计算机科学2005Vol.32№.3

一种有效的并行高维聚类算法*)

冯 永 吴开贵 熊忠阳 吴中福
(重庆大学计算机学院 重庆400030)

摘 要 针对CLIQUE算法聚类结果精确性不高的缺点,提出利用小波变换来生成自适应网格的方法对CLIQUE算法进行改进,将改进算法并行化以增强聚类维数升高时算法的可伸缩性,并将其应用于药品的销售预测。实验表明本算法聚类结果的精确性高,可伸缩性好,并且有效地降低了计算复杂度。

关键词 聚类,CLIQUE算法,小波变换,自适应网格,并行

An Efficient Parallel Clustering Algorithm of High Dimension

FENG Yong WU Kai-Gui XIONG Zhong-Yang WU Zhong-Fu
(College of Computer Science, Chongqing University, Chongqing 400030)

Abstract The classical CLIQUE algorithm is lost to a satisfying accuracy. So, a technique which generates adaptive grid by wavelet transform is put forward to improve it. The scalability of the improved algorithm is enhanced by parallelization with the clustering dimensions increased. In the last, the algorithm is applied to drug sale estimation. The experimental demonstrate that the improved algorithm is more accurate, more scalable, and more efficient in decrease of computation complexity.

Keywords Clustering, The CLIQUE algorithm, Wavelet-transform, Adaptive grid, Parallel

1 引言

将物理或抽象对象的集合分组成为由类似的对象组成的多个类的过程被称为聚类。由聚类所生成的类是一组数据对象的集合,这些对象与同一个类中的对象彼此相似,与其它类中的对象相异[1]。一个好的聚类算法应能识别任意形状的聚类,自动清除孤立点(孤立点指没有包含在任何聚类中的空间对象),对数据的输入顺序不敏感,随输入数据的大小线性地扩展,当数据维数增加时具有良好的可伸缩性。

国内外的学者已经对聚类方法进行了大量的研究。主要的聚类方法可以分为基于距离的聚类、基于密度的聚类以及基于网格的聚类。基于距离的方法其主要思想是同一聚类中对象的距离尽可能小,不同聚类对象间的距离尽可能大。典型的基于距离的方法是k-means和k-medoids,但是它们发现的聚类形状通常是球形,大小也相似。一些改进算法的聚类质量有较大提高,它们可以发现任意形状和大小的聚类,如CURE[2]和BRICH[3]。但是这两种算法的计算复杂度都是$O(n^2)$。基于密度的方法把数据空间中被低密度区域分隔开的对象密集区域当作聚类。代表算法DBSCAN[4]和OPTICS[5]都能发现任意形状的聚类。但它们的计算复杂度依然是$O(n^2)$。基于网格的方法将数据空间划分成为一定数量的网格单元,所有处理以单个的单元为对象。代表算法CLIQUE[6]、WaveCluster[7]和STING[8]的计算复杂度均为$O(n)$。

CLIQUE算法综合了基于密度和基于网格的聚类方法,对于大型数据库中的高维数据聚类非常有效。CLIQUE自动发现高维空间中的高密度聚类,对数据的输入顺序不敏感,无需假定任何规范的数据分布,随输入数据的大小线性地扩展,当数据维数增加时具有良好的可伸缩性。但是CLIQUE不能自动去除孤立点,并且由于方法大大简化,聚类结果的精确性也会降低。本文提出的改进算法利用小波变换来生成自适应网格[9]的方法去除原始数据的孤立点,提高聚类结果的精确性,并将改进算法并行化以增强聚类维数升高时算法的可伸缩性,同时也有效地降低了算法的计算复杂度。

2 CLIQUE算法的局限性及其改进

2.1 CLIQUE算法的局限性

CLIQUE算法将n维数据空间划分为互不相交的长方形单元,识别其中的密集单元。此工作对每一维进行。CLIQUE利用如下性质识别每一维的密集单元:如果一个k维单元是密集的,那么它在$k-1$维空间上的投影也是密集的。即,给定一个k维的候选密集单元,如果检查它的$k-1$维投影单元,发现任何一个不是密集的,那么第k维的单元也不可能是密集的。因此,可以从$k-1$维空间中发现的密集单元来推断k维空间中潜在的或候选的密集单元。CLIQUE算法采用了相对简单的方法,因此也存在着很多的局限性,现归纳如下:

①CLIQUE算法采用固定划分网格的方法,这很容易破坏密集区域的边缘,降低最终结果的准确性。

②CLIQUE算法不能自动去除数据集中的孤立点,需要增加额外的计算步骤去除孤立点,这就增加了计算复杂性。

③CLIQUE算法利用最小描述长度技术来进行剪枝,以减少候选密集单元的数目。但是,利用这种技术可能会剪掉一些密集单元,对最终的聚类结果造成影响。

④CLIQUE算法的很多步骤都采用近似算法,聚类结果的精确性可能因此降低。

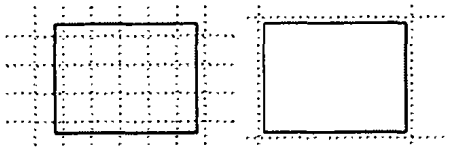
2.2 对CLIQUE算法的改进

针对CLIQUE算法的上述局限性,对CLIQUE算法进行如下改进:

①改进算法采用自适应划分网格,这就避免了固定划分网格所引起的对密集区域边缘的破坏。同时,由于网格的划分

*)基金项目:重庆市科技计划项目应用基础项目(7968)。冯 永 博士研究生,主要研究方向:数据挖掘,并行处理。

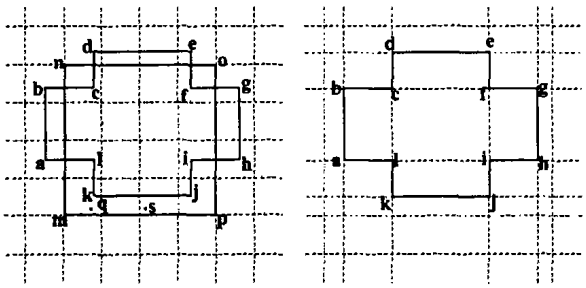
根据实际的密集区域，就不需要 CLIQUE 算法最后找最小覆盖的那一步骤了。图1显示了两种网格的区别。



(a) 固定划分网格　(b) 自适应网格

图1　两种网格的区别

②改进算法采用小波变换形成 $k-1$ 维空间中的聚类，这些聚类由一些自适应网格组合而成。小波聚类强调点密集的区域，而忽略在密集区域外的较弱信息。这样，$k-1$ 维空间中的密集区域就成为了附近点的吸引点(attractor)，距离较远的点成为抑止点(inhibitor)。这意味数据的聚类自动显示出来，并清理了周围的孤立点。小波聚类将 $k-1$ 维空间认为是一个 $k-1$ 维信号。聚类的边界部分对应信号的高频部分，聚类本身对应信号的高振幅低频部分。因此，聚类的边界也就精确确定了，而不需要附加边界修正算法。对 $k-1$ 维空间应用小波变换后，得到 $k-1$ 维空间中的密集单元，这些密集单元数目较少，去除了孤立点，而且形成的网格是自适应的，利用它们再去推断 k 维空间中的密集单元就大大降低了计算量，提高了精度。图2(a)中，字母序列$\{a,b,c,d,e,f,g,h,i,j,k,l\}$包含的部分是原始聚类，CLIQUE 算法找出的聚类是序列$\{m,n,o,p\}$所包围的正方形网格，由16个固定划分的小网格组成，可以发现$\{m,n,o,p\}$包围的正方形网格以外的真实聚类边界丢失了，而孤立点 q 和 s 却被包含在聚类中；图2(b)中，$\{a,b,c,d,e,f,g,h,i,j,k,l\}$包含的部分是应用小波变换后发现的聚类，由5个大小不同的自适应网格组成，形成了精确的聚类边界，孤立点 q 和 s 也被自动清除。



(a) CLIQUE生成的聚类　(b) 小波变换生成的聚类

图2

③为了增强聚类维数升高时算法的可伸缩性，将改进算法并行化。算法的并行模式采用了 Master-Slave 计算模式：Master 运行在主结点上，负责分配数据集、协调负载和汇总从结点的信息；Slave 运行在从结点上，处理 Master 分配给自己的数据集。考虑到每个结点的性能可能存在差异，分配数据时采取自适应的方法，来实现动态的负载平衡。

3　改进 CLIQUE 算法的并行化实现

3.1　算法描述

参数：CDU－密集单元候选集；DU－密集单元；M－生成自适应网格时每一维上预分配的网格数目；k－聚类维数；CDU 信息－CDU 数量和 CDU 维数；DU 信息－DU 数量和 DU 维数。

Master 算法

```
Devide_dataset()－划分数据集
Gen_adptive_grid_m()－生成 master 的自适应网格
Gen_CDU_m()－生成 master 的密集单元候选集
CDU_dist()－分割密集单元候选集
DU_dist()－分割密集单元
begin
While(有新的 CDU 产生)
{ 向 slave 广播 CDU 信息;
  接收来自 slave 的 DU 信息;
  Gen_CDU_m();
  CDU_dist();
  向各 slave 发送新的 CDU;
}
处理结果;
Devide_dataset()
{ dataSize 为初始数据集大小;
  向 slave 发送大小为 dataSize 的数据集;
  datasize=datasize/2;
  while(还有没有分配的数据)
  { 接收完成任务的结点号;
    向完成任务的结点发送大小为 dataSize 的数据集;
    datasize=datasize/2;
  }
  向各结点广播数据分配完成信号;
}
Gen_adptive_grid_m()
{ 向 slave 广播 M 及维数分配方案;
  接收 slave 发送的 DU 信息及小波系数;
  汇总各 slave 发送的 DU 信息及小波系数;
  广播汇总后的 DU 信息及小波系数;
  接收 slave 发送的新的 DU 信息;
  生成自适应网格;
}
Gen_CDU_m()
{ DU_dist();
  向 slave 发送 DU;
  接收各 slave 发送的 CDU 信息;
  汇总所有的 CDU 信息;
  生成 CDU;
}
end
```

Slave 算法

```
Rece_data()－接收 master 的数据
Gen_adptive_grid_s()－生成 slave 的自适应网格
Gen_CDU_s()－生成 slave 的密集单元候选集
Identify_dense_unit()－验证密集单元
WaveletTtrans()－对预分配的单元进行小波变换
begin
接收来自 master 的 CDU 信息;
While(收到新的 CDU)
{ 将本地数据划分到 CDU 中;
  所有 slave 结点两两交换本地 CDU 信息;
  Identify_dense_unit();
  向 master 发送 DU 信息;
  Gen_CDU_s();
  接收来自其它 slave 传来的 CDU;
}
Rece_data()
{ 接收来自 master 的数据分配信号;
  while(数据分配未完成)
  { 接收来自 master 的数据集;
    向 master 发送数据集接收完成信号;
  }
}
Gen_adptive_grid_s()
{ 接收 master 传来的 M 及维数分配方案;
  将本地数据按网格划分好;
  所有 slave 结点两两交换本地单元信息;
  WaveletTtrans();
  将小波变换后产生的 DU 信息及小波系数发送给 master;
  接收 master 传来的汇总后的 DU 信息及小波系数;
  生成本地自适应网格;
  将新的本地 DU 信息发送给 master;
}
Gen_CDU_s()
{ 接收来自 master 发送的 DU;
  while(k－1维 DU 有相同的 k－2维 DU)
  { 依次连接这些 k－1维的 DU，并加入到 CDU 中;
  }
  向 master 传送 CDU 信息;
}
```

3.2　算法的计算复杂度分析

时间复杂度分析　算法的总时间包括用于计算的时间 T_{comp} 和用于通信的时间 T_{comm}。设 N 是数据库中的对象数目，

维数为 K，从结点数为 P，M_P 是各 slave 结点生成自适应网格时每一维上预分配的网格数目最大值的平均值，C_P 是各 slave 结点的每一维中自适应网格数目最大值的平均值。算法在生成自适应网格之前，主要针对预分配网格进行计算，计算复杂度是 $O(M_P^K)$。生成自适应网格后的主要计算是以自适应网格为基础，计算复杂度是 $O(C_P^K)$。由于 $M_P>C_P$，因此 $T_{comp}=O(M_P^K)$。设 S 是结点机间交换信息的大小，T 是 L 条记录通过 I/O 的时间，F 是强制同步的次数，则 $T_{comm}=O(N*T/P*L+F*S*P*K)$。总时间复杂度 $T=T_{comp}+T_{comm}=O(M_P^K+N*T/P*L+F*S*P*K)$。

空间复杂度分析 Master 上需要存储整个数据集，要求空间复杂度为 $O(N)$。各结点上要求的空间复杂度为 $O(N/P)$。因此总的空间复杂度为 $O(N)$。

加速比分析 设 M 是执行串行算法时生成自适应网格时每一维上预分配的网格数目的最大值，C 是执行串行算法时每一维中自适应网格的数目。串行算法的时间复杂度为 $O(M^K+N*T/L)$。并行算法的加速比为：

$$S_p=O(M^K+N*T/L)/O(M_P^K+N*T/P*L+F*S*P*K)$$

上式中 M 大于 M_P，则随着 K 的增长，M^K 将比 M_P^K 增长得快，其它的参数相对于 M^K 与 M_P^K 要小得多。因此，当聚类维数越高时，S_p 的值就越大，表明算法的可伸缩性越好，效率越高。

4 实际应用

药品销售预测是对影响药品销售的各种因素进行非线性分析，及时预知药品的销售趋势和走向，对于指导进货、促销策划、地域分布分析和客户分析等决策提供必要的支持，是聚类分析在数据挖掘中的一个有实际价值和广阔前景的应用。针对药品的销售预测，利用可移植异构编程环境 PVM[10]组成并行计算网络，实现了上述改进算法。

4.1 算法的精确性

为了测试算法的准确性，选取了医药公司1995年1月到2004年6月的部分销售数据，应用本算法对抗生素药品的销售流向进行预测。抗生素药品可用一个5维向量表示：(u,w,x,y,z)，u 表示进货渠道；w 表示商品名称；x 表示销售价格；y 表示销售数量；z 表示化学成分。销售流向分为“华北”、“华东”、“华南”、“华中”4个区域，即4个聚类。通过对20000条抗生素药品记录应用本算法和原始 CLIQUE 算法，比较2种算法的精确性。具体比较结果见表1。

表1 改进算法与 CLIQUE 聚类精确性比较(流量单位：条)

流量分布	华北地区	华东地区	华南地区	华中地区
实际流量	3674	5328	5422	5576
CLIQUE	3800	5425	5398	5377
误差	126	97	24	199
改进算法	3672	5328	5425	5575
误差	2	0	3	1

通过表1可以发现改进算法的聚类误差相对于 CLIQUE 算法的聚类误差要小得多，基本上可以准确地发现聚类。

4.2 算法的效率

为了测试算法的效率，采用8台 HP6000，利用 PVM 组成并行计算网络。数据集分别是医药公司1995年1月到2004年6月的部分销售数据所形成的2维（1000K）、3维(2000K)、4维(4000K)、6维(5000K)、8维(6000K)、10维(8000K)共6个测试数据集。分别将 CLIQUE、串行改进算法、并行改进算法应用于上述数据集，比较3种算法的执行效率，结果见表2。

表2 改进算法与 CLIQUE 的效率比较(单位：秒)

数据维数	2维	3维	4维	6维	8维	10维
CLIQUE	112	213	398	481	563	643
串行改进算法	109	189	347	423	501	579
并行改进算法	18	30	51	60	70	77
加速比	6.1	6.3	6.8	7.1	7.2	7.5

通过表2可以发现，串行改进算法相对于 CLIQUE 的执行效率有所提高，主要是因为应用了小波变换提高了准确性，从而去掉了 CLIQUE 算法中的一些近似步骤，但是提高不是十分显著。并行改进算法明显优于 CLIQUE 和串行改进算法，特别是当聚类维数增长时，算法的加速比呈上升趋势，即算法具有良好的可伸缩性。因此，改进后的 CLIQUE 算法并行化后，适合高维聚类。

结论 ①CLIQUE 算法适用于高维空间聚类，改进的算法采用了 CLIQUE 的主体结构，目的是为了适应高维聚类。

②改进算法利用小波变换生成自适应网格，不但可以自动去除孤立点，形成精确的聚类边界，还可以减少候选密集单元的数目，从而提高算法的准确性，加速了算法的执行效率。

③考虑到 CLIQUE 主要针对高维数据聚类的应用，为了提高算法的可伸缩性，采用 Master-Slave 模式将改进算法并行化实现，并将改进后的并行算法应用到药品销售预测。实验数据充分证明了算法的聚类精度高，当聚类维数增长时，算法的可伸缩性好，执行效率高，适合高维数据聚类。

参 考 文 献

1 Han J, Kamber M. Data Mining: Concepts and Techniques. High Education Press, Morgan Kaufman Publishers, 223～257

2 Guha U, Rastogi R, Shim K. CURE: an efficient clustering algorithm for large databases. Information System, 2001, 26(1): 35～58

3 Zhang T, Ramarkrishnan R, Livny M. BIRCH: an efficient data clustering method for very large database. 1996 ACM 0-89791-794-4/96/0006

4 Ester M, Kriegel H P, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial database with noise. In: Proc. 2nd Int'l Conf. Knowledge Discovering in Database and Data Mining (KDD-96)

5 Ankerst M, Breunig M, Kriegel H P, Sander J. OPTICS: Ordering points to identity the clustering structure. In: Proc. 1999 ACM-SGMOD Int. Conf. Management of Data (SIGMOD'99), 49-60, Philadelphia, PA, June 1999

6 Agrawal J, et al. Automatic subspace clustering of high dimensional data mining applications. In: Proc. ACM SIGMOD Inter'l Conf. Management of Data, 1998. 73～84

7 Sheikholeslami G, Chatterjee S, Zhang A. WaveCluster: A multiresolution clustering approach for very large spatial database. In: Proc. 1998 Int. Conf. Very Large Data Base, New York, 1998. 428～439

8 Wang W, Yang J, Muntz R. STING: A statistical information grid approach to spatial data mining. In: Proc. 1997 Int. Conf. Very Large Data Bases (VLDB'97), Athens, Greece, 1997. 186～195

9 Nagesh H S, Goil S, Choudhary A N. A Scalable Parallel Subspace Clustering Algorithm for Massive Data Sets. In: Intl. Conf. on Parallel Processing, 2000

10 Sunderam V S, Geist G A, et al. The PVM Concurrent System: Evolution, Experiences, and Trends. Applied Mathematical Sciences program of the Office of Energy Research, U. S. Department of Energy, 1993