ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.9, September 2008, pp.2339−2348
DOI: 10.3724/SP.J.1001.2008.02339

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种基于局部密度的分布式聚类挖掘算法*

倪巍伟[1+], 陈 耿[2], 吴英杰[1], 孙志挥[1]

[1](东南大学 计算机科学与工程学院,江苏 南京 210096)

[2](南京审计学院 审计信息工程实验室,江苏 南京 210029)

Local Density Based Distributed Clustering Algorithm

NI Wei-Wei[1+], CHEN Geng[2], WU Ying-Jie[1], SUN Zhi-Hui[1]

[1](School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

[2](Laboratory of Audit Information Engineering, Nanjing Audit University, Nanjing 210029, China)

+ Corresponding author: E-mail: wni@seu.edu.cn

Ni WW, Chen G, Wu YJ, Sun ZH. Local density based distributed clustering algorithm. *Journal of Software*, 2008,19(9):2339−2348. http://www.jos.org.cn/1000-9825/19/2339.htm

Abstract: Distributed clustering is an effect method for solving the problem of clustering data located at different sites. Considering the circumstance that data is horizontally distributed, algorithm LDBDC (local density based distributed clustering) is presented based on the existeding algorithm DBDC (density based distributed clustering), which can easily fit datasets of high dimension and abnormal distribution by adopting ideas such as local density-based clustering and density attractor. Theoretical analysis and experimental results show that algorithm LDBDC outperforms DBDC and SDBDC (scalable density-based distributed clustering) in both clustering quality and efficiency.

Key words: distributed clustering; local density based clustering; local clustering model; density attractor; high dimension data

摘 要: 分布式聚类挖掘技术是解决数据集分布环境下聚类挖掘问题的有效方法.针对数据水平分布情况,在已有分布式密度聚类算法 DBDC(density based distributed clustering)的基础上,引入局部密度聚类和密度吸引子等概念,提出一种基于局部密度的分布式聚类算法——LDBDC(local density based distributed clustering).算法适用于含噪声数据和数据分布异常情况,对高维数据有着良好的适应性.理论分析和实验结果表明,LDBDC 算法在聚类质量和算法效率方面优于已有的 DBDC 算法和 SDBDC(scalable density-based distributed clustering)算法.算法是有效、可行的.

关键词: 分布式聚类;局部密度聚类;局部聚类模型;密度吸引子;高维数据

中图法分类号: TP311 **文献标识码**: A

* Supported by the Doctor Science Research Foundation of the Ministry of Education of China under Grant No.20040286009 (国家教育部高等学校博士学科点科研基金), the Natural Science Foundation of Jiangsu Province of China under Grant No.BK2006095 (江苏省自然科学基金)

Received 2007-06-07; Accepted 2007-11-05

近年来,随着卫星遥感、传感器网络、高能物理研究等技术的发展,大量的数据被存储在数据库中,这些数据具有维度高、数据分布稀疏、噪声数据多的特点.在很多应用场合下,这些数据分布在不同的节点上,由于业务模式和数据规模的因素,几乎不可能将这些分布在不同节点的数据集中起来进行挖掘处理.分布式挖掘是解决这一问题的有效方法,在分布式聚类挖掘方面,已取得了一系列成果.已有的这些分布式聚类算法各有特点,文献[1,2]中提出的基于 *k*-Means 的分布式聚类算法,具有效率高的特点,但对数据的分布形状有限制;相比之下,基于密度的 DBDC(density based distributed clustering)算法[3]虽然效率稍差,但对非均匀分布数据的聚类质量较好.在对高维数据的处理能力上,上述算法均有缺陷.

高维空间数据有如下特点:① 数据分布稀疏、噪声数据较多;② 当高维空间维度高达一定程度时,对给定数据点,距其最近的数据点与最远数据点间的距离随着维度的增加渐趋于 0,在此称为“差距趋零现象”[4].并且,这些特点随着维度的增加更趋明显.

本文针对高维数据分布式聚类问题,以 DBDC 算法为基础,采用局部密度聚类思想,提出一种有效的分布式密度聚类算法 LDBDC(local density based distributed clustering).此算法可以有效地解决高维空间水平数据分布聚类挖掘问题.

1 相关工作

1.1 数据高维性问题

近年来,在数据挖掘研究领域,数据高维性问题引起了研究者的广泛关注,文献[4]通过理论分析和实验验证得出了关于高维空间数据的 3 个结论:

(1) 给定数据集中的查询点 p,随着数据集维度的增大,p 到最近邻数据点的距离越来越接近 p 与距其最远数据点间的距离,即维度的增加使得数据点间的区别在距离意义上变模糊了,出现了距离趋近现象;

(2) 通过对合成数据集的实验分析得出:对维度为 15 的数据集,到给定数据点的最近和最远数据点的差距就可能变得模糊;

(3) 一些用于快速查找最近邻数据点的技术(例如 *R**-树等)在高维空间中可能会失效.

多数聚类算法都直接或间接地以数据对象间的距离差别来判别数据对象的聚簇归属,距离趋近现象的存在使得算法对涉及距离的参数选取特别敏感,聚类质量无法得到保证.

1.2 DBDC算法及存在的问题

分布式聚类挖掘算法 DBDC 分为 3 个阶段:首先,各个节点 S_i 运行 DBSCAN(density based spatial clustering of applications with noise)算法[5]对数据集 D_i 进行聚类分析,对生成的每个聚簇 $C_j(C_j\subseteq D_i)$,用一组满足定义要求的特殊核心点(specific core points)来表示,节点 S_i 上所有聚簇对应的特殊核心点集就构成了 S_i 的局部聚类模型;随后,各个节点将其局部聚类模型发送到协调节点,在协调节点上,对这些特殊核心点再运行 DBSCAN 算法,生成全局聚类模型;最后,协调节点,将全局聚类模型反馈给各个节点 S_i,以更新各个数据点的类属性.算法的相关定义如下:

定义 1. 完全特殊核心点集.设 C 为对数据集 D 运行 DBSCAN 算法(参数 *Eps* 和 Min*Pts*)生成的聚簇,则用 $Cor_C\subseteq C$ 来表示属于聚簇 C 的核心点(核心点定义见 DBSCAN 算法),称满足下列条件的核心点集 $SCor_C\subseteq C$ 为聚簇 C 的完全特殊核心点集:

- $SCor_C\subseteq Cor_C$;
- $\forall s_i,s_j\in SCor_C: s_i\notin N_{Eps}(s_j), N_{Eps}(s_j)$表示数据点 s 的 *Eps* 邻域内的数据点集合;
- $\forall c\in Cor_C \quad \exists s\in SCor_C: c\in N_{Eps}(s)$.

采用二元组(s,ε_s)构成节点 k 的局部聚类模型:

$Local_k:=\bigcup_{i\in 1..n}\{(s,\varepsilon_s)\mid s\in SCor_{C_i}\}$,其中$\varepsilon_s:=Eps+\max\{dist(s,s_i)\mid s_i\in Cor \text{ and } s_i\in N_{Eps}(s)\}$.

算法 DBDC 存在以下缺陷:

(1) 误判噪声点

特殊核心点并不反映局部噪声数据信息,这些局部噪声点可能属于全局聚类模式的某个聚簇.

(2) 对高维数据的聚类质量

构成局部聚类模型的二元组(s,ε_s)仅用核心点 s 及其最大半径ε_s来表征以 s 为中心、ε_s为半径范围内的所有数据点,存在掩盖这一范围数据点分布密度特征的可能,对于数据均匀分布情况,二元组的表征效果较为理想,但在高维环境下,数据分布稀疏以及距离趋近现象使得简单地以(s,ε_s)作为代表对象参与全局聚类,容易导致聚类质量的偏差.

(3) 算法参数设置困难

DBSCAN 算法需要预定义算法的参数 *Eps* 和 Min*Pts*,在文献[6]中已经分析了高维空间数据环境下,参数 *Eps* 设置的困难以及对聚类质量可能产生的影响,*Eps* 设置的微小差异,可能导致最终全局聚类结果的巨大偏差.算法 DBDC 在各个节点上需要进行第 1 轮预设参数 *Eps* 和 Min*Pts*,以生成局部聚类模型,生成局部聚类模型后,需要进行第 2 轮全局聚类的参数 *Eps* 和 Min*Pts* 设置,这时,全局聚类参数 *Eps* 更是没有任何先验知识可供参考.高维空间效应和 DBDC 算法需要两轮设置参数 *Eps*,导致 DBSCAN 算法中半径参数 *Eps* 设置困难的负面影响进一步恶化.

1.3 SDBDC算法及存在的问题

针对 DBDC 算法忽略局部噪声点影响的缺陷,文献[7]提出分布式聚类算法 SDBDC(scalable density-based distributed clustering),算法的聚类质量与各节点发送到中心节点的数据对象的数目有关,相关定义如下:

定义 2. 静态代表点质量(static representation quality)对数据集中数据点 o 及邻域半径ε,

$$\mathrm{StatRepQ}(o,\varepsilon)=\sum_{o_i\in N_\varepsilon(o)}\varepsilon-dist(o,o_i)\ .$$

定义 3. 动态代表点质量(dynamic representation quality)对数据集中数据点 o 及邻域半径ε,

$$\mathrm{DynRepQ}(o,\varepsilon,\mathrm{Rep}_i)=\sum_{\substack{o_i\in N_\varepsilon(o)\\ \forall r\in Rep_i:o_i\notin N_\varepsilon(r)}}\varepsilon-dist(o,o_i),$$

其中,Rep_i 表示节点 i 上已选出的代表点集合.

定义 4. 覆盖半径和局部代表点权值.假设 $\mathrm{Rep}_i=\{r_{i_1},\ldots,r_{i_n}\}$,为节点 i 上选出的前 n 个代表点,

$$\mathrm{CovRad}(r_{i_{n+1}},\varepsilon,\mathrm{Rep}_i)=\max\{\varepsilon\text{-}dist(o,r_{i_{n+1}})|\forall o\in D_i\forall r\in\mathrm{Rep}_i:o\in N_\varepsilon(r_{i_{n+1}})\wedge o\notin N_\varepsilon(r)\},$$

$$\mathrm{CovCnt}(r_{i_{n+1}},\varepsilon,\mathrm{Rep}_i)=\left|\{o|\forall o\in D_i\forall r\in\mathrm{Rep}_i:o\in N_\varepsilon(r_{i_{n+1}})\wedge o\notin N_\varepsilon(r)\}\right|.$$

SDBDC 算法思想是:在各个节点 i 上选取一定规模的数据点(例如,占 D_i 的比例为α,$0<\alpha\le1$)作为 Rep_i,要求所选代表点的静态或动态代表点质量(Rep_i 为空时采用静态代表点质量来衡量,否则采用动态代表点质量衡量)在 D_i 中位于前 100α%.在中心节点上,各个代表点 r 的邻域半径设置为$\varepsilon+\mathrm{CovRad}(r)$,对这些代表点进行密度聚类,生成全局聚类模式.

由分析可知,SDBDC 算法对局部噪声点的处理主要通过用户设定合适的α值,控制各个节点向中心节点提交代表点的规模,α 值设定得越大,聚类质量越好.尽管 SDBDC 算法对 DBDC 算法进行了改进,考虑了各个节点上局部噪声点的影响,但算法 SDBDC 仍存在以下问题:

(1) 用户很难设定合适的提交参数 α,α 值设置得越大,对局部噪声处理效果越好,但效率也越低;既要保证对局部噪声数据的有效处理,又要求算法有较高的效率,很难找到满足条件的 α 值;

(2) SDBDC 算法并没有解决 DBDC 算法参数设置困难的问题,算法所引入的静态代表点质量、动态代表点质量、覆盖半径等对聚类质量有直接影响的定义都严重依赖于邻域半径参数的合理设定.第 2.3 节已经分析了在高维数据环境下,邻域半径参数设置的困难.

1.4 局部密度聚类算法

针对高维空间聚类问题,我们在文献[6]中提出了一种单参数局部密度聚类算法 *k*-PCLDHD,算法只需输入

一个与距离无关的参数 k,引入了 k 邻域点集、k 邻域半径,对 DBSCAN 算法中的核心点、密度相连等概念进行了重新定义,聚类原理与 DBSCAN 算法相同,相关定义如下:

定义 5. k 邻域距离(k-distance).对 $p\in D$,p 的 k 邻域距离定义为 $dist_k(p)$,要求存在 $q_1,q_2,\ldots,q_k\in D$,且对任意 $i(1\le i\le k)$,有 $dist(p,q_k)\le dist_k(p)$,且 $dist(p,q_i)\le dist(p,q_{i+1})$.

数据对象 p 的 k 邻域距离 $dist_k(p)$对应于包含 p 的 k 个最近邻点的 p 的邻域半径.

定义 6. k 邻域点集.对 $p\in D$,p 的 k 邻域点集定义为 $N_k(p)$,$N_k(p)=\{q|dist(p,q)\le dist_k(p)\}$.

定义 7. k 邻域半径(k-radius).对 $p\in D$,p 的 k 邻域半径定义为 $radius_k(p)$,满足 $radius_k(p)=\frac{1}{k}\sum_i dist(p,q_i)$,其中 $q_i\in N_k(p)$.

定义 8. 核心点(core point).对 $p\in D$,若 p 为核心点,则有:$radius_k(p)\le\frac{1}{k}\sum_i radius_k(q_i)$,$q_i\in N_k(p)$.

定义 9. 直接密度可达(directly density-reachable).对 $p,q\in D$,如果 $p\in N_k(q)$,并且 q 是一个核心点,则数据对象 p 从对象 q 出发是直接密度可达的.

通过上述定义,算法 k-PCLDHD 不需要输入距离参数 Eps,同时也弱化了 DBSCAN 算法中参数 MinPts 对聚类结果的影响,对高维数据聚类取得了较好的效果.下一节将结合分布式聚类的特点和高维空间数据聚类要求,对算法 k-PCLDHD 进行扩展,提出适合高维数据的分布式聚类算法 LDBDC.

2 LDBDC 算法思想及相关定义

假设有$|S|$个节点 $S_1,S_2,\ldots,S_{|S|}$,节点 S_i 拥有数据集 D_i,数据集具有相同的属性,需要对所有$|S|$个节点上的数据进行聚类挖掘.算法采取分布式聚类的策略,即在 S_i 上运行 k-PCLDHD 密度聚类算法得到 n_{S_i} 个聚簇 $C_1,C_2,\ldots,C_{n_{S_i}}$.在运行 k-PCLDHD 算法的过程中,每个数据点的邻域半径已经生成,随后在各个节点上,对已经生成的聚簇利用 k-Means 聚类算法获取其特殊中心点.

定义 10. 在 S_i 上运行 k-PCLDHD 密度聚类算法得到 n_{S_i} 个聚簇 $C_1,C_2,\ldots,C_{n_{S_i}}$,定义聚簇集 $C^i=\{C_1,C_2,\ldots,C_{n_{S_i}}\}$.

2.1 局部噪声点的处理

在节点 S_i 上运行 k-PCLDHD 密度聚类算法后,在得到聚簇集 C^i 的同时,也生成噪声数据点集 O_i.

定义 11. 局部噪声点集.节点 S_i 上运行 k-PCLDHD 密度聚类算法所得到的局部噪声点集定义如下:$O_i=\{p|p\in D_i\wedge(\forall C\in C^i|p\notin C)\}$.

在数据水平分布的情况下,这些局部噪声点在进行全局聚类后可能有两种情况:(1) 属于全局噪声点;(2) 隶属于某一聚簇.局部聚类模型中必须要体现这些局部噪声点,否则全局聚类的结果将忽视情况(2)的影响,使得聚类结果出现偏差,为此,引入节点 S_i 的局部噪声模型 LO_i.

定义 12. 局部噪声模型.LO_i 为节点 S_i 的局部噪声模型,$LO_i=\{\langle o,1,dist_k^i(o)\rangle|o\in O_i\}$,其中,$dist_k^i(o)$ 表示节点 S_i 上数据点 o 的 k 邻域距离,1 为权重.

2.2 局部聚类模型

为了描述局部聚类模型,引入特殊中心点定义如下:

定义 13. 特殊中心点.对 $C_j\in C^i(1\le i\le|S|,1\le j\le n_{S_i})$中的数据进行 k-Means 聚类,生成一组特殊中心点 $\langle c,\omega,d\rangle$,其中 c 为聚族中心点,ω为权重,它对应 C_j 中隶属于中心点 c 的数据点数目,d 为均值半径:

$$d=\frac{1}{\omega}\sum_{i=1}^{\omega}radius_k(p_i)$$

p_i 为 c 所代表聚簇中的数据点,聚簇 C_j 的所有特殊中心点的集合记作 $SCor_j$.

如图 1 所示,均值半径 d 为隶属于中心点 c 的数据点 k 邻域半径的均值,较之常见的采用中心点与这些数据点间的平均距离或极值距离计算 d,能够更好地反映 c 所代表的这些数据点的密度分布情况.并且,在各个节点

运行 k-PCLDHD 算法的过程中,每个数据点的领域半径已经生成,无须重复计算.

定义 14. 局部聚类模型.节点 S_i 的局部聚类模型定义为 $LocalModel_i=LO_i\cup\bigcup_{j=1..n_{S_i}}\{\langle c,\omega,d\rangle|\langle c,\omega,d\rangle\in SCor_j\}$.

较之采用特殊核心点构造局部聚类模型的方法,该聚类模型既考虑了噪声等异常数据的影响,又有效地减少了参与全局聚类的数据对象的数量.

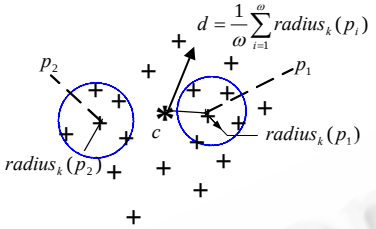

Fig.1 Special center (k=4)

图 1 特殊中心点(k=4)

2.3 全局聚类模型

各个节点 S_i 将局部聚类模型发送到中心节点后,在中心节点上有特殊中心点和局部噪声点两类数据需要分析,生成新的数据集 $D'=\bigcup_{1\le i\le|S|}LocalModel_i$,首先考虑对局部噪声点数据进行处理.

定义 15. 噪声点的全局 k 邻域距离.局部噪声点 $o=\langle o,dist_k^i(o)\rangle,o\in\bigcup_{1\le i\le|S|}LO_i$,则 o 的全局 k 邻域距离为 $glodist_k(o)$,要求存在 $q_1,q_2,\ldots,q_k\in D'$,且对任意 $j(1\le j\le k)$,有 $dist(o',q_j)\le glodist_k(o)$且 $dist(o',q_j)\le dist(o',q_{j+1})$.

性质 1. 在中心节点采用 k-PCLDHD 算法进行全局聚类时,对局部噪声点 $o=\langle o',1,dist_k^i(o')\rangle$,若 $dist_k^i(o)<glodist_k(o)$,则 o'仍为全局噪声点.

证明:在中心节点,影响 o'隶属的数据对象包括节点 S_i 上的噪声点以及特殊中心点、其他节点上的所有噪声点和特殊中心点.

首先对节点 S_i 上 o'的 k 邻域点集中数据点进行分析.这些点经过第 1 轮聚类后,或者隶属于某个聚簇,或者与 o'一样成为 S_i 的局部噪声点.若 $p\in N_k(o')$,且 $p\in C,C\in C^i$,由于 o'为噪声点,故而 $o'\notin C$,且远离 C 的边界点.那么,对聚簇 C 进行 k-Means 聚类生成 C 的特殊中心点集后,必定存在某个特殊中心点$\langle c,\omega,\bar{d}\rangle$,使得 p 隶属于中心点 c,根据 k-Means 聚类算法思想,聚簇 C 中所有数据点将趋近于各个中心点,同时,这些中心点远离边界,可知 $dist(o',c)\ge dist(o',p)$,可见,p 这类噪声点的存在将导致 $glodist_k(o)$的增大.

若 $p\in N_k(o')$,且 $p\in O^j$,即 p 为 S_i 的局部噪声点,根据局部聚类模型的定义,p 将被传送到中心节点,参与全局聚类,$dist(o',p)$保持不变,这一类噪声点不会导致 $glodist_k(o)$的增减.

考虑其他节点 $S_j(j\ne i)$上数据对象对 o'的影响,根据定义 14 易知,仅当有数据对象 $q\in LocalModel_j(q$ 为噪声点或特殊中心点),且 $dist(o',q)<dist_k^i(o')$ 时,$glodist_k(o)$可能减小,噪声点 o'有成为核心点的可能;反之,若 $dist_k^i(o)<glodist_k(o)$,o 的隶属情况不会改变,仍为噪声点. □

根据性质 1,在中心节点上可以对各个节点的局部噪声点进行预处理,对符合性质 1 的噪声点,无须参与全局聚类.

定义 16. 密度吸引子(density attractor).$o_1=\langle c_1,\omega_1,d_1\rangle,o_2=\langle c_2,\omega_2,d_2\rangle$为两个聚类中心点,$o_1$ 对 o_2 的密度吸引子 $da(o_1,o_2)=|dist(c_1,c_2)-(d_1+d_2)|\,e^{\frac{\omega_1}{\omega_1+\omega_2}-\frac{1}{2}}$,其中,$dist(c_1,c_2)$为 c_1 和 c_2 间的欧氏距离.

采用密度吸引子定义中心节点上数据对象间的距离有如下优点:

(1) 充分考虑了特殊中心点所代表的数据点数目及分布对全局聚类的影响.

以图 2 中的两维数据集为例,设 $A=\langle(1,2.4),8,0.7\rangle,B=\langle(3.2,2.4),19,1\rangle,C=\langle(1.4,1.1),6,0.4\rangle$为特殊中心点,隶属于

它们的数据点用“+”表示,由图 2 可知,A,B 所代表的数据点在密度上较邻近,但 $dist(A,B)=2.2>dist(A,C)=1.36$,不能体现这一点,在采用密度吸引子度量时,$da(B,A)=0.613$,$da(C,A)=0.242$,$da(B,A)>da(C,A)$,符合数据实际分布情况.

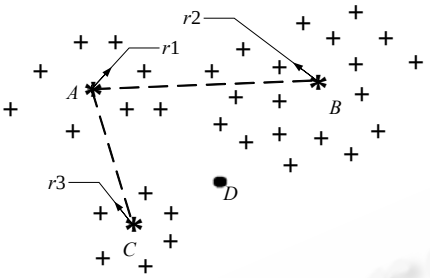

Fig.2 Data objects of central site

图 2 中心节点上的数据对象

(2) 合理地考虑了局部噪声数据对全局聚类的影响.

$D=\langle(2.2,1.3),1,0\rangle$为噪声数据,$D$ 与隶属于 B 的数据点邻近,但 $dist(B,D)=1.487$,$dist(C,D)=0.812$,从单纯距离度量来看,D 反而与 C 更邻近,而密度吸引子 $da(B,D)=0.764$,$da(C,D)=0.589$,$da(B,D)>da(C,D)$符合数据实际分布情况.

(3) 符合密度聚类算法要求的低密度区域向高密度区域聚集的原则.

例如在图 1 中,B 所属区域密度高于 A 对应的区域,则有 $da(A,B)=0.408$,$da(B,A)=0.613$,即 B 对 A 的吸引子大于 A 对 B 的.

用密度吸引子代替欧氏距离应用于定义 5~定义 8.可以得到关于 k 邻域距离、k 邻域点集、k 邻域半径和核心点的新定义,并得到密度吸引子定义下的 k-PCLDHD 算法.在 D'上,运行 k-PCLDHD 算法,得到全局聚类模型.最后,将全局聚类模型发送给各节点 S_i,更新各个数据点所属的聚簇信息.

3 算法性能分析与实验结果

本节对 LDBDC 算法性能进行测试,实验平台配置为 Intel 1.8G/512MB,程序代码用 Visual C++(6.0)实现,实验所使用的数据如下:网络入侵检测数据集 KDD-CUP-1000,取其中 34 维连续属性,该数据集中的数据对象分为 5 大类,包括正常的连接、各种入侵和攻击等,对非数值属性进行数值化处理,添加 4%的干扰数据,并将其均匀分布在 4 个节点上;第 2 种是人工合成数据集 Synthetic DB(如图 3 所示),包括 2 420 个数据点,分布在 4 个节点上,各个节点分布的数据点数目依次为 470,660,880,410(如图 4~图 7 所示).

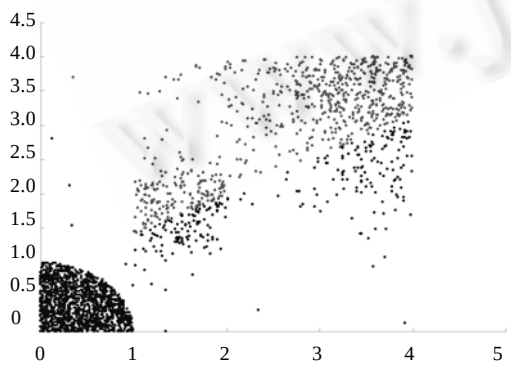

Fig.3 Synthetic data set

图 3 合成数据集

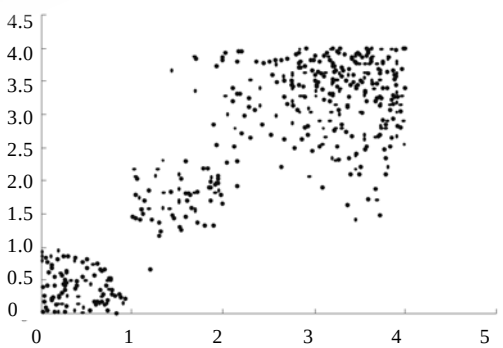

Fig.4 Synthetic data set of site 1

图 4 节点 1 上合成数据集

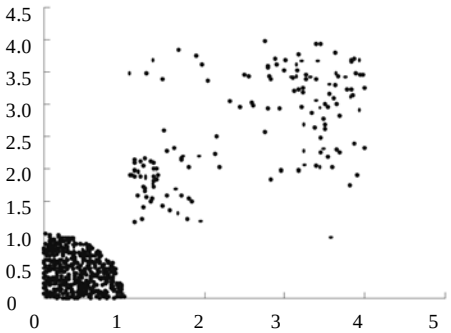

Fig.5 Synthetic data set of site 2

图 5 节点 2 上合成数据集

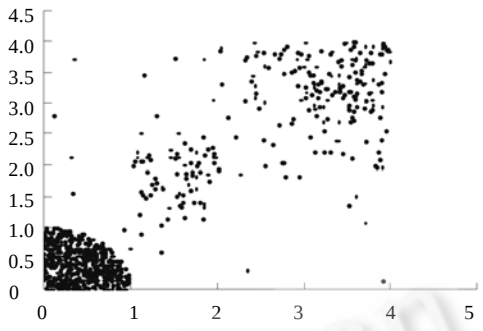

Fig.6 Synthetic data set of site 3

图 6 节点 3 上合成数据集

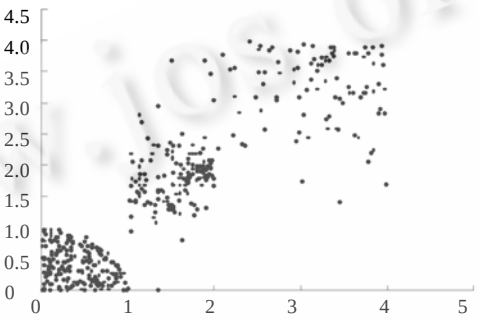

Fig.7 Synthetic data set of site 4

图 7 节点 4 上合成数据集

3.1 聚类质量分析

聚类质量是衡量聚类算法优劣的一项重要指标,本节对 LDBDC 算法、SDBDC 算法和 DBDC 算法的聚类质量进行了分析比较.*k*-PCLDHD 算法的主要计算开销在于分析各个数据点的 *k* 邻域点集和 *k* 邻域半径,为了提高算法生成局部聚类模式的效率,在各个节点上采用文献[6]中设置合适的数据对象作为参考对象,计算各数据点的 *k* 参考点集近似替代其 *k* 邻域点集的方法,使得 *k*-PCLDHD 算法与 DBSCAN 算法具有相同的复杂度.

性质 2. LDBDC 算法的聚类质量优于 SDBDC 算法和 DBDC 算法,算法 SDBDC 的聚类质量优于 DBDC 算法.

证明:由第 2.2 节可知,算法 DBDC 将各个节点执行 DBSCAN 算法生成的特殊核心点提交到中心节点,在中心节点对所有特殊核心点利用 DBSCAN 算法生成全局聚类模式,由于 DBDC 算法不考虑各节点上的局部噪声点,可以得出 DBDC 算法的聚类质量比 DBSCAN 算法要差的推论.

算法 SDBDC 的聚类质量与用户设定的 α 值有关,聚类质量最好的情况对应 α 取最大值时,这时各个节点上几乎所有数据点都将作为代表点参加全局聚类,算法退化成 DBSCAN 算法,可知算法 SDBDC 在最优状况下,具有与 DBSCAN 相似的聚类质量.

算法 LDBDC 以 *k*-PCLDHD 算法为基础,在文献[6]中已经验证了 *k*-PCLDHD 算法的聚类质量优于 DBSCAN;当数据分布规范、数据维度较低、*k*-PCLDHD 算法的优点不能充分体现的情况下,算法聚类质量趋近于 DBSCAN.由此得出,算法 LDBDC 的聚类质量优于 SDBDC 和 DBDC,算法 SDBDC 的聚类质量优于 DBDC 算法. □

进一步对算法的聚类质量进行实验分析,采用文献[8]给出的判断聚类质量的方法,文献[8]提出类内距离和类间密度等概念,并给出聚类质量判定式:$S\_Dbw(c)=Scat(c)+Dens\_bw(c)$,其中 c 为生成的簇集,类间密度

Dens_bw(*c*)用来衡量各个类的平均密度关系,该值较小表明,聚类簇集的类间区分度较好,较小的类内距离 *Scat*(*c*)表明,同一类中数据对象间的相似性较高(具体定义见文献[8]).实验中,*k*-PCLDHD 算法参数 *k* 取 6,*k*-Means 算法参数 *k* 取 8;DBDC 算法参数 Min*Pts* 和 *Eps* 以及 SDBDC 算法的参数ε由各个节点数据集采样估算确定.实验结果如图 8 和图 9 所示,图 8 对应 SDBDC 算法参数α取 0.2 的情况.

分析图 8 可知,算法 LDBDC 对两类测试数据的聚类质量均优于 SDBDC 算法和 DBDC 算法,验证了算法 LDBDC 对含噪声数据以及分布异常数据的处理能力优于 SDBDC 算法和 DBDC 算法的结论.图 9 为不同参数α时,SDBDC 算法聚类精度的比较,由图可知,随着α的增大,SDBDC 算法聚类质量提高,α取值大于 0.2 后,随着α的增大,算法聚类质量的变化趋缓,但整体上,α的增大并不能改变算法 SDBDC 的聚类质量差于图 8 中 LDBDC 算法的相应聚类质量的情况.

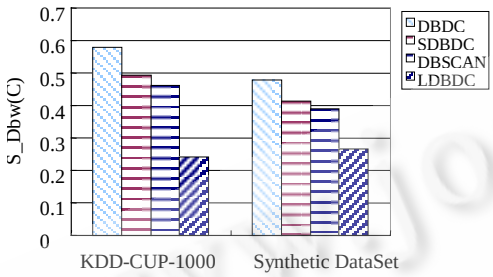

Fig.8 Precision comparing of each algorithm

图 8 不同算法的精度对比

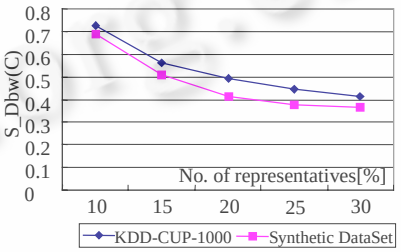

Fig.9 Precision comparing of SDBDC (with different parameter a)

图 9 SDBDC 算法的精度对比(参数 a 不同)

文献[6]分析验证了 *k*-PCLDHD 算法的聚类质量对参数 *k* 有较好的适应性,而 LDBDC 算法以 *k*-PCLDHD 算法为基础,由此可以得出推论,LDBDC 算法的聚类质量对参数 *k* 同样具有较好的适应性.

3.2 算法效率分析

本节对 LDBDC 算法、SDBDC 算法和 DBDC 算法的聚类效率进行了分析比较.假设分布节点数为|*S*|,各节点上局部数据集 D_i 平均包含 *N* 个数据点,各节点上的平均聚簇数为 n_s,各节点局部噪声点的平均值为 n_o,n_c 为 DBDC 算法中各个聚簇包含的完全特殊核心点的平均数目值,CPU 的单位计算时耗为 t_C,节点和中心节点间传送单位数据的时耗为 t_S(假设各算法向中心节点提交的单个代表点占用字节数相同),通常有 $t_S>>t_C$.

性质 3. 在确保算法有较高聚类质量的前提下,LDBDC 算法的效率优于 SDBDC 算法,并且,SDBDC 算法的时间消耗与α近似呈线性关系.

证明:分布式聚类算法的时间消耗主要由 3 部分构成:各节点生成局部聚类模式的时耗、各节点与中心节点的通信时耗以及中心节点上生成全局聚类模式的时耗,为了讨论方便,忽略中心节点生成全局聚类模式后将全局聚类模式传回各节点及各节点更新数据点聚簇属性的时间消耗(该部分时间消耗可以近似认为 3 种算法是相同的),由上文对 3 种算法的分析可知,在各个节点上生成局部聚类模式的时间消耗近似相同,均近似为 N^2t_C,各算法的近似时间消耗分别如下:

$$T_{LDBDC}\approx N^2\ t_C+(k\ n_s+n_o)t_S+((k\ n_s+n_o)|S|)^2\ t_C$$
$$=(N^2+(k\ n_s+n_o)^2|S|^2)t_C+(k\ n_s+n_o)t_S.$$
$$T_{SDBDC}\approx N^2\ t_C+\alpha N\ t_S+(\alpha N|S|)^2\ t_C$$
$$=(N^2+\alpha^2N^2|S|^2)t_C+\alpha N\ t_S.$$
$$T_{DBDC}\approx N^2\ t_C+n_sn_ct_S+(n_sn_c|S|)^2\ t_C$$
$$=(N^2+n_s^2n_c^2\ |S|^2)t_C+n_sn_ct_S.$$

根据第 3.1 节对聚类质量的分析可知,SDBDC 算法在 α 取值较大时才可能取得较理想的聚类质量,而参数 *k* 的取值较小,n_s 为各节点的平均聚簇数,可得出 $kn_s<<N$ 的推断,且局部噪声点仅占数据集较小的比例,有 $n_o<<N$.

因此可以得出推断:在α取值较大时,有$(k\ n_s+n_o)<\alpha N$,从而可以得出:在α取值较大时,LDBDC 算法的效率优于 SDBDC 算法的推论.

对 T_{LDBDC} 和 T_{DBDC} 进行分析,由于 n_o 以及 n_c 的大小与具体数据集密切相关,因而 T_{LDBDC} 和 T_{DBDC} 的关系需根据具体数据集进行分析.

对 T_{SDBDC},由于 $t_S>>t_C$,T_{SDBDC} 近似等于 $\alpha N t_S$,因此有 SDBDC 算法的时间消耗与 α 近似呈线性关系的推论.

□

进一步通过实验对算法的效率进行分析,实验中 *k*-PCLDHD 算法参数 *k* 取 6,*k*-Means 算法参数 *k* 取 8;DBDC 算法参数 Min*Pts* 和 *Eps* 以及 SDBDC 算法的参数ε由各个节点数据集采样估算确定.实验结果如图 10 和图 11 所示.图 10 对应 SDBDC 算法参数α取 0.2 的情况,由实验结果可知,算法 LDBDC 的效率优于 SDBDC 算法,对所采用的 KDD Cup 数据集和合成数据集,算法 LDBDC 的效率优于 DBDC 算法.图 11 对应的α值由 0.1 增加到 0.3 时,SDBDC 算法的执行时间,实验结果验证了性质 3 关于 SDBDC 算法的时间消耗与α值近似呈线性关系的推论;进一步地,对照图 9 和图 11 的实验结果可以发现,随着α值的增大,SDBDC 算法的聚类质量渐趋提高,且聚类质量始终弱于 LDBDC 算法,付出的代价是,算法的时间消耗随着α的增大近似呈线性激增.

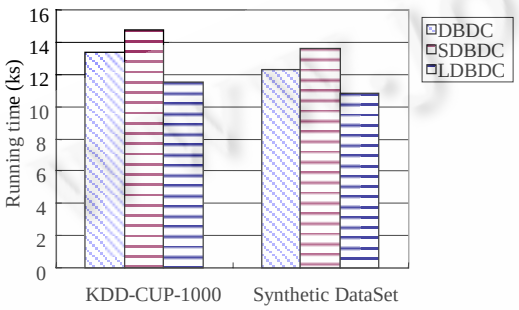

Fig.10 Running time comparing of each algorithm

图 10 不同算法的执行时间对比

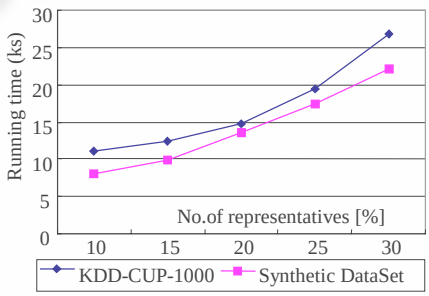

Fig.11 Running time comparing of SDBDC (with different parameter α)

图 11 SDBDC 算法的执行时间对比(参数α不同)

4 结 论

本文针对高维数据水平分布聚类问题,以 DBDC 算法为基础,采用局部密度聚类思想,提出一种有效的分布式密度聚类算法 LDBDC.算法可以有效解决已有分布式密度算法存在的对噪音和异常数据处理能力弱、不适应高维数据以及各节点局部聚类结果规模较大的不足.理论分析和实验结果表明,算法在聚类质量和效率上优于已有算法,算法是有效可行的.进一步地,我们将针对各个节点数据集出现更新时的全局聚类模式的分布式更新问题进行研究.

References:

[1] Inderjit S. Dhillon, Dharmendra S. Modha. A data-clustering algorithm on distributed memory multiprocessors. In: Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining (SIGKDD'99).1999. 245–260. http://www.cs.rpi.edu/~zaki/WKDD99/dhillon.ps.gz

[2] Nagesh HS, Goil S, Choudhary A. A scalable parallel subspace clustering algorithm for massive data sets. In: Proc. of the 2000 Int'l Conf. on Parallel. 2000. 477–484. http://citeseer.ist.psu.edu/377365.html

[3] Januzaj E, Kriege HP, Pfeifle M. Towards effect and efficient distributed clustering. In: Proc. of the 3th IEEE Int'l Conf. on Data Mining. 2003. http://citeseer.ist.psu.edu /januzaj03towards.html

[4] Beyer K, Goldstein J, Ramakrishnan R. When is 'Nearest Neighbor' Meaningful?. In: Beeri C, Buneman P, eds. In: Proc. of the 7th Int'l Conf. on Database Theory (ICDT'99). 1999. 217–235. http://citeseer.ist.psu.edu/605885.html

[5] Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis E, Han J, Fayyad UM, eds. Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining. Oregon: AAAI Press, 1996. 226−231.

[6] Ni WW, Sun ZH, Lu JP. K-LDCHD: A local density based *k*-neighborhood clustering algorithm for high dime nsional space. Journal of Computer Research and Development, 2005,42(5):784−791 (in Chinese with English abstract).

[7] Januzaj E, Kriege HP, Pfeifle M. Scalable density-based distributed clustering. In: Proc. of the 8th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD), Knowledge Discovery in Databases: PKDD 2004. Berlin: Springer-Verlag, 2004. 231−244. http://www.sse-tubs.de/publications/Januzaj_et_al_PKDD_04.pdf

[8] Halkidi M, Vazirgiannis M. Clustering validity assessment: Finding the optimal partitioning of a data set. In: Proc. of the 1st IEEE Int'l Conf. on Data Mining. 187−194. http://citeseer.ist.psu.edu/519636.html

附中文参考文献:

[6] 倪巍伟,孙志挥,陆介平.K-LDCHD:高维空间 *k* 邻域局部密度聚类算法.计算机研究与发展,2005,42(5):784−791.

倪巍伟(1979－),男,江苏淮安人,博士,副教授,主要研究领域为空间数据库,数据库知识发现.

吴英杰(1979－),男,博士生,主要研究领域为数据库知识发现.

陈耿(1965－),男,博士,教授,CCF 高级会员,主要研究领域为模式识别,数据库知识发现.

孙志挥(1941－),男,教授,CCF 高级会员,主要研究领域为数据库系统,应用及知识发现.