

一种改进 CHAMELEON 算法的聚类算法 COCK

朱烨行¹, 李艳玲², 杨献文³

(1 西安邮电大学 经济与管理学院, 陕西 西安 710121;

2 第二炮兵工程大学, 陕西 西安 710025;

3 西安财经学院 信息与教育技术中心, 陕西 西安 710061)

摘 要: 通过对现有的 CHAMELEON 算法进行改进, 并借鉴 ROCK 算法的计算步骤, 提出了一个新的层次聚类算法 COCK. 改进之处在于: 合并的簇的内部紧密性、合并的簇的内部互连性、相对紧密性和相对互连性的计算方法进行了改变, 并取消了 CHAMELEON 算法原有两个阶段的第一个阶段. 由簇 U 和簇 V 合并构成的簇 W 的内部紧密性由两个因素决定, 一个是簇 U 和簇 V 本身的内部紧密性的加权和, 另一个是簇 U 和簇 V 之间的绝对紧密性; 簇 W 的内部互连性由两个因素决定, 一个是簇 U 和簇 V 本身的内部互连性的加权和, 另一个是簇 U 和簇 V 之间的绝对互连性; 由簇 W 和簇 J 合起来的簇的相对紧密性的计算方法是, 先求出簇 W 和簇 J 内部紧密性的加权和, 用这个加权和去除簇 W 和簇 J 之间的绝对紧密性; 计算两个簇 W 和 J 之间的相对互连性的方法是, 把簇 W 和簇 J 之间的绝对互连性除以簇 W 和簇 J 内部互连性的加权和.

关键词: 文本聚类; 文档聚类; CHAMELEON; ROCK; 算法

中图分类号: TP391.4

文献标识码: A

文章编号: 1000-7180(2015)12-0173-04

Clustering Algorithm COCK Improved from CHAMELEON Algorithm

ZHU Ye-hang¹, LI Yan-ling², YANG Xian-wen³

(1 School of Economics and Management, Xi'an University of Posts and Telecommunications, Xi'an 710061, China;

2 Department of Electronic Engineering, The Second Artillery Engineering University, Xi'an 710025, China;

3 Information and Educational Technology Center, Xi'an University of Finance and Economics, Xi'an 710061, China)

Abstract: By improving the existing algorithm CHAMELEON, and using the calculate steps of ROCK algorithm for reference, we propose a new hierarchical clustering algorithm COCK, the improvement comprise that we change the calculation method of the combined cluster internal closeness, the combined cluster internal interconnectivity, relative closeness, relative interconnectivity. And cancel the first phase of CHAMELEON original two-stage algorithm. Two factors determine internal closeness of Cluster W which is constituted of cluster U and V, one is the weighted sum of internal closeness of cluster U and V, another is absolute closeness between cluster U and V; internal interconnectivity of cluster W is determined by two factors, one is weighted sum of internal interconnectivity of cluster U and V, another is the absolute interconnectivity between cluster U and V. Calculation method of relative closeness of the cluster combined by cluster W and J is, at first calculate the weighted sum of internal closeness of cluster W and J, then divide the absolute closeness between cluster W and J by this weighted sum, calculate method of relative interconnectivity between cluster W and J is, divide the absolute interconnectivity between cluster W and J by the weighted sum of internal interconnectivity of cluster W and J.

Key words: text clustering; document clustering; CHAMELEON; ROCK; algorithm

收稿日期: 2014-04-09; 修回日期: 2014-05-28

基金项目: 国家自然科学基金青年项目(71102149)

1 引言

CHAMELEON 聚类算法于 1999 年由 Karypis 等人提出^[1]. 它首先根据数据项之间的相似度建立 K 最近邻图, 然后进行聚类, 其具体过程分为两步: 第一步使用图分割算法把所有数据项分为许多小的簇;^[2] 第二步反复合并这些小的簇, 合并的依据是簇之间的“相对互联性(Relative Inter-Connectivity)”和“相对紧密性(Relative Closeness)”的大小^[3].

两个簇之间的绝对“互连(Absolute Inter-Connectivity)”被定义为连接两个簇的所有边的权重的和; 两个簇之间的绝对“紧密性(Absolute Closeness)”被定义为这些边的权重的平均值^[3].

使用图分割算法把一个簇分为大小差不多大的两部分, 连接这两部分的所有边的权重的和被定义为该簇的“内部互连”; 连接这两部分的所有边的权重的均值被定义为该簇的“内部紧密性”^[3].

两个簇之间的“相对互连(relative inter-connectivity)”被定义为两个簇之间的绝对互连除以两个簇的内部互连的加权和; 两个簇之间的“相对紧密性(relative closeness)”被定义为两个簇之间的绝对紧密性除以两个簇的内部紧密性的加权和^[4].

2 COCK 聚类算法原理

我们把本文提出的算法命名为 COCK, 其命名源自 CHAMELEON 算法名字的头一个字母和 ROCK 算法名字去掉头一个字母的其余部分, 意味着 COCK 算法是揉合 CHAMELEON 算法和 ROCK 算法形成的一个聚类算法.

我们提出的 COCK 算法只有不断地合并较小的簇这一个步骤.

我们首先定义只含有一个点的簇的内部紧密性值和内部互连性值都为值 1.^[4]

2.1 合并的簇的内部紧密性

假设簇 U 有 N_U 个点, 簇 V 有 N_V 个点, 已知簇 U 和簇 V 的内部紧密性(Internal Closeness) 分别为 ICL_U 和 ICL_V , 若簇 U 和 V 合并为簇 W , 如图 1 所

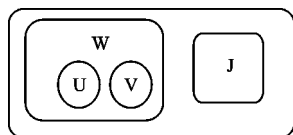


图 1 COCK 算法聚类过程示意图

示, 我们用式(1) 来计算合并后的簇 W 的内部紧密性 ICL_W , 这里 α 是常数 0.87, 经过不断尝试, 我们发现当聚类文档时, α 取该值效果最好^[3].

式(1) 中的 $ACL_{(U,V)}$ 表示簇 U 和簇 V 之间的绝对紧密性(Absolute Closeness), 被定义为连接簇 U 中点和簇 V 中点的所有边的权值的平均值^[3].

$$ICL_W = \left(\frac{N_U}{N_U + N_V} * ICL_U + \frac{N_V}{N_U + N_V} * ICL_V \right) * \alpha + ACL_{(U,V)} * (1 - \alpha) \quad (1)$$

可见由簇 U 和簇 V 合并构成的簇 W 的内部紧密性由两个因素决定: 一个是簇 U 和簇 V 本身的内部紧密性的加权和; 另一个是簇 U 和簇 V 之间的绝对紧密性(Absolute Closeness) $ACL_{(U,V)}$. 由参数 α 决定这两部分在合并簇 W 的内部紧密性(Internal Closeness) 值 ICL_W 中所占的比例^[3].

2.2 合并的簇的内部互连性

已知簇 U 和簇 V 的内部互连性(Internal inter-Connectivity) 分别为 ICO_U 和 ICO_V , 一个系数按式(2) 计算^[3]:

$$\epsilon = \frac{|N_U - N_V|}{N_U + N_V} \quad (2)$$

计算由簇 U 和簇 V 合并构成的簇 W 的内部互连性的公式为式(3), 式(3) 中的 $ACO_{(U,V)}$ 表示簇 U 和 V 之间的绝对互连(Absolute inter-Connectivity), 它仍被定义为连接簇 U 中点和簇 V 中点的所有边的权值的和^[3].

$$ICO_W = \left(\frac{N_U}{N_U + N_V} * ICO_U + \frac{N_V}{N_U + N_V} * ICO_V \right) * \epsilon + ACO_{(U,V)} * (1 - \epsilon) \quad (3)$$

当簇 U 和簇 V 所含点数相同时, 系数 ϵ 值为零, 合并的簇 W 的内部互连完全由簇 U 和簇 V 之间的绝对互连 $ACO_{(U,V)}$ 决定; 簇 U 和簇 V 所含点数相差越大, 系数 ϵ 值越大, 簇 U 和簇 V 之间的绝对互连 $ACO_{(U,V)}$ 在合并簇 W 的内部互连 ICO_W 中所占比重越小^[3].

2.3 相对紧密性

文献^[3] 用式(4)^[3] 计算簇 W 和簇 J 合起来的簇的相对紧密性(Relative Closeness): 先求出簇 W 和簇 J 内部紧密性(Internal Closeness) 的加权和, 用这个加权和去除簇 W 和簇 J 之间的绝对紧密性(Absolute Closeness) $ACL_{(W,J)}$.^[3]

$$tRCL = \frac{ACL_{(W,J)}}{\frac{N_W}{N_W + N_J} * ICL_W + \frac{N_J}{N_W + N_J} * ICL_J} \quad (4)$$

但在试验中发现,这样最终聚类结果形成的簇中往往有一个特别大,含有绝大多数点,而其他的簇一般只含有几个点.文献[5]中提到过这种现象,他们的解决办法是把结果除以两个簇 W 和簇 J 合起来的簇的内部关联的期望值,这样可以阻止关联少的点被聚到同一个簇中,因为关联少的点“放到同一个簇中将引起关联的期望值的增加值超过实际的关联值的增加值,结果会使判断函数的值变小”^[5].文献[5]中给出大小为 N_w 的簇,其内部的关联的期望值的大小为 $(N_w) * (1 + 2 * f(\theta))$,其中 $f(\theta)$ 的计算公式为式(5).因而由簇 N_w 和簇 N_j 合起来的簇大小为 $N_w + N_j$,其内部关联的期望值^[5]为 $(N_w + N_j) * (1 + 2 * f(\theta))$.这里系数 θ 取值为 0.4,经不断尝试我们发现当聚类文档时取该值效果最好.据此我们利用式(5)、(6)来对式(4)得到的值进行修正,得到最终两个簇 W 和簇 J 合起来的簇的相对紧密性^[5].

$$f(\theta) = \frac{1 - \theta}{1 + \theta} \quad (5)$$

$$dRCL = \frac{tRCL}{(N_w + N_j)(1 + 2 * f(\theta))} \quad (6)$$

2.4 相对互连性

对于两个簇 W 和 J 之间的相对互连性(Relative inter-Connectivity),文献[4]中采用式(7)进行计算,把簇 W 和簇 J 之间的绝对互连性(Absolute inter-Connectivity) $ACL_{(W, J)}$ 除以簇 W 和簇 J 内部互连性(Internal inter-Connectivity)的加权和^[4].

$$tRCO = \frac{ACO_{(W, J)}}{\frac{N_w}{N_w + N_j} * ICO_w + \frac{N_j}{N_w + N_j} * ICO_j} \quad (7)$$

但在试验中发现,这样最终形成的簇中往往有一个特别大,含有绝大多数点,而其他的簇一般只含有几个点.而文献[5]中提到过这种现象,他们的解决办法是除以两个簇 W 和簇 J 之间关联的期望值,这样可以阻止关联少的点被聚到同一个簇中,因为关联少的点“放到同一个簇中将引起关联的期望数的增加值超过实际的关联数的增加值,结果会使判断函数的值变小”^[5].文献[5]中给出大小为 N_w 的簇,其内部的关联的期望值为 $(N_w) * (1 + 2 * f(\theta))$,大小为 N_j 的簇,其内部的关联的期望值为 $(N_j) * (1 + 2 * f(\theta))$,其中 $f(\theta)$ 的计算公式为式(5).而由簇 N_w 和簇 N_j 合起来的簇大小为 $(N_w + N_j)$,其内部关联的期望值为^[5]

$$(N_w + N_j) * (1 + 2 * f(\theta)).$$

合起来的簇 $(N_w + N_j)$ 的内部关联值减去簇 N_w 内部的关联值,再减去簇 N_j 内部的关联值,就剩下簇 W 和簇 J 之间的关联值^[5]:

$$(N_w + N_j)^{1+2*f(\theta)} - N_w^{1+2*f(\theta)} - N_j^{1+2*f(\theta)} \quad (8)$$

这里系数 θ 取值同前,为 0.4.据此我们利用式(5)和式(8)来对式(7)得到的值进行修正,得到最终两个簇 W 和簇 J 之间的相对互连性^[5].

$$dRCO = \frac{tRCO}{(N_w + N_j)^{1+2*f(\theta)} - N_w^{1+2*f(\theta)} - N_j^{1+2*f(\theta)}} \quad (9)$$

2.5 COCK 聚类算法的判断函数

最终,我们将式(6)、(9)、(10)做为判断函数(Criterion function)来决定哪一对簇目前最相似,应该合并.按式(10)计算取值最大的一对簇就被认为是最相似的簇,也就是目前应该合并的簇^[4].

$$\text{Similarity}(CW, CJ) = dRCO(CW, CJ) * dRCL(CW, CJ) \quad (10)$$

2.6 COCK 聚类算法步骤

我们的 COCK 算法属于层次凝聚聚类算法,聚类步骤如下所示^[5].

(1) 求出所有点的对称的 K 最近邻图.

(2) 按照式(10)计算当前每个簇两两之间的相似度值,合并当前关系最密切的两个簇,形成合成的簇,按照式(1)~(3)计算刚合成的簇的内部互连性值和内部紧密性值.

(3) 按照式(6)、(9)计算刚合成的簇与其他剩余簇间的相对互连性值和相对紧密性值.

(4) 如果当前的簇数已减小到指定的簇数,则终止.否则返回步骤(2).

我们提出的 COCK 聚类算法更详细的计算步骤可参考 ROCK 聚类算法^[5].

3 COCK 聚类算法理论分析

CHAMELEON 聚类算法复杂,使用图分割算法把所有数据项分为许多小的簇;再反复合并这些小的簇.而我们的 COCK 聚类算法只有一个步骤,即反复合并当前最相似的簇.显然运算时间上肯定比 CHAMELEON 快.

象 CHAMELEON 算法一样,我们的 COCK 聚类算法在聚类时不仅考虑了两个簇之间的互连性,也考虑了两个簇之间的紧密性.而层次凝聚聚类算法 UPGMA 就没有考虑这么多因素,因而我们的算法聚类效果应该好于 UPGMA 算法.

4 实验结果

我们的另一个算法——PCOCK 算法包括两个阶段:先分割,再凝聚.在凝聚的阶段使用了我们的 COCK 算法.假如在第一阶段完全相同的情况下,在第二阶段使用其他的凝聚聚类算法,产生的结果可以用来比较我们的 COCK 算法的聚类质量.

UPGMA 被认为是在文档聚类时最有效的层次凝聚聚类算法,在我们的 PCOCK 算法的第二阶段改用 UPGMA 凝聚聚类,将得到的结果与在第二阶段使用我们的 COCK 算法凝聚聚类的结果进行比较,表 1 显示了在数据集 TR12、RE1 和 WAP 上,这两种聚类方法结果的熵、纯度和 F1 值的比较.

从表 1 可以看出,在数据集 TR12、RE1 和 WAP 上,我们的 COCK 算法聚类结果的熵、纯度和 F1 值均大幅度优于 UPGMA 算法.

表 1 COCK 算法和 UPGMA 算法聚类结果比较

数据集	聚类算法	entropy	purity	F1
TR12	UPGMA	0.312 421	0.757 188	0.749 997
	COCK	0.261 032	0.785 942	0.788 731
RE1	UPGMA	0.384 723	0.621 002	0.512 056
	COCK	0.298 744	0.692 215	0.534 534
WAP	UPGMA	0.456 47	0.537 179	0.510 819
	COCK	0.295 957	0.717 949	0.680 952

虽然 UPGMA 的速度快于我们的 COCK 算法,但 UPGMA 聚类结果的质量比我们的 COCK 算法差很多.

5 结束语

CHAMELEON 算法复杂,在简化 CHAMELEON 算法的基础上提出了一个聚类算法 COCK,比 CHAMELEON 算法运算速度快,此 COCK 算法性能好,是一种性能优良的层次凝聚聚类算法,以很

大幅度优于 UPGMA 算法,而 UPGMA 目前被认为是文档聚类时最优良的层次凝聚聚类算法^[7].

参考文献:

- [1] ZHU Ye-hang. Research on document clustering algorithms[D]. Xi'an: Northwestern Polytechnical University, 2009.
- [2] Karypis G, Han Eui-Hong, Kumar V. Chameleon: hierarchical clustering using dynamic modeling[J]. Computer, 1999, 32(8): 68-75.
- [3] Karypis G, Han Eui-Hong, Kumar V. Chameleon: a hierarchical clustering algorithm using dynamic modeling[R]. Minneapolis, USA: [s. n.], 1999.
- [4] Guha S, Rastogi R, Shim K. Rock: a robust clustering algorithm for categorical attributes[C]//Proceedings of the 15th International Conference on Data Engineering. Washington, DC, USA: IEEE Computer Society, 1999: 512-521.
- [5] Guha S, Rastogi R, Shim K. Rock: a clustering algorithm for categorical attributes[J]. Information System, 2000, 25(5): 345-366.
- [6] Fisher D H. Knowledge acquisition via incremental conceptual clustering[J]. Machine Learning, 1987, 2(2): 139-172.
- [7] Zhao Ying, Karypis G. Hierarchical clustering algorithms for document datasets[R/OL]. [2014-04-09] (2008-10-29). <http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download>.

作者简介:

朱烨行 男,(1969-),博士研究生,讲师.研究方向为数据挖掘. E-mail:zhuyehang@126.com.

李艳玲 女,(1971-),博士研究生,副教授.研究方向为数据挖掘.

杨献文 男,(1967-),工程师.研究方向为数据挖掘.

(上接第 172 页)

- [3] TIA/EIA-644-A; Electrical characteristics of low voltage differential signaling (LVDS) Interface Circuits [EB/OL]. [2014-04-09]. http://www.tiaeia.org/index.php?main_page=product_info&path=1&products_id=4155.
- [4] Hyper transport technology consortium; HyperTransport I/OLink Specification Revision 3.00[S]. 2001.
- [5] EcriBgoati. 信号完整性分析[M]. 李玉山,译.北京:电子工业出版社,2005.
- [6] Behzad Razvai. 模拟集成电路设计[M]. 曾朝阳,赵阳,

方顺,译.北京:机械工业出版社,2005.

- [7] DAJohns, KMartin. Analog integrated circuit design [M]. New York: John Wiley & sons, 1997.

作者简介:

文艺女,(1990-),硕士研究生.研究方向为宽带时域电路设计及测试技术.

张健 男,(1985-),研究生,工程师.研究方向为 FPGA 中嵌入的模拟集成电路和高速 IO 接口设计.

E-mail:15011589626@163.com.