类效果。其定义如下：

定义 7(聚类纯度) 对于数据集 D，令 P 表示 D 上一个聚类结果(即一个划分)，$p\in P$ 表示该划分中的一个簇。令 L 表示 D 上的一个手工分类的结果(也是一个划分)，$l\in L$ 表示该手工分类的一个类别，则有：

对于 $\forall p\in P$，p 相对于 $l\in L$ 的查准率定义为

$$Precision(p,l)=\frac{|p\cap l|}{|p|}$$

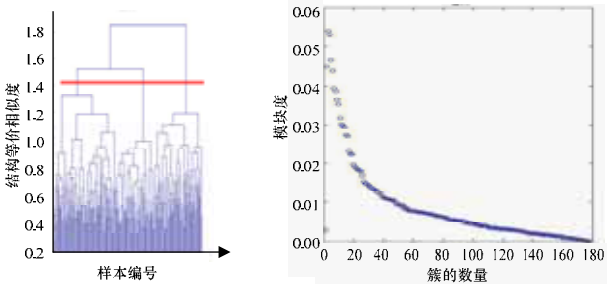
聚类纯度定义为 $Purity(P,L)=\sum_{p\in P}\dfrac{|P|}{|D|}\max_{l\in L}Precision(p,l)$。

由定义 7 可以看出，聚类纯度是每个簇的查准率的加权平均值，而各簇的查准率取决于它相对各手工分类类别的最大查准率。该指标能较全面地反映聚类效果。

本文实验数据集 wine[6]选取自 UCL 的机器学习数据集，

其来源是意大利某食品检验中心对 3 个不同产地的酿酒分析数据，该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。

作为对比，本文选取 *K*-means[7]和 *K*-means-CP[8]算法作为对比。M-Chameleon 算法聚类结果如图 2 所示，可以看出，当 Q 有最大值 0.054 时，对应最终聚类结果为 3 个簇。由图 3 可知，M- Chameleon 算法的聚类结果具有最好的聚类纯度，达 96.84%。



(a)划分图　　(b)Q 值图

图 2　M-Chameleon 算法聚类结果

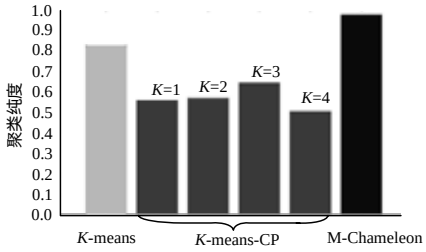


图 3　实验结果对比

5　结束语

使用 Chameleon 算法时，没有掌握数据集背景知识的人员无法精确确定一些参数，限制了该算法的适用范围。因此，本文提出基于模块度的 Chameleon 改进算法 M-Chameleon，并通过实验证明其可行性和先进性。下一步工作的重点是通过改进该算法存储结构提高算法效率。

参考文献

[1] Karypis G, Han Eui-Hong, Kumar V. Chameleon: Hierarchical Clustering Using Dynamic Modeling[J]. Computer, 1999, 32(8): 68-75.

[2] Newman M E J, Girvan M. Finding and Evaluating Community Structure in Networks[J]. Phys. Rev. E, 2004, 69(2): 113-127.

[3] Newman M E J. Analysis of Weighted Networks[J]. Phys. Rev. E, 2004, 70(5): 1-9.

[4] Burt R S. Positions in Networks[J]. Social Forces, 1976, 5(1): 93-122.

[5] Sun Ying, Zhu Qiuming, Chen Zhengxin. An Iterative Initial-points Refinement Algorithm for Categorical Data Clustering[J]. Pattern Recognition Letters, 2002, 23(7): 875-884.

[6] Forina M. PARVUS — — An Extendible Package for Data Exploration, Classification and Correlation[R]. Genoa, Italy: Institute of Pharmaceutical and Food Analysis and Technologies, Tech. Rep.: 16147, 1988.

[7] Marques J P. Pattern Recognition Concepts, Methods and Applications[M]. 2nd ed. Beijing, China: Tsinghua University Press, 2002.

[8] Ding C H Q, He Xiaofeng. K-nearest-neighbor in Data Clustering: Incorporating Local Information into Global Optimization[C]//Proc. of the ACM Symp. on Applied Computing. Nicosia, Cyprus: ACM Press, 2004: 584-589.

编辑　陈　晖

(上接第 188 页)

但是检测性能有一定程度的降低，主要原因在于本算法所求出的分类器是 Adaboost 算法的相对最优结果，它以小的性能牺牲换取了较快的训练速度。

表 1　不同训练方法训练的人脸检测器检测结果对比

方法	检测率/(%)	误检率/(%)	训练时间
改进的 Adaboost 人脸检测器	87.9	15.6	10 h
OpenCV 中的人脸检测器	92.3	12.2	超过 7 天

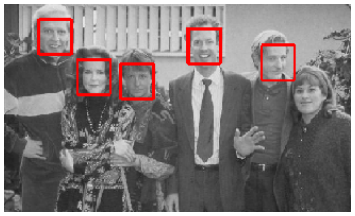


图 3　部分检测结果图

5　结束语

本文对传统 Adaboost 训练算法进行了改进，避免了 Adaboost 算法中弱分类器频繁的训练过程，最大程度地提高了分类器的训练速度，使 Adaboost 分类器可以灵活应用于更多的领域。

参考文献

[1] Sung Kahkay, Poggio T. Example-based Learning for View-based Human Face Detection[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1998, 20(1): 39-51.

[2] Yang Ming-Hsuan, Kriegman D, Ahujua N. Detecting Faces in Images: A Survey[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2002, 24(1): 34-58.

[3] Freund Y, Schapire R. Experiments with a New Boosting Algorithm[C]//Proc. of the 13th Conference on Machine Learning. San Francisco, USA: Morgan Kaufmann, 1996.

[4] Viola P, Jones M. Rapid Object Detection Using a Boosted Cascade of Simple Features[C]//Proc. of IEEE Conf. on Computer Vision and Pattern Recognition. Hawaii, USA: IEEE Press, 2001.

[5] Rafael C, Richard E, Steven L. Digital Image Processing Using Matlab[M]. [S. l.]: Prentice Hall, 2005.

[6] 刘瑞祯, 于仕琪. OpenCV 教程——基础篇[M]. 北京: 北京航空航天大学出版社, 2007.

编辑　张　帆