DOI:10.15918/j.tbit1001-0645.2008.01.017

基于凝聚的层次聚类算法的改进

石剑飞[1,2], 闫怀志[2], 牛占云[1]
(1. 北京理工大学 计算机网络攻防对抗技术实验室, 北京 100081; 2. 北京理工大学 软件学院, 北京 100081)

摘 要: 为提高基于凝聚的层次聚类算法的准确率. 在研究了空间级约束适用情况的基础上, 以Single Link算法为例, 验证了空间级约束条件对聚类结果的影响. 与实例级约束Single Link算法相比, 空间级约束Single Link算法只需较少约束条件即可达到较高准确率. 实验结果证明, 空间级约束可以有效提高聚类的准确率.

关键词: 聚类算法; 实例级约束; 空间级约束
中图分类号: TP 393 文献标识码: A 文章编号: 1001-0645(2008)01-0066-04

Improved Algorithm Based on Agglomerative of Hierarchical Clustering

SHI Jian-fei[1,2], YAN Huai-zhi[2], NIU Zhan-yun[1]
(1. Lab of Network Defense Technology, Beijing Institute of Technology, Beijing 100081, China;
2. School of Software, Beijing Institute of Technology, Beijing 100081, China)

Abstract: To enhance the accuracy of hierarchical clustering algorithm, based on a study of space-constrained application, and using the Single Link algorithm as an example, the impact of space-constrainment is certificated to the result. Compared with the case-constrain Single Link algorithm, space-constrained Single Link algorithm needs less constrained conditions to achieve higher accuracy. Experiments showed the space-constrainment can enhance the clustering accuracy effectively.

Key words: clustering; case-constrained; space-constrained

聚类就是将数据对象分组变成多个簇, 使同一个簇中的对象具有较高的相似性, 而不同簇中的对象具有较大的相异性. 良好的聚类方法产生的聚类结果具有簇内对象高度相似, 簇间对象很少相似的特性[1].

聚类过程的实质是寻找聚类目标函数最优解的优化过程. 基于层次的聚类方法[2]将数据对象在不同阶段组成不同粒度的簇, 并在簇的分裂和合并过程中不断改善聚类的效果, 以达到逐步求精的目的. 层次聚类法根据层次分解方向可以分为凝聚[3]的和分裂的层次聚类.

作者讨论了一种改进约束条件的凝聚层次聚类算法, 通过使用空间级约束条件改进实例级约束的层次聚类算法的性能, 以Single Link算法为例, 实现空间级约束的算法并验证改进后算法的性能.

1 实例级约束的局限性分析

虽然针对算法添加实例级约束的方法使聚类结果满足了所有约束条件, 却没有深入挖掘这些约束中的隐含信息[4]. 例如, 图1(a)为原始数据集, 实例有两个属性, 可以用平面图直观表示. 每个叉号代表一个数据. 图1(b)的两双线表示被连接的两个实例属于必相邻数据集. 图1(b)与图1(c)的聚类结果都满足所有的约束, 但可以看出, 图1(c)的聚类结果较好. 这是因为实例级别的约束隐性地改变了这两个聚类周围的实例间的关系, 也就是使一个空间的距离产生了变化. 不仅要使属于必相邻约束集中的实例对分配到同一个聚类中, 还要使距这两个

收稿日期: 2007-07-05
基金项目: 国家部委基础科研项目(20021823)
作者简介: 石剑飞(1982—), 男, 博士生, E-mail: jfshi0311@163.com; 闫怀志(1975—), 男, 副教授.

实例非常近的实例也分配到同一个聚类中. 必不相邻约束的情况类似, 不仅使属于必不相邻集中的实例不在同一个聚类中, 与这两个实例非常近的实例也应不在同一个聚类中. 上述实例级别约束聚类算法由于不能充分利用约束的隐含信息, 在约束个数特别少的情况下, 聚类准确率提高幅度小.

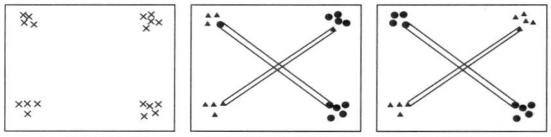

图 1　实例级约束图

Fig. 1　Case-constrainment

作者的实例级约束算法改进之处在于, 为每个实例分配聚类时检查这个实例是否在约束中存在, 然后在保证这个实例不会违反约束的条件下, 将其分配到最相似的聚类中. 但这样做经常会得到与图 1(b)所示结果类似的情况, 聚类结果与直观感受相差较远, 这是因为该种算法没有利用空间级别约束, 没有最大限度地使用约束蕴含的丰富信息, 因此, 尽管它与所有约束不冲突, 却与现实中的自然感受相冲突.

2　空间级约束聚类的分析与实现

空间级约束应当具备的特点为约束条件信息不仅给出两个实例之间的关系, 而且还隐含着它们的邻居间关系, 所以考虑利用这种约束聚类之前, 一定要检查实例是否满足该条件. 作者以 Single Link 算法为例, 介绍适合使用约束聚类的情况, 以及空间级约束聚类的实现方法, 并阐述改进后的约束 Single Link 算法. 约束情况如图 2 所示.

图 2　使用约束情况图

Fig. 2　Use constriction

如果将数据集明显地分为两个集合, 如图 2(a)所示, 则没有必要为聚类算法添加约束, 因为任何聚类算法处理该情况的准确率都很高. 从图 2(b)中可以看出数据是存在模式的, 但是聚类算法很难在没有约束帮助的情况下找到正确模式, 此时为使用约束聚类的最佳时机. 图 2(c)中, 数据分布极为混乱, 无信息量可言, 约束的作用微乎其微[5].

深度挖掘约束中所隐含的信息. 聚类算法通常需要维护一个距离矩阵, 该矩阵由实例间的距离组成. 作者提出的算法根据约束来更新该矩阵, 然后再按照普通的聚类步骤进行聚类, 这样就有效简化了算法的逻辑.

更新距离矩阵时, 作者首先更新被约束的实例对的距离, 更新基本原则如下:

原则 1　如果一个实例对属于必相邻集合, 那么应该使这两个实例之间的距离尽可能近.

原则 2　如果一个实例对属于必不相邻集合, 那么应该使两个实例之间的距离尽可能远.

基于上述原则, 更新其余实例间的距离时, 有如下两个原则:

原则 3　如果两个实例间的距离很近, 那么与这两个实例距离相近的实例间的距离也应该很近, 如图 3 所示.

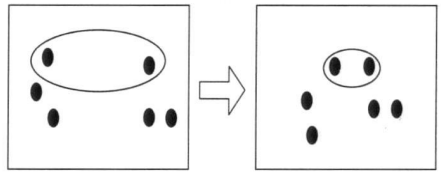

图 3　必相邻约束度实例之间距离的影响

Fig. 3　Effect of the distance between must-link cases

原则 4　如果两个实例间的距离很远, 那么与这两个实例距离相近的实例间的距离也应该很远, 如图 4 所示.

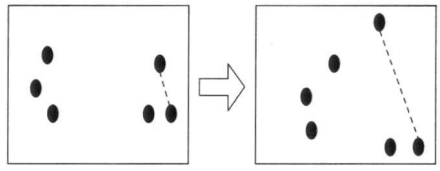

图 4　必不相邻约束度实例之间距离的影响

Fig. 4　Effect of the distance between cannot-link cases

给出该算法的实现细节. 如果约束只包含必相邻约束, 那么在更新距离矩阵时只需要减小部分实例间的距离. 具体来说, 首先把必相邻集合中实例对的距离在距离矩阵中修改为 0, 然后再更新其邻居间的距离.

如图 5 所示, A 与 B 的距离为 2, B 与 C 的距离为 3, 由于 B 与 C 属于必相邻集合, 所以将 B 与 C 的距离置为 0, 此时, 如果走 A→C→B 的路线, A 与 B 的距离只有 1, 就将 A 与 B 的距离更新为 1. 距离矩

阵的变化如图6所示.

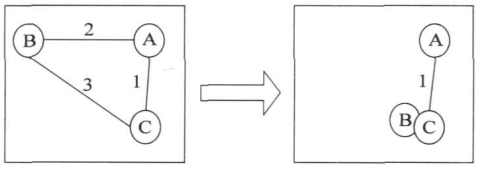

图5 距离更新算法

Fig. 5 Distance update algorithm

$$\begin{bmatrix}0&2&1\\2&0&3\\1&3&0\end{bmatrix}\Rightarrow\begin{bmatrix}0&2&1\\2&0&0\\1&0&0\end{bmatrix}\Rightarrow\begin{bmatrix}0&1&1\\1&0&0\\1&0&0\end{bmatrix}$$

图6 距离矩阵更新过程

Fig. 6 Distance matrix update course

步骤描述:① 给出距离矩阵;② 将必相邻元素之间矩阵对应的距离值变为0;③ 将必相邻元素邻居间的距离变为最短路径距离.

可以看出,这种算法与多源最短路径算法非常接近[9],可得到时间复杂度为 $O(n^2c)$ 的距离矩阵更新算法,其中 c 为必相邻矩阵中涉及到的不重复实例的个数,而不是原算法的 $O(n^3)$. 又因为 $c<N$,所以本算法要优于多源最短路径算法.

给出约束Single Link算法之前,首先给出Single Link算法的具体步骤.

步骤1 将所有的实例均初始化为一个聚类.

步骤2 计算所有实例间的距离,从而得到距离矩阵.

步骤3 选择聚类之间距离最小的两个聚类,并将其融合. 以分别来自两个聚类中的实例间的最小距离为聚类距离. 融合的方法是删除原有的两个聚类,创建一个新的聚类,这个聚类中的元素是原有两个聚类实例的合集.

步骤4 循环步骤3,直到聚类只剩一个.

下面给出约束Single Link算法.

步骤1 将所有的实例均初始化为一个聚类.

步骤2 计算距离矩阵.

步骤3 根据必相邻集合,将距离矩阵中相应的实例对的距离更新为0.

步骤4 根据改进算法,更新距离矩阵.

改进算法如下:

对于距离矩阵 $\boldsymbol{I}(i,j)$ $\exists j\neq i,(i,j)\in E_{\text{must}}$

for $k\in\boldsymbol{I}$, for $i\in\{1:n\}$, for $j\in\{1:n\}$

$$D_{ij}=\min(D_{ij},D_{ik}+D_{kj})$$

然后遍历距离矩阵,如果有实例对距离为0,而此实例对又不属于必相邻集合,则将此实例对加入必相邻集合 E_{must}.

步骤5 与Sinlgle Link算法步骤3相同.

步骤6 循环步骤5,直到聚类只剩一个.

约束中包含必不相邻约束的处理方法与包含必相邻约束的处理方法类似,只是根据必不相邻集合,将距离矩阵中相应必不相邻实例对距离更新为 $D_{ij}=\max(D_{ij},D_{ik}+D_{kj})$. 遍历距离矩阵,如果有实例对距离为最大,而此实例对又不属于必不相邻集合,则将此实例对加入必不相邻集合 E_{cannot},其余步骤类似.

但是如果不进一步处理,在修正距离矩阵时会存在冲突问题. 例如,若一个实例A同时与两个其他实例B,C属于必相邻关系,但B,C属于必不相邻关系;又如,若一个实例A同时与两个其他实例B,C属于必不相邻关系,但B,C属于必相邻关系等.

如果同时根据必相邻和必不相邻两种约束更新矩阵,时间复杂度太大,所以提出如下冲突解决方案.

预先人为定义与目标实例相关联的其他实例的影响度系数,一旦发生冲突,执行影响度系数高的实例关系,忽略其他关联关系;如果系数相等,对于所涉及关系,采取全部忽略的办法.

3 实验及结果分析

为了验证约束Single Link算法有效性,作者使用了两组人工数据集进行测试,图7为第1组数据的二维分布图,可以看出数据分为两类,呈圆圈型.

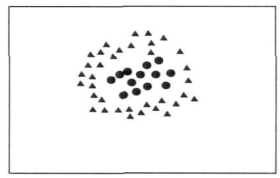

图7 合成数据集1,圆圈型

Fig. 7 Synthesis data set 1, circle

图8为实验结果,可以看出,与实例级约束的Single Link算法相比,带空间级约束的Single Link算法只需要较少的约束条件即可以达到较高的准确率,随着约束个数的增加,两种算法的准确率都趋近于1.

图9为第2组人工数据集的人工分布图,数据分为两类,成云图形状. 图10是对第1组数据集聚类的实验结果,这与第1组数据的测试结果类似,空

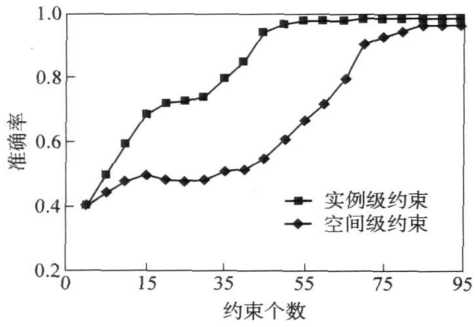

图 8　对数据集 1 的测试结果

Fig. 8　Data set 1 testing result

间级约束的准确率要明显优于实例级约束.

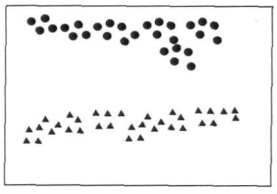

图 9　人工数据集 2, 云图

Fig. 9　Artificial data set 2, cloud

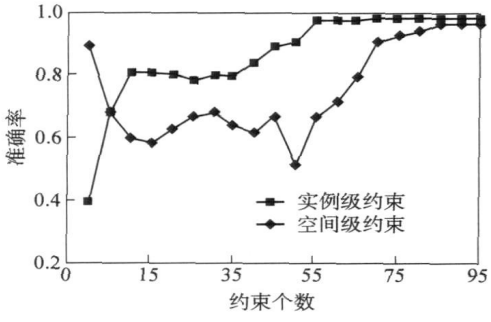

图 10　对数据集 2 的测试结果

Fig. 10　Data set 2 testing result

4　结　论

作者讨论了对层次聚类算法进行实例级约束的方法和局限性, 提出了空间级约束的方法及其适用范围, 并将该理论应用到 Single Link 算法中, 最后对算法进行了测试与分析. 实验证明, 在一定条件下, 相对与实例级约束而言, 可以提高聚类的准确率.

对层次聚类算法约束的改进具有理论和实际意义, 尚有一些问题需要继续深入探索. 如约束中的冲突消解, 实例间关联系数定义的客观性, 聚类逻辑的改进及实现约束条件更大的鲁棒性等都是下一步的研究方向.

参考文献:

[1] Sanguthevar R. Efficient parallel hierarchical-clustering algorithms[J]. IEEE Transactions on Parallel and Distributed Systems, 2005, 16(6): 497－502.

[2] Kanungo T, Mount D M, Netanyahu N, et al. A local search approximation algorithm for *K*-means clustering [J]. Computational Geometry: Theory and Applications, 2004, 28: 89－112.

[3] Margaret H D. 数据挖掘教程[M]. 郭崇慧, 译. 北京: 清华大学出版社, 2005.
Margaret H D. Data mining[M]. Guo Chonghui, transl. Beijing: Tsinghua University Press, 2005. (in Chinese)

[4] Dan K, Sepandar D K, Christopher D M. From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering, CA 94305－9040[R]. Stanford, USA: Department of Computer Science, Stanford University, 2003.

[5] Fraley C, Raftery A E. How many clusters? which clustering method? answers via model-based cluster analysis [J]. The Computer Journal, 1998, 41: 578－588.

[6] 王银燕. 基于二度量的单播最短路径算法[J]. 计算机工程, 2007, 33(5): 89－90.
Wang Yinyan. Algorithm for two-metric unicast shortest path[J]. Computer Engineering, 2007, 33(5): 89－90. (in Chinese)

[7] 孙强. 求带单一限制条件的单源多权最短路径的一个算法[J]. 计算机工程, 2002, 28(8): 135－137.
Sun Qiang. A new algorithm of the shortest path problem with single restriction and multiple weights[J]. Computer Engineering, 2002, 28(8): 135－137. (in Chinese)

(责任编辑: 康晓伟)