计算机科学 2005Vol.32No.12

基于数据分区的最近邻优先聚类算法

王 鑫 王洪国 张建喜 谷建军
(山东师范大学信息管理学院 济南 250014)

摘 要 聚类是数据挖掘领域的一个重要研究方向。最近邻优先吸收(NNAF)算法可以快速进行聚类并且能有效处理噪声点,但当数据密度和聚类间的距离不均匀时聚类质量较差。本文在分析 NNAF 算法不足的基础上,提出了一种基于数据分区的 NNAF 算法—PNNAF 算法,较好地改善了聚类质量。

关键词 数据挖掘,聚类,数据分区,最近邻优先吸收

A Data-Partitioning-Based Nearest-Neighbors-First Clustering Algorithm

WANG Xin WANG Hong-Guo ZHANG Jian-Xi GU Jian-Jun
(Information Management School of Shandong Normal University , Jinan 250014)

Abstract Clustering is an important research direction in the field of Data Mining. This paper analyses the Nearest Neighbors Absorbed First (NNAF) clustering algorithm. This algorithm can cluster quickly with noisy . However, clustering quality will degrade when the cluster density and distance between clusters are not even. In this paper, a Nearest-Neighbors-First clustering algorithm based on data partitioning is proposed. The new algorithm improves the quality of clustering.

Keywords Data Mining, Clustering, Data partitioning, Nearest neighbor first

1 引言

近 10 年来,数据挖掘[1]逐渐成为数据库和人工智能等研究领域的一个热点。聚类(Clustering)是数据挖掘中重要的研究课题之一。所谓聚类,就是将物理或抽象对象的集合组成为由类似的对象组成的多个类或簇的过程。由聚类所生成的簇是一组数据对象的集合,同一簇中的对象尽可能相似,而不同簇中的对象尽可能相异。

1.1 相关工作

迄今为止,数据库研究人员已经提出了许多聚类算法,主要有以下几类:划分方法:主要有 K-Means[2],K-Mediods (PAM)以及它们的变种 CLARA 和 CLARANS。层次方法:主要有 BIRCH 算法[3]、CURE[4]算法、最短距离法和 CHAMALEON 算法等。基于密度的方法:主要有 DBSCAN[5], OPTICS,DENCLUE 等。基于网格的方法:主要有 STING[6]方法,Wave Cluster 算法,CLIQUE 算法。基于模型的方法:典型的基于模型的聚类方法有神经网络方法和统计的方法。

1.2 本文的工作

首先介绍了最近邻优先吸收算法 NNAF(Nearest Neighbors Absorbed First)的基本思想,分析其不足;然后针对这些不足提出了基于数据分区(data-partitioning)的最近邻优先吸收算法—PNNAF 算法;接着对新算法进行分析,最后得出了结论。

2 最近邻优先吸收(NNAF)算法

NNAF 算法是基于"同类相近"的思想提出的一种改进的最短距离聚类算法。最短距离法又称最近邻连接法,其基本思想是把两个类的距离定义为两类中距离最近的元素之间的距离。并依此逐次选择最"靠近"的类聚集,直到满足终止条件。

NNAF(Nearest Neighbors Absorbed First)算法的基本思想是:空间中的每一点和与之最近的点属于同一类的可能性最大。如果两个距离最近的点之间的距离小于 d(用户输入的距离阈值),那么就认为它们属于同一类。当某一聚类所包含的元素个数大于 q(用户输入的数量阈值)时,则该类数据成为一个真正的聚类;否则为噪声数据集合。

定义 1 设 V 是高维数据空间中的点集合,$V=\{p_1,p_2,\cdots,p_n\}$,$p_1\in V$,$p_2\in V$, p_1 和 p_2 之间的距离记为 $D(p_1,p_2)$;给定距离阈值 d,$d>0$,则:

1)如果 $D(p_1,p_2)<D(p_1,p_3)<\cdots<D(p_1,p_n)$,则称 p_2 为距离 p_1 最近的点,即 p_2 为 p_1 的最近邻,记为 $MN(p_1)=p_2$;

2)如果 $MN(p_1)=p_2$,并且 $D(p_1,p_2)\leqslant d$,那么 p_2 与 p_1 属于同一类。即:当 p_1 点属于第一类,而 p_2 尚没有归类时,则把 p_2 点也归为第一类;当 p_1 尚没有归类,而 p_2 点属于第一类时,则把 p_1 点也归为第一类;当 p_1 点属于第一类,而 p_2 属于第二类时,则把第一类和第二类合并为一个新类,并把 p_1,p_2 点和分别属于原第一类和第二类的所有点都归于这个新类。

定义 2 设 $V=\{p_1,p_2,\cdots,p_n\}$是一任意空间点集,点 $p_i\in V$, $p_h\in V$ 。

1)V 中任一点 p_i 都存在一个五元组的属性集合(Dir, $Distance$,$Cluster$,$Neighbors$,$Anti\text{-}Neighbors$),其中,Dir 表示点 p_i 的坐标;$Distance$ 表示点 p_i 到其最近邻点之间的距离;$Cluster$ 表示点 p_i 的类别;$Neighbors$ 表示点 p_i 的最近邻点的集合;$Anti\text{-}Neighbors$ 表示以点 p_i 为最近邻点的点集

王 鑫 硕士研究生,研究方向:数据挖掘,知识发现。王洪国 博士后,教授,硕士生导师。张建喜 硕士研究生。谷建军 硕士研究生。

合。

2)如果 $MN(p_i)=p_h$,且 $MN(p_h)=p_i$,则称点 p_i 和 p_h 为互近邻点。

3)如果 $MN(p_i)=p_h$,且 p_h 点的属性 *Cluster* 的值不为空,则称 p_h 点为 p_i 点的已归类近邻点。否则称为未归类近邻点。

显然,互近邻点其实是已归类近邻点的一种特例,即当某点的已归类邻点是以其为最近邻点的点时,已归类近邻点就变成了互近邻点。出现互近邻点和已归类近邻点这两种情况是 NNAF 算法中形成每个聚类的终止条件。

NNAF 算法聚类过程如下:先把一个点作为一类,然后把这一点的所有的最近邻点和所有以这一点为最近邻点的点归于这一类;然后再把所有以新加入这一类的点为最近邻点的点和新加入这一类的点的所有最近邻点也归于这一类。反复这样操作,直到这一类的点不再增加为止,即所有以新加入这一类的点为最近邻点的点和新加入这一类的点的所有最近邻点全部都是已归类点。然后再以一个未归类点为下一个聚类,按上述方法进行聚类操作,直到把所的点都归入某一聚类,这时聚类过程完成。

算法 1 NNAF 算法

输入:点集 $V=\{p_1,p_2,\cdots,p_n\}$,距离阈值 d,数量阈值 q。
输出:每一点的 *Cluster* 属性的值。
Begin
01. For(i=1;n;i++)//调用 SNN 算法(后面叙述)找每一点的最近邻点
02. {
03. 用 SNN 算法找点 p_i 的所有最近邻点及其之间的距离 *Distance*,并在其最近邻点的属性 *Anti-neighbors* 集合中加入 p_i;
04. 建立点 p_i 的属性集五元组;
05. }
06. j=1;//j 为类别编号
07. 堆栈 Stack=NULL;
08. Point=NULL;
09. While(V<>NULL)do
10. {
11. *Push*(V 中任一个点),把该点赋给 *Point*,并把它从 V 中删除,把其 *Cluster* 置为 j;
12. Repeat
13. If 点 *Point* 的 *Neighbors* 和 *Anti-neighbors* 中存在 *Cluster* 值为空的点
14. Then *Push*(*Neighbors* 和 *Anti-neighbors* 中 *Cluster* 值为空的点),把压入堆栈的点的 *Cluster* 置为 j,并把这些点从 V 中删除;
15. *Point*=*Pop*;// 栈顶点的最近邻点和以之为最近邻点的点全为互近邻点和已归类近邻点
16. Until(堆栈 Stack=NULL);
17. j=j+1;
18. }
19. If(某类包含的点的个数>q)
20. Then 输出这些类别
End.

NNAF 算法主要分为 3 个步骤:1)求出每一点的最近邻点;2)求出所有的聚类;3)输出满足阈值条件的聚类。

在搜索某点的最近邻点时,为了避免比较其它所有点到它之间的距离,采用了最近邻搜索 SNN(Searching Nearest Neighbors)算法 ,它的目标是快速找到每一点的最近邻点。该算法只需计算每一点到它们的 d(用户设置的距离阈值)邻域附近的点之间的距离 ,然后通过比较得到距其最近的点。

算法 2 SNN 算法

输入:点集 $V=\{p_1,p_2,\cdots,p_i,\cdots,p_m\}$,其中数据点已按某一维(称为排序维)排序,距离阈值 d。
输出:每一点的五元组属性值,其中,属性 *cluster* 值为空,属性 *Neighbors* 和 *Antineighbors* 中的点分别是满足距离阈值的点。
Begin
01. 点集 S=NULL;
02. for(i=1;i<=m;i++)
03. {
04. 找出在排序维上位于 p_i 点的 d 邻域内的点集 S;
05. if S 中的某点在某一维上不在 p_i 点的 d 邻域内
then 从 S 中删除该点
06. if S<>NULL
then 计算 S 中所有的点到 p_i 点的距离,然后比较得出距 p_i 点最近的点及其之间的距离 *Distance*;
07. if *Distance*<= d
then 把 p_i 的最近邻的点写入其 *Neighbors* 属性,把 *Distance* 写入其 *Distance* 属性,把 p_i 写入其所有最近邻点的 *Anti-neighbors* 属性,p_i 的 *Cluster*=NULL;
08. else *Distance* = NULL, *Neighbors* = NULL, *Anti-neighbors* = NULL, *Cluster* = NULL,认为 p_i 是孤立点;
09. }
End.

SNN 算法避免了求两两之间的距离,只需比较它的一个较小邻域内的点,从而大大提高了计算速度。其时间复杂度为 $O(n*\log(n))$。当数据是通过扫描空间图像所得时,因为扫描后数据自然已按某一维排序,所以其时间复杂度就会降为 $O(n)$。而传统的逐点比较法的时间复杂度为 $O(n^2)$。

但是,SNN 算法要求用户指定一个全局的距离阈值 d,该算法只搜索满足距离阈值条件的最近邻点,对于不满足阈值条件的最近邻点,则在用户重新设定 d 后再用 SNN 算法搜索。因此,距离阈值 d 大小的设定十分重要。然而当数据密度和类间距离分布不均匀时就出现了这样的问题:若根据较密的那些类选取较小的 d 值,那么对于客观上相对较稀的那些类中的对象,则会丢失最近邻点,从而不被用于所在类的进一步扩展,结果是较稀的类被划分成多个性质相似的类。若根据较稀的那些类来选取较大的 d 值,那么离得较近而密度较大的几个类将很可能被合并为同一个类,它们之间的差异也就因为选取较大的 d 值而被忽略。很明显,在上述两种情况下,很难选取一个合适的 d 值来进行聚类且得到比较准确的聚类结果。

基于以上存在的问题,我们提出了基于数据分区[7](data-partitioning)的最近邻优先的聚类算法——PNNAF 算法。

3 基于数据分区的最近邻优先的聚类算法

PNNAF 算法的基本思想是:根据数据库中数据在某一维或多维上的分布特性,将整个数据库数据空间划分为若干个局部区域;然后,依次对每个局部区域使用局部 d 值,用 NNAF 进行局部聚类;最后将各个局部聚类合并,从而完成整个数据库数据的聚类分析。

为了便于说明问题而又不失一般性,下面以二维空间中的数据聚类为例进行讨论。

3.1 数据分区

在二维数据空间里,采用基于数据统计分布特性来进行数据库分区。这里要求所划分的分区能比较真实地反映数据的分布特性。划分标准是这样确定的:首先,对整个数据库进行取样,分别统计投影在 X 轴和 Y 轴上的数据分布特性,根据分析结果,确定要在哪一维上进行分区以及划分为多少个分区和分区的边界,目的是使得在每个分区中数据的密度分布尽可能均匀。在每一维上,我们采用直方图方法进行数据分布的统计分析。

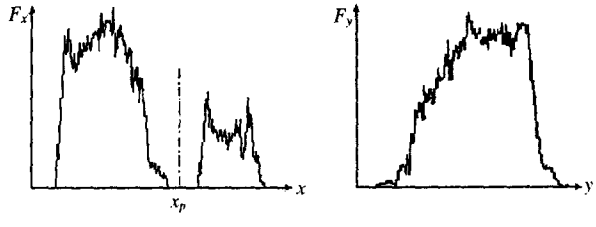

图 2 二维数据库的数据分布图

图 2 所示是一个二维空间数据库取样[8]后分别投影在 X 轴和 Y 轴上的数据分布图。可以看到，X 轴上的分布呈现出两个密集区域，而且两个区域中的密度相差较大；而 Y 轴上的分布呈连贯性。因此我们将数据库在 X 轴上划分为两个部分，分界线为两峰间低谷区域的任意一点对应的扫描线 $x=x_p$。划分为矩形格的好处在于可以简化分区后局部聚类的合并。

3.2 局部聚类

局部聚类采用 NNAF 算法。

为了进行局部聚类的合并，所以在此过程中，需要记录可能在合并阶段有用的那些类的代表点和边界信息，即该类中的某些对象与数据库划分边界之间的距离不大于 d，同时还要记录那些与分区边界距离不大于 d 的噪声点，因为它们可能是全局中某个聚类的边界点或者某一被分割的小聚类中的点。

3.3 局部聚类的合并

数据分区后可能出现一个类被划分到两个相邻的分区中的情况，所以当局部聚类完成后，应对那些被分割在相邻分区中的类进行合并。有以下几种情况：

(1)两个类 A 和 B 的合并

两个类 A 和 B 合并，当且仅当：

①A，B 分别处于相邻的两个分区 P_A，P_B 中；

②设 $d(P_A)$，$d(P_B)$ 分别是 P_A，P_B 的 d 值，$d(P_A,P_B)=\min\{d(P_A),d(P_B)\}$，$E_A$，$E_B$ 是在局部聚类过程中保存的 A 和 B 分区的边界区域点集，对于 $p\in E_A$，$q\in E_B$，满足 $d\{p,q\}\leqslant d(P_A,P_B)$。

(2)归并噪声点

处在分区线附近的噪声点可能是全局中某个类的边界点，因此必须考虑将这些临时噪声点归到相邻分区的某个类中。一个噪声点 p_N 被归入一个类 C，当且仅当：

①p_N 和 C 分别处于相邻的两个分区中；

②$p\in E_C$，E_C 是局部聚类过程中保存在类 C 所在分区的边界区域点集，满足 $D\{p_N,p\}\leqslant d(P_C)$。

(3)由噪声点产生新类

在数据分区阶段，一些小的聚类可能被划分到不同分区中去。由于处在每个分区中的数据量太小，在各分区的局部聚类过程中均被标记为噪声点，因此还需要对这些点进行处理。设 P_A，P_B 为两个相邻分区，E_{PA}，E_{PB} 为这两个分区的边界区域点集。考虑 E_{PA} 和 E_{PB} 中噪声点的集合为 SN，若 $p_0\in SN$，$p_i\in SN(i=1,2,\cdots,m,m\geqslant q)$，且 $p_i\neq p_0$。满足 $D\{p_0,p_i\}\leqslant d(P_A,P_B)$，则认为 p_0 是全局中的一个新类的核心点，$\{p_i\}(i=1,2,\cdots,m)$是这新类的其它点。剩余噪声点按(2)进行处理。

4 实验结果

使用人工合成的数据对本文提出的算法进行了实验。首先使用二维数据进行实验，原始数据的分布如图 3 所示。

在这组数据中，采用了具有噪声的不同分布形状的数据。对于这样的数据分布，很多传统的聚类算法将无能为力，而采用 PNNAF 算法，得到的聚类结果如图 4 所示。

图 3 原始数据集

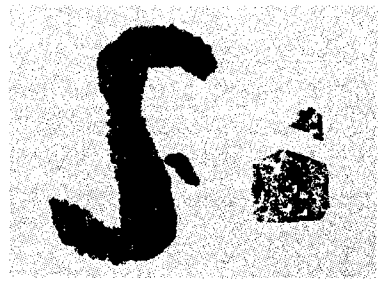

图 4 FNNAF 算法聚类结果

实验结果表明，本文提出的算法可以有效处理具有噪声的、密度分布不均匀的、以任意形状分布的数据的聚类问题。

结论 聚类分析是数据挖掘中广为研究的问题之一。本文从 NNAF 算法出发，针对其局限性，提出了一种基于数据分区的 NNAF 算法。使用模拟数据对其进行测试，结果表明 PNNAF 算法是有效的。

参 考 文 献

1 Han Jiawei, Kamber M. Data Mining : Concepts and Techniques [C], Mongan Kaufmann publishers, 2000. 225～278

2 Kanfan L., Rousseeuw P J. Finding groups in data: an introduction to cluster analysis [M]. New York: John Wiley&-Sons, 1990

3 Zhang T, et al. BIRCH: An efficient data clustering method for very large databases. In: Proc. of the ACM SIGMOD Int'l. Conf. on Management ent of Data. Montreal: ACM Press, 1996. 73～84

4 Guha S, Rastogi R, Shim K. CURE: An efficient clustering algorithm for large databases. In: Proc. of the ACM SIGMOD Int'l. Conf. on Management ent of Data. Seattle: ACM Press, 1998. 73～84

5 Enter M, et al. A density-based algorithm for discovering clusters in large spatial databases with noise: In: Proc. of 2nd Int'1 Conf on Knowledge Discovering in Databases and Data Mining KDD-96). Portland: AAAI Press. 1996

6 Zhang W, et al. STING: A atatistical information grid approach to spatial data mining. In: Proc. of the 23rd VLDB Conference. Athens: Morgan Kanfmann, 1997. 186～195

7 Zhou Shuigeng, et al. A fast density- based clustering algorithm (in Chinese). Department of Computer Science, Fudan University: [TechRep: 1999011]. 1999

8 Vitter J. Random sampling with reservoir. ACM Trans on Mathematical Software, 1985, 11(1):37～ 57