文章编号: 1001-0920(2007)04-0469-04

基于免疫原理的一种动态数据聚类方法

张 雷[1,2], 李人厚[1]
(1. 西安交通大学 系统工程研究所, 西安 710049; 2. 河南科技大学 电信学院, 河南 洛阳 471003)

摘 要: 提出一种基于免疫原理的动态聚类算法,它能在噪声环境下得到任意形状的聚类,并能有效地实现动态聚类操作. 算法包括 3 个步骤:首先基于生物免疫机制得到一个反映当前数据分布特征的抗体集合;然后使用最小生成树方法得到聚类的初始结构;最后针对数据库的更新设计了动态聚类算法. 仿真结果表明了该算法实现动态聚类的有效性.

关键词: 免疫原理; 动态聚类; 聚类
中图分类号: TP18 **文献标识码**: A

Dynamic clustering algorithm based on immune principle

ZHANG Lei[1,2], *LI Ren-hou*[1]
(1. Institute of System Engineering, Xi'an Jiaotong University, Xi'an 710049, China; 2. School of Electronics and Information Engineering, He'nan University of Science and Technology, Luoyang 471003, China. Correspondent: ZHANG Lei, E-mail: lzhang@stu.xjtu.edu.cn)

Abstract: A dynamic clustering algorithm based on immune principle is proposed. Arbitrary shape clusters are generated in the presence of noise by using the algorithm, and dynamic clustering is implemented efficiently. A set of antibodies are obtained based on biology immune mechanism reflecting the distributing information of current data set. The initial clustering structure is constructed by using the minimum spanning tree method. The dynamic clustering algorithm is developed to update the clusters. Experiment results show that the effectiveness of the proposed algorithm.

Key words: Immune principle; Dynamic clustering; Clustering

1 引 言

聚类是将数据集进行分组,并使得同一组数据之间的相似性尽可能大,不同组数据之间的相似性尽可能小. 它在模式识别、图像处理和数据分析等领域得到了广泛应用,但在许多实际应用中,数据库通常是动态变化的. 动态聚类方法就是要解决这种数据随时间变化的聚类问题.

针对动态数据库实施聚类操作已引起了广泛关注,并提出了相应的聚类算法[1-4]. 增量式 DBSCAN[1]是一种基于密度标准的增量式聚类算法,对于数据库的更新它只需针对密度受影响区域实施操作. 但该算法需要构建复杂的索引树来进行相似性搜索操作,其计算成本较大. BIRCH 算法也是一种支持增量式聚类的算法[2],它引入了聚类特征和聚类特征树(CF 树)两个概念,用于概括聚类描述,但该方法不能处理任意形状的聚类. 因为算法中采用了半径(或直径)的概念来控制聚类的边界,所以只能得到球形的聚类. 文献[4]提出了一种动态语义聚类方法,用于获得网络用户兴趣模式,帮助并改进网站的设计. 由于用户兴趣是随时间动态变化的,该方法基于马尔可夫模型并通过用户浏览页面来动态跟踪用户兴趣变化. 聚类的结果是一个动态多层用户的兴趣模型,用于表示用户的一般和特殊兴趣.

文献[5]提出了一种新的聚类模型,它结合人工免疫原理和稀疏分布式记忆模型,并采用协同进化遗传算法得到演化的聚类. 该方法能动态跟踪数据集中聚类结构的变化. 在该算法中,每个数据相当于一个抗原,而每个抗体描述一个聚类,抗体的识别半径描述了聚类的大小. 该算法的不足是根据适应值函数的定义,其运算成本较高.

本文提出了一种基于人工免疫原理的动态聚类算法,它通过抗体种群的分布来反映一个聚类的结

收稿日期: 2005-12-18; **修回日期**: 2006-03-16.
作者简介: 张雷(1974—),男,河南洛阳人,博士生,从事智能计算、数据挖掘等研究; 李人厚(1935—),男,浙江宁波人,教授,博士生导师,从事 CSCW 和智能控制等研究.

构特征. 当数据集进行更新时,通过更新抗体种群来改变当前的聚类结构,并实现动态聚类. 仿真结果表明,它能在噪声环境下得到任意形状的聚类,并能有效地实现动态聚类操作,同时该算法在动态聚类的速度性能上优于增量式 DBSCAN 算法.

2 基于免疫原理的动态数据聚类算法的结构

算法的结构如图 1 所示,包括 3 个步骤:首先针对当前数据库基于生物免疫机制,得到反映数据库分布特征的抗体集合;然后利用最小生成树方法得到聚类的初始结构;最后针对数据库的更新设计动态聚类算法. 其中前 2 个步骤用于得到初始聚类结构,而第 3 步则用于实现聚类的更新,它针对更新的数据,基于已有聚类结构实施结构的更新操作.

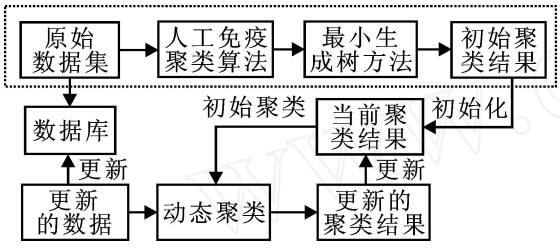


图 1 本文方法的结构图

定义 1 抗原和抗体之间的亲和度,即

$$w_{ij} = \exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right). \tag{1}$$

式中:d_{ij} 表示抗体 Ab_i 和抗原 x_j 之间的欧几里得距离;σ_i 为抗体 Ab_i 的识别半径,该参数用于控制亲和度在距离上的衰减率,因而反映了该抗体能有效识别周围抗原的区域大小.

定义 2 抗体的识别区域. 将抗体 Ab_i 的识别区域 $R_a(Ab_i)$ 定义为该抗体在形状空间中可有效识别抗原的范围,如下式所示:

$$R_a(Ab_i) = \{x_j \in X \mid d_{ij} \leq \sigma_i\}, \tag{2}$$

即该抗体能有效识别该区域内的抗原. d_{ij} 和 σ_i 的含义同定义 1.

定义 3 抗体的激励强度 S_t. 假定 Ab_i 所能识别的抗原数目为 n,则 Ab_i 的激励强度定义为

$$S_t(i) = \sum_{j=1}^{n} w_{ij}, \tag{3}$$

它反映了该抗体能有效识别抗原的数目.

定义 4 抗体之间的亲和度. 抗体 Ab_i 和 Ab_j 之间的亲和度 A_{ij} 定义为它们之间欧几里得距离的倒数.

3 初始聚类的结构

数据集中的每个数据对应于算法中的一个抗原,它们与由抗体组成的网络进行相互作用. 算法中所使用的免疫隐喻主要有亲和度成熟、克隆增殖和网络抑制. 亲和度成熟是指,抗原呈递给抗体网络,抗体进行超变异操作,使得更有效地识别抗原;克隆增殖是指,受抗原刺激激励强度较高的个体被选择出来进行克隆操作,使抗体网络的规模增长;网络抑制是指,如果两个抗体之间的亲和度高于设定的阈值,则激励强度低的抗体从种群中移去.

算法步骤如下:

Step1: 初始化. 从数据库 X 中随机选择 n_1 个数据作为初始抗体种群 C,并取 X 中数据之间的平均距离作为每个抗体的识别半径 σ.

Step2: 对 X 中每个抗原 $x_j(j = 1,2,\cdots,n)$ 执行以下步骤:

Step2.1: 计算它与抗体种群 C 中每个抗体 Ab_i 的距离 d_{ij}.

Step2.2: 若 C 中存在抗体使得 x_j 位于它们的识别区域内,则基于下式更新这些抗体的激励强度:

$$S_t(i) = S_t(i) + w_{ij}; \tag{4}$$

否则将抗原 x_j 的复制作为新产生的抗体,加入到抗体种群 C 中,并计算其激励强度值.

Step2.3: 克隆和超变异操作. 选择与该抗原 x_j 亲和度最高的抗体进行克隆操作,克隆数目为 1. 进而对克隆个体 Ab 实施超变异操作,即

$$Ab' = Ab + rand * mr * (x_j - Ab). \tag{5}$$

式中:rand 是 0 ~ 1 之间的随机数,mr 为变异率.

Step2.4: 种群数目控制机制. 将抗体种群基于激励强度进行排序,保留激励强度最高的 N_p 个抗体,作为下一代的抗体种群 C.

Step2.5: 当迭代次数达到设定值时算法结束,否则转 Step2.1.

通过上述算法得到的抗体种群能反映数据集的分布特征,但考虑到数据集中存在噪声,所以选择其中激励强度值高的抗体来确定聚类结构. 由于数据集是动态的,抗体代表的是否是噪声数据或孤立点,还需通过数据集的更新来确定. 因而,可将另一部分抗体作为动态聚类的初始抗体种群. 记 C 中激励强度大于阈值 T_a 的抗体为集合 P_d,其余的则记为集合 P_c.

对于集合 P_d,利用最小生成树得到初始聚类划分的步骤如下:

Step1: 计算集合 P_d 中个体之间的相互距离,并将它们按升序进行排列,得到构成最小生成树候选边的集合.

Step2: 从得到的候选边中,由最短的边开始,选择不和已选边构成回路的边,直到被选边的数目达到 $(N_r - 1)$ 为止,其中 N_r 为 P_d 中抗体的数目.

对于得到的最小生成树,通过删除其不一致的

边(即长度明显大于相邻的边)可得一些子图,每个子图对应于一个聚类;然后,将数据库中的每个数据划分到与其距离最近抗体所在的聚类,并标记为类的标签,于是得到初始的聚类划分结果.

确定不一致边的最简单方法是将长度大于设定阈值的边作为不一致边,但该方法的阈值有时较难确定. 本文采用文献[6]的方法,它基于局部密度标准来确定不一致边,同时能自动确定聚类的数目.

4　基于免疫原理的动态聚类算法

对上述的最小生成树(MST)进行分割,每个子图对应于一个聚类. 为了迅速确定新增数据对当前聚类结构的影响,建立一个抗体的连接图(ALG),通过 ALG 的变化来实施动态聚类操作. ALG 中相连的节点表示一个聚类结构,数据集中每个数据通过划分到与其距离最近的节点所在聚类来实现聚类划分. ALG 是对数据集中隐含聚类结构的概括描述,当数据集进行更新时,它能较为迅速地确定当前受到影响的聚类. 另外,通过 ALG 还可方便地实施聚类结构的更新操作. ALG 中增加新节点表示聚类形状的改变,而连接图之间的合并则表示聚类的合并操作.

ALG 的建立方法是:对于 MST 进行分割得到每一个子图,若节点两两之间的距离小于连接阈值,则将它们进行连接.

每个更新数据对应于动态聚类算法中的一个抗原,它们与抗体所组成的网络相互进行作用,通过对抗体种群的更新来实现动态聚类. 数据集中新增加一个数据所引起的聚类结构变化有以下 3 种情形:1) 若数据属于已有聚类,则合并到该聚类中;2) 若数据不属于任何已有聚类,则可能增加一个新聚类;3) 数据的加入引起两个或更多聚类的合并. 具体情形如图 2 所示.

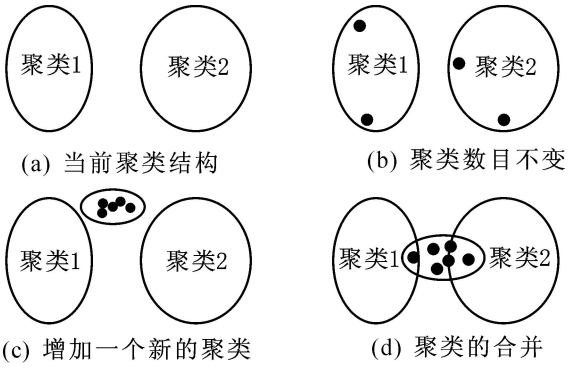


图 2　更新引起聚类结构变化的几种情形

人工免疫动态聚类算法的具体步骤如下:

Step1: 将初始聚类所得到的抗体集合 Pc 作为初始抗体种群 C,对算法中相关参数赋初值.

Step2: 对每个更新数据集中的抗原执行如下操作:

Step2.1: 若 C 中存在的抗体能识别该抗原,则更新这些抗体的激励强度;否则将该抗原的复制作为新生成的抗体加入种群 C 中,并计算新抗体的激励强度.

Step2.2: 由于抗体激励强度的更新,C 中可能会有部分抗体因更新的激励强度高于设定阈值,生成抗体的集合 S_a,并将它们从 C 中移除. 若集合 S_a 为空,则转 Step2.5;否则继续执行.

Step2.3: 计算 S_a 中每个抗体与 ALG 中所有抗体的亲和度,若它们之间的亲和度都小于抑制阈值,则将该抗体作为新节点加入 ALG 中. 若无新节点加入 ALG 中,则转 Step2.5;否则继续执行.

Step2.4: 更新 ALG. 对于 ALG 中每个新加入的抗体 Z,计算它与 ALG 中其他抗体的亲和度,若高于连接阈值,则建立新的连接关系. 设 n 为新建连接的数目,如果 $n = 0$,表示 Z 不与周围的抗体相连接,则 Z 作为一个独立节点,对应于新增加一个聚类的情形;如果 $n = 1$,则 Z 加入到某个连接图中,对应于改变某个聚类形状的情形;当 $n > 1$ 时,若 Z 与同一个聚类连接,则对应于某个聚类形状的改变,若 Z 与不同的聚类连接,则对应于两个(或更多)聚类进行合并的情形.

Step2.5: 若 ALG 没有改变,则只对更新的数据进行聚类划分操作,即将其划分到 ALG 中与其亲和度最高的抗体所属聚类中;否则,对当前数据集重新进行聚类划分操作.

5　仿真实验

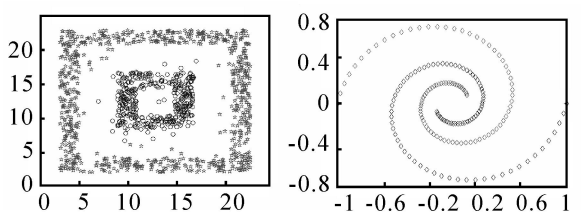
本节通过几个数据集的仿真实验来评价算法的性能,并与相关的增量式聚类算法 DBSCAN[1] 进行性能比较. DBSCAN 算法是一种广泛应用的基于密度的聚类算法,而增量式 DBSCAN 算法可针对动态变化数据库实现增量式聚类. 本文算法和增量式 DBSCAN 算法都采用 MATLAB 程序来实现,运行在 2GMHz,内存 256M 的 Pentium IV 微机上.

5.1　算法的有效性

为了评价算法生成任意形状聚类的能力,本文采用两个合成数据集进行仿真实验. 数据集 1 为两个同心正方形环,其中包含噪声数据,而数据集 2 为双螺旋线. 它们聚类的结果分别如图 3(a) 和图 3(b) 所示,图中用不同颜色表示不同聚类. 可以看到本文算法能针对任意形状聚类分布的数据集得到正确的聚类结果.

5.2　动态聚类

本节评价了算法实施动态聚类的性能,实验结



(a) 数据集1 (b) 数据集2

图 3 聚类结果

果表明,本文算法在聚类的速度上优于增量式DBSCAN算法.仿真实验采用二维合成数据集,数据由 2 000 个更新到 3 000 个,聚类数目由 4 个增加到 6 个,每个聚类中的数据数目基本相等.

首先,当数据集更新时,对分别采用本文算法重新进行聚类和实施动态聚类方法进行性能比较.比较的指标采用下式所定义的加速因子:

$$S_f = \frac{\text{重新聚类的运行时间}}{\text{动态聚类的运行时间}}, \qquad (6)$$

它反映了两种方法聚类时间的比值.

初始数据集的规模为 2 000 个数据,每次数据更新的数目为 100 个,实验结果如图 4 所示.从结果可知,采用增量式动态聚类方法可显著减少聚类的时间,并且当数据集规模较大时,时间优势更为明显.

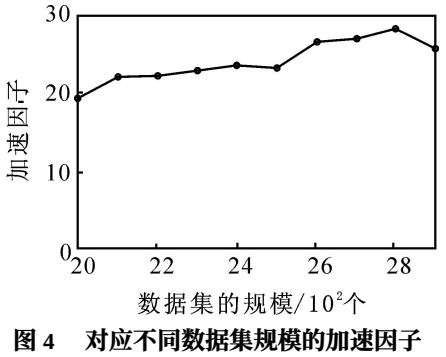


图 4 对应不同数据集规模的加速因子

其次,针对运行时间和增量式DBSCAN算法进行了性能比较.原始数据集包括 2 000 个数据,更新数据的数目是从 100 个到 1 000 个,实验结果如图 5 所示.由结果可知,本文算法在动态聚类的运行时间上优于增量式DBSCAN算法.对于增量式DBSCAN算法,虽然数据密度的更新操作只在更新数据的邻居点内进行,但对于数据集规模很大的情形,数据点邻居搜索的计算量较大,因而动态聚类的时间较长.

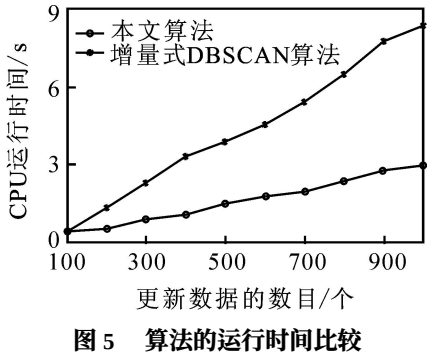


图 5 算法的运行时间比较

6 结 论

本文提出了一种基于自然免疫原理的动态聚类算法.该算法将数据库中的每个数据视为抗原,能产生一个反映数据库分布特征的抗体集合,用于表示聚类的结构;进而利用最小生成树方法确定数据库的聚类数目及初始聚类结构;然后针对数据库的动态更新设计了动态聚类算法.仿真结果表明,该算法能产生任意形状的聚类,并能有效实现动态聚类,同时本文算法在动态聚类的速度性能上优于增量式DBSCAN 算法.

参考文献(References)

[1] Ester M, Kriegel H P, Sander J, et al. Incremental clustering for mining in a data warehouse environment [C]. Proc of 24th Int Conf on Very Large Data Base. New York, 1998: 323-333.

[2] Zhang T, Ramakrishnan R, Livny M. BIRCH: An efficient data clustering method for very large databases [C]. Proc of ACM SIGMOD Int Conf on Management of Data. Montreal, 1996: 103-114.

[3] Karypis G, Han E H, Kumar V. Chameleon: Hierarchical clustering using dynamic modeling [J]. IEEE Computer, 1999, 32(8): 68-75.

[4] Chen J J, Gao J, Liao B S, et al. Dynamic semantic clustering approach for web user interest [C]. GCC Workshops. Wuhan, 2004: 59-66.

[5] Hart E, Ross P. Clustering moving data with a modified immune algorithm [C]. Applications of Evolutionary Computing. Springer: Boers E, 2001: 394-404.

[6] Bezerra G B, de Castro L N. Bioinformatics data analysis using an artificial immune network[C]. Proc of 2nd Int Conf on Artificial Immune Systems. Edinburgh, 2003: 22-33.

[7] Timmis J, Neal M. A resource limited artificial immune system for data analysis [J]. J of Knowledge-based Systems, 2001, 14(3): 121-130..

[8] Nasraoui, Dasgupta D, Gonzalez F. An artificial immune system approach to robust data mining [C]. Genetic and Evolutionary Computation Conf (GECCO). New York, 2002: 356-363.

[9] Ng R, Han J. Efficient and effective clustering methods for spatial data mining[C]. Proc 20th Int Conf on Very Large Data Base. Santiago, 1994: 144-155.