k-DmeansWM：一种基于P2P网络的分布式聚类算法

李 榴 唐九阳 葛 斌 肖卫东 汤大权
（国防科技大学信息系统与管理学院 长沙 410073）

摘 要 传统的分布式聚类算法设立中心节点来实现聚类过程的控制，这不仅降低了系统可靠性，而且容易出现单点失效问题。提出一种基于P2P网络的分布式聚类算法k-Dmeans Without Master（简称k-DmeansWM），即采用对等分布的思想，摒弃中心节点，完全由对等节点来实现聚类过程的控制。理论分析与实验结果表明，k-DmeansWM在保证聚类准确性与效率的情况下，大大提高了系统的可靠性与扩展性。

关键词 分布式聚类，P2P，可靠性
中图法分类号 TP393 **文献标识码** A

k-DmeansWM: An Effective Distributed Clustering Algorithm Based on P2P

LI Liu TANG Jiu-yang GE Bin XIAO Wei-dong TANG Da-quan
(College of Information Systems and Management, National University of Defense Technology, Changsha 410073, China)

Abstract Centralized master node is established in normal distributed clustering algorithms in order to control clustering processes, which reduce the system's reliability, single point failure problem occurs easily. The effective distributed clustering algorithm k-Dmeans Without Master(k-DmeansWM) based on peer-to-peer concepts, which abandons the master node, uses peer nodes to achieve controlling of the clustering process entirely. Both theoretical analysis and experimental results show that k-DmeansWM not only ensures the accuracy and efficiency of cluster cases, but also greatly improves the reliability and scalability of the system.

Keywords Distributed clustering, P2P, Reliability

1 引言

聚类分析是数据挖掘领域的一项非常重要的研究课题。迄今为止，人们已经提出了许多聚类算法，如k-means[1]，DBSCAN[2]，Birch[3]，Cure[4]，Sting[5]等，但这些算法只适用于集中式数据的聚类。近年来，随着卫星遥感、生物基因分析、网站用户个性化等技术的发展，数据库中存储的数据量越来越庞大，这使得针对超大规模数据集的聚类分析变得尤为重要，然而现有的聚类算法对于超大规模数据的聚类均存在着扩展性与效率等方面的问题。

在这样的背景下，人们提出了由多台计算机共同参与的分布式聚类算法[6-9]，大多是传统聚类算法在分布式环境下的拓展与改进。S. Kantabutra等人[6]提出的k-Dmeans算法基于k-means[1]划分聚类思想，实现分布式的聚类。中心节点将数据集随机划分为k个子集，分布到k个子节点上（每个节点对应一个聚簇C_i，$1\leqslant i\leqslant k$），各子节点计算其对应点集的中心点，并将其中心点通知给另外$k-1$个子节点，各子节点计算自己的相应数据集中的数据点到各中心点的距离，按每个数据点隶属于距其最近的中心点所代表的聚簇这一规则进行聚类，并将不属于自己的数据点传递给与该数据点所属聚簇相对应的子节点。这一过程迭代进行，直到满足判别函数的要求（例如平方差值稳定）。该算法可以有效解决k-means算法的扩展性问题，但每一次迭代过程都由中心节点控制，容易造成网络拥塞与单点失效，系统的可靠性较低。郑苗苗等人[10]对k-Dmeans算法进行改进，提出了DK-Means算法，在每次迭代过程中，站点间只需传送各簇信息就可以达到分布式聚类的目标，降低了分布式聚类过程中的数据通信量，在保证聚类效果的同时大大提高了聚类速度。

然而，现有的分布式聚类算法都必须存在一个中心节点作为控制聚类过程及资源分发的核心。在现有的复杂网络环境下，这种集中式网络结构抗毁能力非常弱，一旦中心节点受到攻击，整个系统就会瘫痪。在对等网络的研究领域，拥有中心节点的集中式网络结构已经逐渐被完全对等的全分布式网络结构所替代。全分布式网络结构中的节点均为功能相同的对等节点，并不存在功能相异的中心节点，其优势在于各节点负载均衡，网络扩展能力强，不会出现某些节点因为事务处理过多而死机，继而导致系统崩溃单点失效的问题。在全分布式网络下实现分布式聚类，则需要对k-Dmeans进行改进，去除中心节点，由对等节点来实现聚类过程的控制。

本文所提出的分布式聚类算法k-DmeansWM，就是建立

到稿日期：2009-02-25 返修日期：2009-05-08 本文受国家自然科学基金项目（60172012），湖南省自然科学基金项目（03JJY3110）资助。
李 榴（1985－），男，硕士生，主要研究方向为对等计算、无线传感器网络等，E-mail：2815010@sina.com；唐九阳（1978－），男，博士，讲师，主要研究方向为对等计算、信息集成、知识管理等；葛 斌（1979－），男，硕士，讲师，主要研究方向为信息资源管理、语意检索等；肖卫东（1968－），男，博士，教授，主要研究方向为信息管理、信息系统集成、对等计算；汤大权（1971－），男，博士，副教授，主要研究方向为信息资源管理、信息检索。

在 P2P 网络基础之上的，它摒弃中心节点，所有参与聚类的节点的能力相同，在每一轮次的聚类过程中，自动选择网络带宽最大、能力最强的节点作为中心节点，实现聚类过程的控制。k-DmeansWM 在保证聚类准确性与效率的情况下，大大提高了系统的可靠性与扩展性。

2 k-DmeansWM 算法

2.1 算法思想

结合对等网络的思想，k-DmeansWM 算法不再有中心节点与子节点之分，而将中心节点的功能融合到每一个对等节点中，网络结构的变化如图 1 所示。在此算法中，所有节点均为对等节点，即节点的代码相同，功能相同。在聚类过程中，每一轮迭代的控制角色由对等节点选举出的一个节点来担任，选举的标准为节点运算能力与网络带宽。被选举出的节点会接收其他节点发送的判别函数值 E_{ij}，通过计算 $\sum_i E_{ij}$ 来判定聚类迭代是否终止。

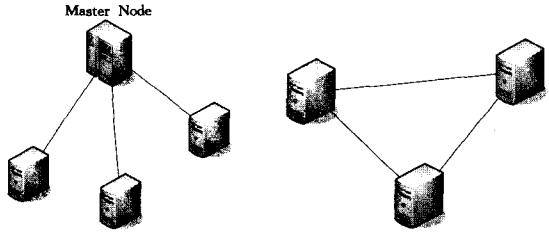

图 1 集中式网络结构与 P2P 网络结构比较

对等节点拥有的能力为：

1)开始一次新的聚类。在 P2P 网络中所有对等节点均已启动完毕后，任意一个节点可以开始一次新的聚类过程。

2)对子数据集进行聚类：对自己所拥有的子数据集进行聚类操作，此过程与 k-Dmeans 算法中的子节点的聚类过程相同。

3)通过判别函数 E 来判断是否终止聚类：每一个节点均会收到其他节点发送的 E_{ij}，通过 $\sum_i E_{ij}$ 作为全局判别式来判定是否达到结束聚类的条件。

4)每一个对等节点在初始状态时都拥有随机大小的子数据集，各节点子数据集的大小不必相等，子数据集中包含的数据可有重复。与 k-Dmeans 算法中必须由中心节点来分配初始子数据集相比，本算法大大简化了初始过程，提高了可扩展性。

2.2 算法描述

输入：全局数据集合 D

输出：K 个聚簇

步骤：

1)Each node has a subset P_i which is part of set D;

2)While E is not stable;

3)Compute a mean $\overline{X}_{me}$ of the subset P_i;

4)Broadcast the mean $\overline{X}_{me}$ to every other nodes;

5)Compute distance $d(i,j)$, $1\leqslant i\leqslant K$, $1\leqslant j\leqslant |P_i|$ of each vector in P_i such that $d(i,j)=||\overline{X}_i-v_j||$;

6)Choose vector members of the new K subsets according to their closest distance to $\overline{X}_i$, $1\leqslant i\leqslant K$;

7)Send K subsets to the node whose $\overline{X}_{me}$ the distance is the most shortest with;

8)Compute new mean $\overline{X}_{me}$ of the new subset P' which contains old vectors and received vectors;

9)Compute E_i of the new subset P_i';

10)Broadcast E_i to other nodes;

11)Compute $E=\sum_{i=1}^{K} E_i$ when receiving E_i completed;

2.3 算法分析

k-DmeansWM 在步骤 9)－步骤 11)中对 k-Dmeans 进行了改进，通过增加节点间的信息交互，以网络流量的小幅增加来换取系统可靠性的提高。下面从网络流量与系统可靠性两个方面对 k-DmeansWM 与 k-Dmeans 进行理论分析。

2.3.1 网络流量

设系统中节点数量为 n，网络中存在的数据包有如下 3 种：

1)传递 $\overline{X}_{me}$ 的数据包——P_X

2)传递 K subsets 的数据包——P_S

3)传递 E_i 的数据包——P_E

对于 k-Dmeans 算法所描述的系统，假设完成一次聚类活动需要进行 k 轮次，网络总流量为

$$F_{\text{k-Dmeans}}=\sum_{i=1}^{k}[n(n-1)P_{Xi}+n(n-1)P_{Si}+2nP_{Ei}]$$

对于 k-DmeansWM 算法所描述的系统，增加的流量主要体现在 P_E 上。在最差情况下，完成一次聚类活动，即每一轮中，n 个节点均完全接收到其他 $n-1$ 个节点的 P_E 时，网络总流量为

$$F_{\text{k-DmeansWM}}=\sum_{i=1}^{k}[n(n-1)P_{Xi}+n(n-1)P_{Si}+2n(n-1)P_{Ei}]$$

在最好情况下完成一次聚类活动，即每一轮中，某一节点首先完全接收其他 $n-1$ 个节点的 P_E，而其他所有节点均未开始接收时，网络总流量为

$$F_{\text{k-DmeansWM}}'=\sum_{i=1}^{k}[n(n-1)P_{Xi}+n(n-1)P_{Si}+2nP_{Ei}]$$

此时 $F_{\text{k-DmeansWM}}=F_{\text{k-Dmeans}}$。可以看出，在最坏情况下，k-DmeansWM 算法的网络流量比 k-Dmeans 的流量多 $\sum_{i=1}^{k}2n(n-2)P_{Ei}$。而在实际系统中，每一轮次均出现最坏情况的概率极低，因此增加的流量相对于整个系统的网络流量来说是非常小的。

2.3.2 系统可靠性

设中心节点的可靠性为 $R_m(t)=\frac{1}{k_m t+1}$，$t\in[0,+\infty)$，子节点与对等节点的可靠性相同，均为 $R_s(t)=\frac{1}{k_s t+1}$，$t\in[0,+\infty)$，其中由于中心节点事务处理更多，网络交互更为频繁，因此可靠性比子节点要低，即 $k_m>k_s$。

对于 k-Dmeans 算法所描述的系统，总体可靠性为

$$R_{\text{k-Dmeans}}(t)=R_m(t)\sum_{i=1}^{n-1}R_s(t)$$

对于 k-DmeansWM 算法所描述的对等网络系统，总体可靠性为

$$R_{\text{k-DmeansWM}}(t)=\sum_{i=1}^{n}R_s(t)$$

由 $k_m>k_s$ 可得 $R_{\text{k-DmeansWM}}(t)>R_{\text{k-Dmeans}}(t)$，即 k-DmeansWM 算法能够明显增强系统的整体可靠性。

以上从静态的、全局的角度分析了 k-DmeansWM 算法能够提高系统的可靠性。接着考虑其在动态聚类过程中的可靠

性。对于 k-Dmeans 算法所描述的系统，在聚类过程中，中心节点的失效直接导致了聚类过程无法继续进行，进而系统崩溃。而对于 k-DmeansWM 算法所描述的对等网络系统，我们设置一个超时值 Timeout，在聚类的某一轮次中，若充当中心节点的子节点失效，则在 Timeout 后剩余的子节点中重新选举出一个新的中心节点，继续未完成的聚类过程。即 k-DmeansWM 算法实现了聚类过程的自动容错与自我修复，保证聚类过程不会因为节点的失效而异常中止。

3 算法验证

为验证算法的可行性，在 100M 局域网的环境中，使用配置为 Intel Pentium Ⅳ/2. 8G/768M/120G，Windows XP SP2 版的 PC 机进行测试，程序代码均用 C# 实现。

为保证测试效果，测试数据集选用 Iris[11] 与 KDD-Cup-99[12] 两个数据集进行测试。其中 Iris 植物数据集用 4 个特征来区分 3 种植物，每种植物有 50 个样本，共 150 个样本。网络入侵检测数据集 KDD-Cup-99 来源于 1998 DARPA 入侵检测评估程序，每条记录有 41 个特征属性。本实验选择其中 37 个连续的特征属性，筛选出 normal，smurf，neptune 3 类入侵模式，并对数据进行了归一化处理。测试数据集如表 1 所列。

表 1 测试数据集

数据集	维度	数据点个数
Iris	4	150
KDD-Cup-99 900	37	900
KDD-Cup-99 3000	37	3000
KDD-Cup-99 15000	37	15000

数据集被随机划分为 3 个子数据集，存放在每一台 PC 机中。其中验证 k-Dmeans 算法的环境为 4 台 PC 机，作为中心节点的 PC 机没有存放数据；验证 k-DmeansWM 算法的环境为 3 台 PC 机，均存放子数据集。

1)聚类准确率

由图 2 可以看到，k-DmeansWM 比 k-Dmeans 算法的聚类准确率略高，这是由于 k-DmeansWM 在初始状态对子数据集的分布方式比 k-Dmeans 要优，在无中心节点的情况下能够提高聚类质量。

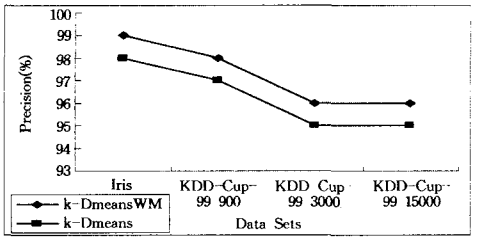

图 2 k-DmeansWM 与 k-Dmeans 聚类准确率

2)聚类效率

由图 3 可以看到，在聚类数据较少时，节点计算时间与网络通信时间处于同一数量级。k-DmeansWM 由于增加了从各节点中选举中心节点的过程，网络通信时间较长，因此聚类速度比 k-Dmeans 稍微慢一些。但是随着聚类数据的增多，节点计算时间占据了更大的比重，网络通信时间越来越被忽略不计，此时 k-DmeansWM 与 k-Dmeans 的聚类效率保持一致，这保证了 k-DmeansWM 的有效性与实用性。

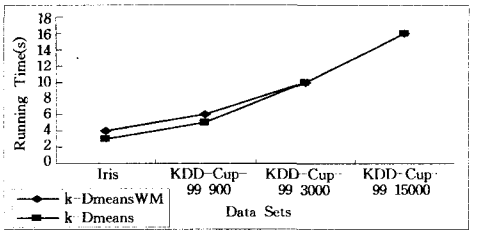

图 3 k-DmeansWM 与 k-Dmeans 聚类效率

结束语 本文针对 k-Dmeans 等分布式聚类算法中由于设立中心节点所带来的单点失效问题，提出了基于 P2P 网络的分布式聚类算法 k-DmeansWM，实现了无中心节点环境下对等节点的聚类，大大提高了系统的可靠性与扩展性，并能达到与 k-Dmeans 等效的聚类质量。理论分析与实验结果表明，本算法是一种有效可行的分布式聚类算法。

参 考 文 献

[1] Han Jiawei, Kamber M. Data Mining: Concepts and Techniques [D]. San Francisco: Morgan Kaufmann Publishers, 2000: 232-233

[2] Ester M, Kriegel H P, Sander J, et al. A density based algorithm of discovering clusters in large spatial databases with noise[C]//Proc. the 2nd Int'l Conf. Knowledge Discovery and Data Mining. Portland: AAAI Press, 1996: 226-231

[3] Zhang Tian, Ramakrishnan R, Livny M. BRICH: An efficient data clustering method for very large database[C]// Proc. ACM SIGMOD Int'l Conf. Management of Data. New York: ACM Press, 1996: 73-84

[4] Guha S, Rostogi R, Shim K. CURE: An efficient clustering algorithm for large databases[C]// Proc. The ACM SIGMOD Int'l Conf. Management of Data Seattle. New York: ACM Press, 1998: 73-84

[5] Wang Wei, et al. STING: A statistical information grid approach to spatial data mining[C]// Proc. 23rd VLDB Conf. San Francisco: Morgan Kaufmann, 1997: 186-195

[6] Kantabutra S, Couch A L. Parallel k-means clustering algorithm on Nows[J]. NECTEC Technical Journal, 1999, 1(1): 243-247

[7] Prodio H, Lawrence H. Scalable clustering : A distributed approach[C]// The IEEE Int'l Conf. on Fuzzy Systems. Budapest, Hungary, 2004

[8] Tasoulis D K, Vrahatis M N. Unsupervised distributed clustering[C]// The IASTED Int'l Conf. on the Parallel and Distributed Computing and Networks. Innsbruck, 2004

[9] Januzaj E, Kriegel H P, Pfeifle M. DBDC: Density based distributed clustering[C]// Proc. of the 9th Int'l Conf. on Extending Database Technology. Berlin: Springer, 2004: 88-105

[10] 郑苗苗，吉根林. DK-Means——分布式聚类算法 K-Dmeans 的改进[J]. 计算机研究与发展，2007，44(Suppl.)：84-88

[11] Iris. http://www.ics.uci.edu/~mlearn/databases/iris, 1998

[12] Kddcup99. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999